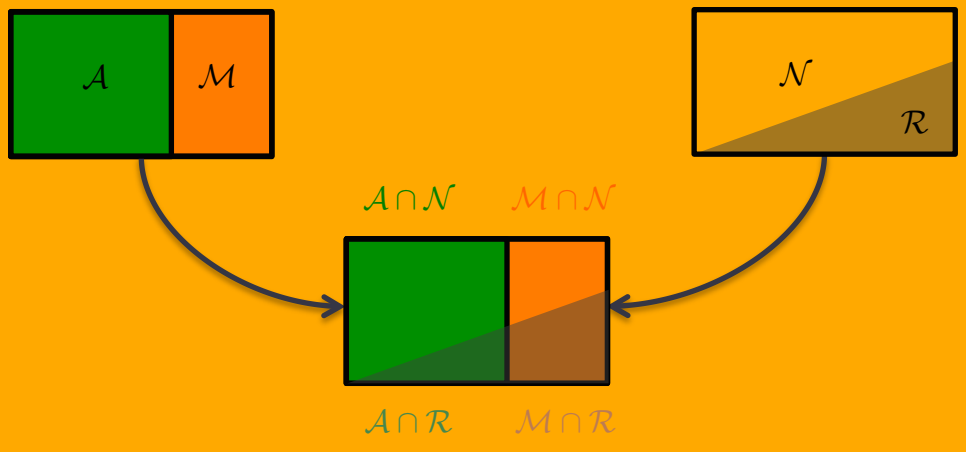
Robust Image Classification with Context and Rejection



Filipe J. C. Condessa

bimagicLab
Dept. of Electrical and Computer Engineering
Carnegie Mellon University

Robust Image Classification with Context and Rejection

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering.

Filipe J. C. Condessa

Carnegie Mellon University
Pittsburgh, PA

May 2016

Keywords: robust classification, classification with rejection, classification with rejection

“We demand rigidly defined areas of doubt and uncertainty!”
Douglas Adams

Abstract

Classifications systems are ubiquitous; despite efforts going into training and feature selection, misclassifications occur and their effects can be critical. This is particularly true in classification problems where overlapping classes, small or incomplete training sets, and unknown classes occur. In this thesis, we mitigate misclassifications and their effects by adapting the behavior of the classifier on samples with high potential for misclassification through the use of robust classification schemes that combine context and rejection. We thus combine the advantages of using contextual priors in classification with those of classification with rejection. In classification with rejection, we are able to improve classification performance at the expense of not classifying the entire data set.

We thus add the following tools to the robust classification toolbox: 1) we derive performance measures for evaluating of classifiers with rejection; 2) we create a family of convex algorithms, SegSALSA, to classify with context; 3) we design architectures for robust classification with context and rejection that encompass interactions between context and rejection. We validate our approach on two different real-world data sets: histopathological and hyperspectral images.

Acknowledgments

Every journey begins with a single step. Yet, no single step on this journey would have been possible without the support of many.

This work would have not be possible without the endless support of my family: my girlfriend Patrícia, my parents Pedro and Isabel, and my sister Luzia.

To all my friends and colleagues in both sides of the Atlantic (and to those whose paths took them to different oceans, seas or lakes), I owe them a heartfelt thank you for making this an unique journey. I have to acknowledge the help from my past and present colleagues from bimagicLab and from the Center from Bioimage Informatics in Pittsburgh, Ramu Bhagavatula, Anupama Kuruvilla, Mike Mccann, Jackie Chen, Siheng Chen, Rohan Varma, Anuva Kulkarni, Soheil Kolouri, and Serim Park, and from the Pattern and Image Analysis group in Lisbon, Miguel Simões, Lina Zhuang, Yi Liu, Marina Ljubenovic, Milad Niknejad, Joshin Krishnan, Joanna Bachmatiuk, and Afonso Teodoro. To all my friends that accompanied me on this path, from West to East, I would like to thank them for their help. Pedro Osório for providing a constant flow of interesting ideas, Anupama Kuruvilla for all those long walks and longer talks, Anuva Kulkarni for tolerating me for misspelling her name in the most creative ways, Sérgio Pequito for being a mentor both in Pittsburgh and in Lisbon, João Veiga for the continually expanding my musical horizons, and Carlos Santiago and Catarina Barata for the many and long conversations over lunch.

I would like to thank Prof. Gustavo Rohde for suggesting the application of classification with rejection to the histopathology image classification problem in 2011, Dr. John Ozolek and Dr. Carlos Castro for providing us the histopathology data and their insight on the problem, and to the members of the thesis proposal committee, Prof. José Moura, Prof. Aswin Sankaranarayanan, and Prof. Mário Figueiredo, for their helpful suggestions throughout the proposal.

I would like to thank the members of the doctoral committee, Prof. Jelena Kovačević (chair), Prof. José Bioucas-Dias, Prof. Aswin Sankaranarayanan, Prof. José Moura, Prof. Mário Figueiredo, and Prof. Pedro Aguiar.

I gratefully acknowledge the support from the Portuguese Science and Technology Foundation and the CMU-Portugal (ICTI) program under grant SFRH/BD/51632/2011.

Finally, I would like to thank my advisors Jelena Kovačević and José Bioucas-Dias for their invaluable guidance and support through this journey.

Contents

List of Figures

List of Tables

List of acronyms

ADMM	Alternated Direction Method of Multipliers
CRF	Conditional Random Field
DRF	Discriminative Random Field
EM	Expectation Maximization
H&E	Hematoxylin and Eosin
HV	Histopathology Vocabulary
INTRASC-MK	Simplified Superpixel-based Classification via Multiple Kernels
JCR	Joint Context and Rejection
JCR-E	JCR with entropy-weighted probability of classifier failure
JCR-U	JCR with uniform probability of classifier failure
LORSAL	Logistic Regression via variable Splitting and Augmented Lagrangian
LORSAL-MLL	LORSAL with multilevel logistic Markov random field priors
LBP	Loopy Belief Propagation
MAP	Maximum *A Posteriori*
MLR	Multinomial Logistic Regression
MLR-GCK	MLR with generalize composite kernel
MMAP	Marginal Maximum *A Posteriori*
MPO	Moreau Proximity Operator
MSS	Minimum Superpixel Size
MRF	Markov Random Field
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
SALSA	Split Augmented Lagrangian Shrinkage Algorithm
SegSALSA	Segmentation via Split Augmented Lagrangian Shrinkage Algorithm
SegSALSA-*	Family of SegSALSA based algorithms
SegSALSA-VTV	SegSALSA using Vectorial Total Variation regularization
SegSALSA-GTV	SegSALSA using Graph Total Variation regularization
SegSALSA-STR	SegSALSA using Structure Tensor Regularization
SC-MK	Superpixel-based Classification via Multiple Kernels
SCR	Sequential Context and Rejection
SVM	Support Vector Machine
SVM-CK	SVM with Composite Kernels
TRWS	Tree-reweighted Message Passing
TV	Total Variation
VTV	Vectorial Total Variation
WTA	Winner Take All

Part I

Introduction and background

Chapter 1

Introduction

Machine learning and pattern recognition play an increasing role in the modern world. As classification techniques are applied to a large variety of applications, a significant amount of resources is currently invested into problem specific areas, namely in feature design, feature selection, classifier design, construction of representative training sets. However, in a large family of classification problems, the performance of the classification is limited by the inherent characteristics of the problem, such as nonrepresentative training sets, nonseparable classes, or even unknown classes. To address the limit on the performance imposed by these characteristics of the problem, we approach the classification problem in a robust way: through the adaptation of the behavior of the classifiers where failure is expected.

To achieve a robust approach to the classification problem, we combine two techniques: classification with context, and classification with rejection. These approaches are based on two central ideas, the exploitation and exploration of structure in the data, through the use of classification with context, and a willingness to improve classification performance at the expense of not classifying the entire data. Based on these two central ideas, we are able to construct classification systems that adapt their behavior when errors are expected. A robust classifier with context and rejection is able to decide, when misclassifications are expected, whether to explore the use of context to solve the classifications, or avoid the classification of a group of samples.

In this thesis, we approach the task of robust image classification through the simultaneous use of context and rejection. We consider and explore three significant challenges:

1. Creation of a framework for computation of classification with context that is flexible enough to include a large variety of concepts of context while associating pixelwise levels of confidence to the context;
2. Derivation of performance measures for the evaluation of classification with rejection;
3. Design of architectures for robust classification with context and rejection that encompass different possible interactions between context and rejection.

1.1 Well-posed and ill-posed classification problems

In this work, we focus on a family of classification problems where at least one of the following characteristics is present:

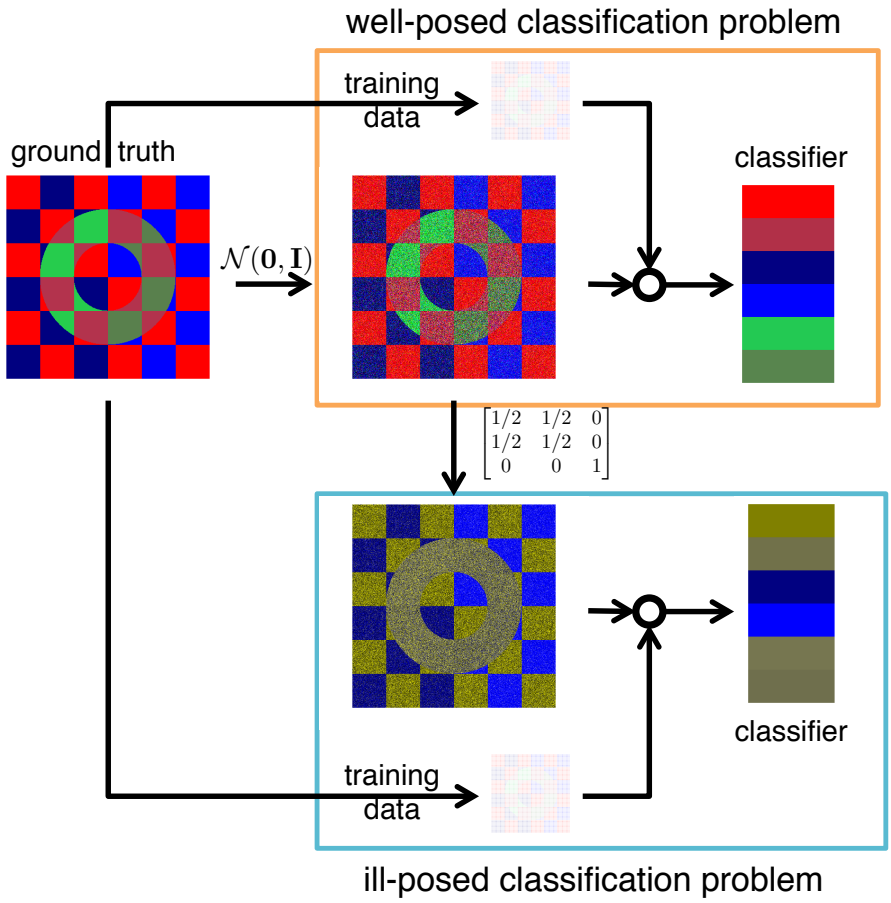


Figure 1.1: Example of well-posed classification (top) *vs.* ill-posed classification problem (bottom).

- Distinct samples with similar feature values exist in different classes;
- Training sets are highly overlapped (little to no separability) or incomplete;
- Samples can belong to unknown classes.

We name this family of problems *ill-posed* classification problems. Conversely, if neither of these characteristics are present in a classification problem, we name it a *well-posed* classification problem.

Fig. 1.1 shows a toy example of a well-posed and an ill-posed classification problem. Whereas the introduction of additive white Gaussian noise to the image does not transform the classification problem into an ill-posed classification problem, as the classifier is still able to discriminate between the different colors, by fading the distinction between red and green, the problem becomes ill-posed. As the distinction between red and green fades, classes that were distinguishable and well separated become indistinguishable. This toy example illustrates the hardness of ill-posed classification problems. As the classes become overlapped, even by increasing the size of the training set, the performance of the classifier stagnates, as it is unable to discriminate the overlapped classes.

Owing to their inability to deal with uncertainty, as classifiers are constrained by the information that they can generalize, general classification systems under-perform in ill-posed classification problems. This means that ill-posed classification problems have to be approached from different angles. If structure can be explored and exploited from the data to provide contextual cues to classification, we are able to apply context to the classification. On the other hand, if we are able to accept an improvement of the performance of the classifiers by allowing the classifier to abstain in situations where it is expected to fail, we are able to apply rejection to the classification. We thus approach the problem of ill-posed classification by combining both classification with context and classification with rejection into a robust classification framework where we have simultaneously the aid of contextual cues for the classification and we are able to selectively abstain when misclassifications are expected.

1.2 Contributions

The core contribution of this thesis is the design of a framework for robust classification built on two pillars: classification with context and classification with rejection. To this extend, we provide contributions to the area of classification with context through the development of the SegSALSA family of algorithms for classification with context. In addition, contributions towards classification with rejection are made through the creation of performance measures for the evaluation and comparison of the performance of classifiers with rejection.

1.2.1 Classification with context

To avoid the discrete nature associated with the context-related optimization algorithms and their inherent combinatorial characteristics, the SegSALSA family of algorithms was developed. Based on the idea of a continuous hidden field driving a discrete labeling and on a marginalization across the discrete labels, we are able to approach the problem of context as a convex

problem. Furthermore, the use of a continuous hidden field, allows for flexible use of priors. Based on this flexibility, in this thesis we present three members of the SegSALSA family of algorithms

- SegSALSA-VTV [1, 2] — using a Vectorial Total Variation (VTV) prior based regularization;
- SegSALSA-STR [3] — using a form of Structure Tensor Regularization (STR) prior based on patch-based Schatten norm minimization;
- SegSALSA-GTV [4] — using a form of Graph-based Total Variation (GTV) prior on a graph derived from the data structure.

1.2.2 Classification with rejection

To evaluate and compare the performance of classifiers with rejection, we present a set of properties that performance measures should hold based on concepts of outperformance that hold for a family of reasonable loss-functions, and derive three performance measures that hold such properties [5]:

- Nonrejected accuracy — measures the performance of the classifier on the subset of samples that are not rejected;
- Classification quality — measures the proportion of correct decisions made by correctly classifying and not rejecting or by rejecting and incorrectly classifying;
- Rejection quality — measures the ability of the rejector to reject misclassified samples.

1.2.3 Robust classification with context and rejection

To integrate classification with context and classification with rejection into a robust classification framework, we present two algorithms for robust classification, using different architectures for the interaction between context and rejection, and different approaches to context.

- The first algorithm is focused on the robust classification of histopathology data with non-representative training sets and unknown classes, based on a graph representation of the structure of the data [6, 7];
- The second algorithm [8–10] is built onto the SegSALSA family of algorithms and allows the use of multiple architectures for the combination of context and rejection, and is applied to hyperspectral image classification.

1.3 Thesis organization

This thesis is divided in four different parts: introduction and required background on classification with rejection and classification with context; robust classification with context and rejection; algorithms for robust classification with context and rejection; and concluding remarks and further work.

- Part I introduces of the robust classification problem in Chapter 1, the background into classification with rejection in Chapter 2, and the background into classification with context in Chapter 3. In Chapter 1, we describe the family of classification problems that motivates us, *ill-posed* classification problems, we outline the steps taken towards achieving improved performance on ill-posed classification problems, and present our contributions. In Chapter 2, we present the background for classification with rejection, starting with an overview of classification with rejection, and illustrate different possible architectures for incorporating rejection onto classification, showing the difficulties associated with the performance evaluation of classification with rejection. In Chapter 3, we provide a brief overview of the use of context in classification, under a Bayesian point of view, explore the use of spatial (local) context in classification, and describe the limitations associated with classification with context resulting from the combinatorial nature of the application of context.
- Part II proposes performance measures for the evaluation of classification systems with rejection in Chapter 4, presents a convex method for the computation of classification with context that sidesteps from the often discrete nature of classification with context in Chapter 5, and introduces multiple architectures for robust classification with context and rejection in Chapter 6. In Chapter 4, we propose a set of properties that a performance measure for classification with rejection should satisfy, and present three measures that allow the quantification of the performance of a classifier with rejection (nonrejected accuracy, classification quality, and rejection quality). In Chapter 5, we propose a family of algorithms (SegSALSA) for classification with context that sidestep from the discrete nature of the optimization problems associated with classification with context, with multiple contextual priors, and illustrate the characteristics of the algorithm on classification with context of different image classification problems. In Chapter 6, we present the general architecture for robust classification with context and rejection, analyze two different instantiations of the architecture through different interactions between context and rejection, and discuss their strengths and weaknesses.
- Part III presents different algorithms for robust classification with context and rejection. In Chapter 7 we present an algorithm for robust classification of histopathology images, with context and rejection. The algorithm takes into account the biological connection between multiple types of tissues, through the concept of super classes, and compute jointly the context and rejection on a graph that represents the multiscale structure of the histopathology image. In Chapter 8 we present a family of algorithms for robust classification with context and rejection, based on the use of SegSALSA for context, with the possibility of joint context and rejection or sequential context and rejection (first context, then rejection), and apply these algorithms to the robust classification of hyperspectral images.
- Part IV closes the thesis with some concluding remarks and indications of further work in the area of robust classification with context and rejection.

Chapter 2

Classification with rejection

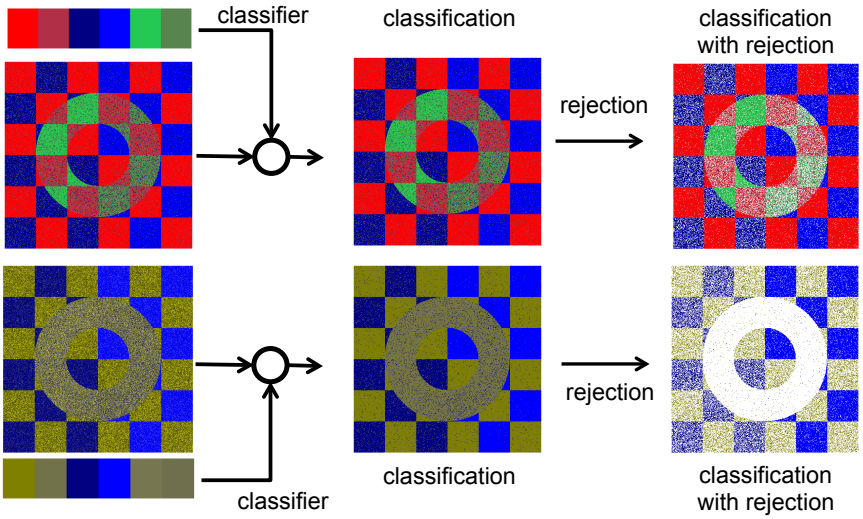


Figure 2.1: Classification with rejection: well-posed companion problem (top) and ill-posed companion problem (bottom). Rejected samples are shown in white.

Classification with rejection is an interesting option in real world applications of machine learning and pattern recognition, when it is possible to improve the performance of classification at the expense of abstaining in difficult classifications where the classifier is more likely to fail. The core idea behind the use classification with rejection is that, through the adaptation of the behavior of the classifier to take in account the confidence associated with each classification, foreseeable errors in the classification can be avoided if some samples are rejected instead of classified, as seen in Fig. 2.1.

In many real world applications of machine learning, we are dealing with problems that are close to ill-posed classification problems. Some classes can be underrepresented in the training data, their features vectors can be overlapped, the training set available can be unbalanced, or we may have unknown classes. In a variety of problems, it is possible to decide not to classify the entire data and thus achieve performance improvements at the expense of the selective classification through the use of classification with rejection: from automated medical diagnosis [6, 11],

to landcover classification [8,9,12], biometrics [13], to image retrieval [14] and scene classification [15,16]. A classifier with rejection can also cope with unknown information, reducing the threat posed by the existence of samples belonging to unknown classes or mislabeled training samples that could harm the performance of the classifier.

The concept of classification with rejection is built upon two simple mechanisms:

- Classification confidence — an implicit ordering of the samples according to their potential to be rejected;
- Rejected amount — a threshold that is able to control the amount of samples that are rejected.

These two mechanisms are the keystones for any form of classification with rejection. Even though the concept of classification confidence can be hidden in the classification process, to achieve classification with rejection, it must be possible to compare classified samples according to their expected potential to be misclassified.

2.1 Overview of classification with rejection

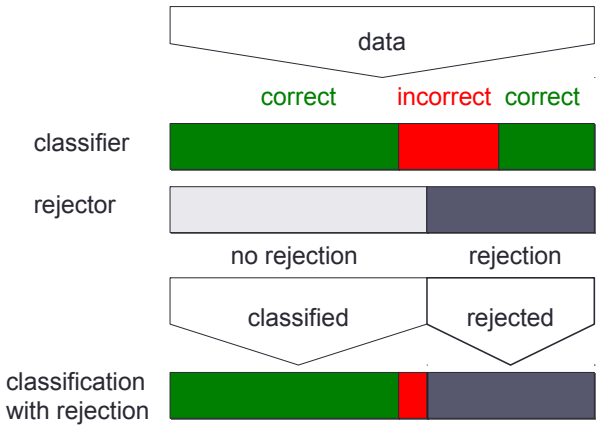


Figure 2.2: General system for classification with rejection.

The central idea of classification with rejection is to associate to the classification its reliability or confidence. Samples that can be accurately classified with a high degree of confidence should be classified, whereas samples that cannot be accurately classified with a high degree of confidence, or samples that will be misclassified with a high degree of confidence, should not be classified and should be handled exceptionally, as schematized in Fig. 2.2.

2.1.1 Rejection formulation

Classification with rejection was first analyzed in [17,18], where Chow's rule for optimum error-reject threshold was presented. In a binary classification setting, Chow's rule allows for the determination of a threshold for rejection such that the classification risk is minimized. Let us

consider a binary $(0, 1)$ classification problem where the ith sample is classified according to the posterior probabilities

$$\widehat{\mathbf{y}}_i = \begin{cases} 0 & , \text{ if } \quad p(\widehat{\mathbf{y}}_i = 0|\mathbf{x}) \geq 1/2, \\ 1 & , \text{ if } \quad p(\widehat{\mathbf{y}}_i = 1|\mathbf{x}) > 1/2, \end{cases} \tag{2.1}$$

where $p(\widehat{\mathbf{y}}_i = K|\mathbf{x})$ is the posterior probability of the ith sample belonging to the class K given the feature vector $\mathbf{x}$.

With the knowledge of the posterior probabilities, the optimal Chow's rule allows for the determination of a threshold t such that

$$\widehat{\mathbf{y}}_i = \begin{cases} 0 & , \text{ if } \quad p(\widehat{\mathbf{y}}_i = 0|\mathbf{x}) \geq 1 - t, \\ 1 & , \text{ if } \quad p(\widehat{\mathbf{y}}_i = 1|\mathbf{x}) > 1 - t, \\ \text{reject} & , \text{ otherwise.} \end{cases} \tag{2.2}$$

The determination of the threshold t is dependent of the existence of a cost function (loss function). Let W_M, W_R and W_A denote the costs for misclassification, rejection, and accurate classification, respectively, then Chow's rule states that the optimal error-rejection tradeoff is given by

$$t = \frac{W_R - W_A}{W_M - W_A}.$$

It is clear that, when we have $t > 1/2$ in (2.2), we are in a situation where we have no rejection, and the problem collapses into a problem of classification without rejection (2.1).

Chow's rule for optimum error-rejection threshold is based on two assumptions:

- Perfect knowledge of the posterior probabilities;
- Existence of a cost function that specifies the cost of misclassification and the cost of rejection.

As pointed in [19–21], the assumption of perfect knowledge of the posterior probabilities is not feasible in the real-world, where posterior probabilities might not be estimated without error. On the other hand, the design of problem specific cost functions can be unfeasible for each possible problem of classification with rejection.

As Chow's rule only provides the optimal error-rejection threshold if the posterior probabilities are exactly known, a combination of multiple class-related thresholds is proposed in [19]. The multiple thresholds are obtained through a parameter selection process that results from the constrained maximization of an heuristic performance metric: the classification accuracy subject to upper bounds on the rejection rate (fraction of samples rejected).

In [20, 22], the problem of the design of the rejection rule for binary $(0, 1)$ classification is approached under a cost minimization of receiver operating characteristic (ROC) curve. This leads to the minimization of the expected cost incurred by the classifier with rejection, whereas Chow's rule leads to a minimization of the error rate for a given rejection rate (proportion of the data rejected). On a situation where the posterior probabilities have to be estimated, the use of empirical ROC curves to design the rejection rule can achieve better performance than Chow's rule [23].

2.2 Architectures for classification with rejection

Following the categorization in [24], we will focus on two different architectures for classification with rejection:

- Plug-in rejection — rejection as a secondary sequential binary classifier;
- Embedded rejection — rejection as a risk minimization process through the use of hinge-like loss functions in the classifier.

2.2.1 Plug-in rejection

Combining multiple classifiers

In addition to the seminal work on classification with rejection [18], which is based on the use of a plug-in rule, we will look into plug-in rules that extend classification with rejection to a multi-class approach.

In [25], rejection is the result of a threshold on the reliability of a Bayesian combining rule to aggregate the result of multi-expert systems. More recently, a framework for multilabel classification with rejection was introduced in [26] with a combination of multiple class-specific contingency tables that result from class-specific rejection thresholds.

In [21, 27], rejection is brought onto a multi-class setting through the combination of multiple single-class classifiers. To mitigate the difficulties associated with a large spread within the classes, leading to poor results with density-based methods, an heuristic approach is proposed based on the combination of density-based models with distance-based models. Through a process of output normalization of each single-class classifier, a normalization both on samples that should be rejected (referred as outliers) and on samples that should not be rejected (referred as targets), a class specific rejection threshold is proposed. This approach is also explored in [28], where the performance of sequential one-class classifiers is compared to the performance of multi-class classifiers on ill-defined classification problems.

Parallel

A different route to classification with rejection is taken in [29], where rejection is not the result of a secondary sequential binary classifier, but the result of a secondary *parallel* binary classifier. The central idea of this approach is that the rejection is trained in parallel with the main classifier to assess the reliability of the main classifier, thus providing a reject option. This results in a more complex training process.

2.2.2 Embedded rejection: structural risk minimization

The key idea of embedding rejection in a classifier is the minimization of a risk that takes in account both the misclassification cost and the rejection cost. This idea is present in [30], where structural risk minimization is achieved by the use of a LASSO-type penalty, through the use of a surrogate convex hinge loss function.

A significant trend in classification with rejection has been the incorporation of rejection in the training stage of Support Vector Machines (SVM). An example is the use of SVM with a rejection rule inspired on the use of a ROC curve and on the minimization of the misclassification and rejection costs incurred, proposed in [31]. Another approach is the embedding of the reject option in the SVM formulation in close association with the separating hyperplane resulting from the formulation [32, 33]. This results on a noncovex problem which is solved by finding a surrogate loss function [34]. The statistical properties of the surrogate loss functions, and their application to classification with rejection, have been extensively studied in [30, 35, 36].

2.3 Existing gaps

Whereas the use of rejection in classification provides significant levels of robustness to automated classification systems, we consider that there is a significant gap in the evaluation and comparison of classification systems with rejection.

2.3.1 Performance measures

There is no standard measure for the assessment of the performance of a classifier with rejection. Accuracy-rejection curves, used in [12, 19, 33, 37, 38], and their variants based on the analysis of the F_1 score, used in [26, 39], albeit popular in practical applications of classification with rejection have significant drawbacks. Obtaining sufficient points for an accuracy rejection curve might not be feasible for a classifier with embedded reject option, which requires retraining the classifier to achieve a different rejection ratio, or for classifiers that combine contextual information with rejection, where changes in the amount of rejection require a recomputation of the context. This means that accuracy-rejection curves and the F_1 rejection curves, in the real world, are not able to describe the behavior of the classifier with rejection in all cases.

In [40], a different approach is taken, a 3D ROC plot of a 2D ROC surface is obtained by decomposing the false positive rate into false positive rate for outliers belonging to known classes and false positive rate for outliers belonging to unknown classes, with the volume under the curve as the performance measure. The use of ROC curves for the analysis of the performance suffers from the same problems associated with accuracy-rejection curves.

2.4 Concluding remarks

The association of rejections to classifiers provides a significant degree of resilience to the classification systems. Through a process of selective abstention, predictable misclassifications can be avoided. This leads to classification systems that can be made impervious to imperfect knowledge.

Chapter 3

Classification with context

3.1 Introduction

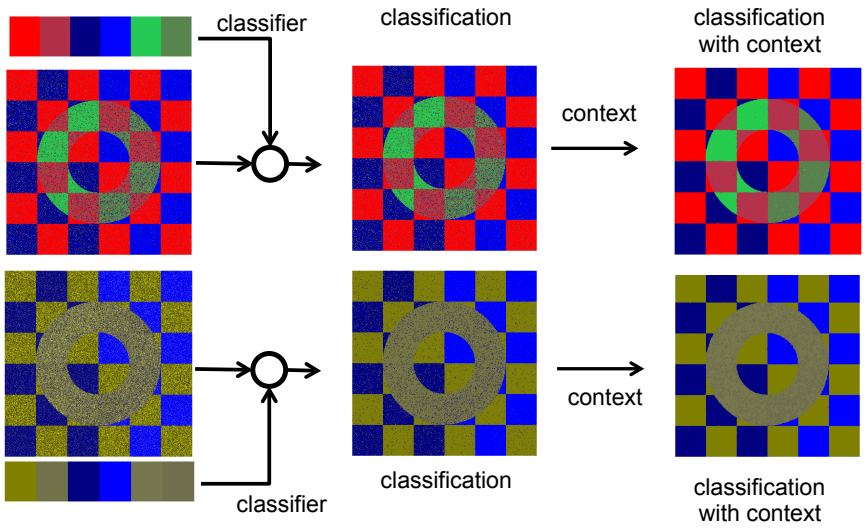


Figure 3.1: Classification with context: well-posed companion problem (top) and ill-posed companion problem (bottom).

The use of classification with context is central in image segmentation and image classification techniques, where contextual cues aid the classification process, as seen in Fig. 3.1. The application of context to classification is often performed through the formulation of an energy minimization problem [41, 42]. We consider as a key formulation for the use of context in classification the following energy minimization problem

$$\mathbf{y}_C \in \arg\min_{\mathbf{y}\in\mathcal{L}^n} E_d(\mathbf{y}) + E_s(\mathbf{y}), \tag{3.1}$$

where $\mathbf{y}_C$ denotes a classification with context, $\mathcal{L}^n$ is the space of possible labelings, $\mathbf{y}$ is a labeling, E_d is a data term that is associated with the classification (class probabilities), and E_s

is a regularizer that imposes some contextual characteristics on the classification (*e.g.* through the promotion of smooth solutions). In Fig. 3.1, we see the difference between the optimization on the data term alone E_d, resulting in the classification in the middle, and the optimization on both the data term E_d associated with the classification and the regularizer E_s associated with the promotion of contextual characteristics, resulting in the classification with context on the right.

3.2 Overview of classification with context

The central idea of classification with context is to harness the existence of prior information associated with the problem at hand, and use it as an aid to solve the problem. The use of context in classification is widespread in image segmentation and image classification tasks. From image restoration [43] and texture modeling [44], to early vision [45] and stereo matching [46–48]. It is also widely used for segmentation based tasks, such as interactive segmentation [49, 50] and photo and video editing [51]. More recently, classification with context has lead to significant improvements in the performance of classification systems in hyperspectral image classification tasks, such as the elaboration of thematic maps [52].

3.3 Maximum a posteriori setting

We denote all matrices by bold upper-case letters, and all vectors by bold lower-case letters. Let $\mathbf{x} \in \mathbb{R}^{d \times n}$ denote a n-pixel d-dimensional feature image, such that $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector corresponding to the ith pixel. Let $\mathcal{S} = \{1, \ldots, n\}$ be the set indexing the image pixels, $\mathcal{L} = \{1, \ldots, K\}$ the set of possible K labels of the image, and $\mathbf{y} \in \mathcal{L}^n$ be a labeling (segmentation) of the image.

The segmentation problem, in a Bayesian framework, can be approached through the maximum *a posteriori* (MAP) segmentation of the image

$$\widehat{\mathbf{y}} \in \arg\max_{\mathbf{y} \in \mathcal{L}^n} p(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{L}^n} p(\mathbf{x}|\mathbf{y}) p(\mathbf{y}), \tag{3.2}$$

where $p(\mathbf{y}|\mathbf{x})$ denotes the posterior probability of the labeling $\mathbf{y}$ given the feature image $\mathbf{x}$, $p(\mathbf{x}|\mathbf{y})$ the observation model, and $p(\mathbf{y})$ the prior probability of the labeling.

A usual assumption in low level image segmentation [43] is that of conditional independence of the features given the labels. We thus have that the observation model can be represented as

$$p(\mathbf{x}|\mathbf{y}) = \prod_{i \in \mathcal{S}} p(\mathbf{x}_i|\mathbf{y}_i).$$

The rewriting of the observation model on each pixel,

$$p(\mathbf{x}_i|\mathbf{y}_i) = p(\mathbf{y}_i|\mathbf{x}_i) \frac{p(\mathbf{x}_i)}{p(\mathbf{y}_i)},$$

combined with the conditional independence of the observation model, yields, the following discriminative class model

$$p(\mathbf{x}|\mathbf{y}) = \prod_{i \in \mathcal{S}} p(\mathbf{y}_i|\mathbf{x}_i) \frac{p(\mathbf{x}_i)}{p(\mathbf{y}_i)}.$$

Under assumption of equiprobability of both the features and the class labels, we have thus that

$$p(\mathbf{x}|\mathbf{y}) \propto \prod_{i \in \mathcal{S}} p(\mathbf{y}_i|\mathbf{x}_i). \tag{3.3}$$

This shift from a generative model to a discriminative model is frequently associated with less complex discriminative models that tend to achieve better performance on smaller training sets than their generative counterparts [53]. For simplicity of notation, we often represent the class probabilities on the ith pixel in a vectorized form as

$$\boldsymbol{p}_i = \left[p(\mathbf{y}_i = 1|\mathbf{x}_i), \ldots, p(\mathbf{y}_i = K|\mathbf{x}_i)\right].$$

Combining the MAP formulation in (3.2) with the pixelwise discriminative models for the class labels obtained from (3.3), we reformulate the MAP problem as

$$\begin{aligned} \widehat{\mathbf{y}} &\in \arg\max_{\mathbf{y} \in \mathcal{L}^n} \left(\prod_{i \in \mathcal{S}} p(\mathbf{y}_i|\mathbf{x}_i) \right) p(\mathbf{y}) \\ &= \arg\min_{\mathbf{y} \in \mathcal{L}^n} \left(\sum_{i \in \mathcal{S}} -\log(p(\mathbf{y}_i|\mathbf{x}_i)) \right) - \log(p(\mathbf{y})). \end{aligned} \tag{3.4}$$

3.4 Energy minimization

The problem (3.4) can be approached through the use of graphical models, such as Markov Random Fields (MRF) [43,54,55] based models. MRF models are usually generative in their nature, modeling the joint probability of the image features $\mathbf{x}$ and of the image labels $\mathbf{y}$. Whereas in binary problems the prior associated with the MRF is an Ising model (often isotropic), in multilabel problems we have a Potts model [56] associated with the prior [55]. Formulating a problem of classification with context through a MAP estimation with a MRF prior allows the reformulation of the problem as an energy minimization problem [57]. The MRF prior serves as basis for the Conditional Random Fields (CRF) [41] modeling, where the posterior probability $p(\mathbf{y}|\mathbf{x})$ is modeled directly as a Gibbs field. This formulation is further extended in the Discriminative Random Fields (DRF) [42, 58], where the assumption of conditional independence, prevalent in the MRF formulation, is discarded.

The posterior in (3.4) is a particular case of the DRF formulation, where

$$\begin{aligned} \widehat{\mathbf{y}} &\in \arg\min_{\mathbf{y} \in \mathcal{L}^n} \overbrace{\left(\sum_{i \in \mathcal{S}} -\log(p(\mathbf{y}_i|\mathbf{x}_i)) \right)}^{E_d(\mathbf{y})} \overbrace{-\log(p(\mathbf{y}))}^{E_s(\mathbf{y})} \\ &= \arg\min_{\mathbf{y} \in \mathcal{L}^n} E_d(\mathbf{y}) + E_s(\mathbf{y}). \end{aligned} \tag{3.5}$$

In the DRF formulation, E_d corresponds to the association potential and E_s to the interaction potential.

As shown in [47], the optimization (3.5) is NP-hard except for binary problems with submodular interactions, where it can be mapped into a max-flow problem and an optimal solution can be found in polynomial time. To mitigate the NP-hardness associated with the computation of context, either approximated solutions to the problem can be found, or a convex relaxation to the context approach is applied.

3.5 Approximate solutions to classification with context

Classification with context can be achieved by approaching (3.5) in an approximate way. In this section, we look into graph-cuts based approaches to solve the problem of classification with context. The problem of classification with context can also be solved using message passing algorithms, such as loopy belief propagation (LBP) [45, 59–61], and tree-reweighted message passing (TRWS) [62, 63]. A more detailed description of approximation methods to solve the problem of classification with context can be found in [64, 65] and performance comparisons between message passing methods and graph-cuts based methods can be found in [66–68].

3.5.1 Graph-cuts

The graph-cuts algorithm [47, 51, 69] is one of the most important tools available for including context in the classification through the use of energy minimization approaches. We will discuss the two well-known variants of graph-cuts algorithms: expansion move (or α-expansion) algorithm, and the swap move (or $\alpha - \beta$-swap) algorithm, both introduced in [47]. The core mechanic of both graph-cuts algorithms is based on the sequential computation of global solutions for binary labeling problems. The two algorithms differ in the possible actions on each of the binary labeling problems (through different definitions of the move-space): local minimum through the computation of successive global minima allowing only expansion moves on each label; and local minimum through the computation of successive global minimal allowing only swap moves on each pair of labels.

Graph-cuts: expansion move

An expansion move is defined, given a subset of pixels with the label α, as a move that expands the subset of pixels with the label α, as illustrated in Fig 3.2, such that the energy function is minimized on the binary problem. The expansion move algorithm works by finding a local

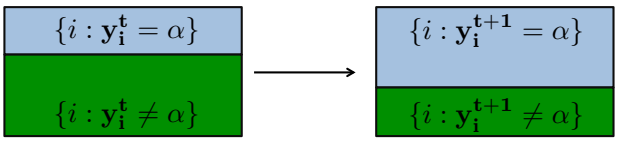


Figure 3.2: Graph-cuts, example of an expansion move.

minimum such that no expansion move, for any possible label $\alpha \in \mathcal{L}$, results in a labeling with lower energy.

The expansion move algorithm requires the smoothness term in (3.1) to be a sum of metric functions. This means that for the smoothness term

$$E_s(\mathbf{y}) \propto \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} V_{i,j}(\mathbf{y}_i, \mathbf{y}_j),$$

where $\mathcal{N}_i$ denotes the neighboring pixels of the ith pixel, $V_{i,j}$ is metric, this is

$$\begin{aligned} V_{i,j}(\alpha, \beta) = 0 &\iff \alpha = \beta, \\ V_{i,j}(\alpha, \beta) &= V_{i,j}(\beta, \alpha), \\ V_{i,j}(\alpha, \beta) &\leq V_{i,j}(\alpha, \gamma) + V_{i,j}(\gamma, \beta), \end{aligned}$$

for any $\alpha, \beta, \gamma \in \mathcal{L}$.

Graph-cuts: swap move

A swap move is defined, given a subset of pixels with the label α and a subset of pixels with the label β, as a move that swaps pixels between these two subsets, as seen in Fig 3.3, such that the energy function is minimized in the binary problem. The swap move algorithm works by finding

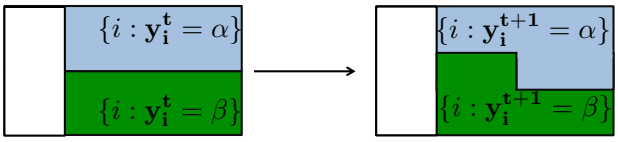


Figure 3.3: Graph-cuts, example of a swap move.

a local minimum such that no swap move, for any possible pair of labels $\alpha \in \mathcal{L}$ and $\beta \in \mathcal{L}$, results in a labeling with lower energy.

The swap move algorithm requires the smoothness term in (3.1) to be a sum of semimetric functions. This means that for the smoothness term

$$E_s(\mathbf{y}) \propto \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} V_{i,j}(y_i, y_j),$$

where $\mathcal{N}_i$ denotes the neighboring pixels of the ith pixel, then $V_{i,j}$ is semimetric, this is

$$\begin{aligned} V_{i,j}(\alpha, \beta) = 0 &\iff \alpha = \beta, \\ V_{i,j}(\alpha, \beta) &= V_{i,j}(\beta, \alpha), \end{aligned}$$

for any $\alpha, \beta, \in \mathcal{L}$.

3.6 Convex relaxations

By approaching the problem in a convex relaxation setting, the energy minimization problem in (3.1) can be transformed into

$$\begin{aligned} \arg \min_{\boldsymbol{u} \in [0,1]^{K \times n}} & E_d(\boldsymbol{u}) + E_s(\boldsymbol{u}), \\ \text{subject to: } & \mathbf{1}^T \boldsymbol{u} = \mathbf{1}, \\ & \boldsymbol{u} \geq \mathbf{0} \end{aligned} \tag{3.6}$$

resulting from a convex relaxation of the integer optimization problem associated with a discrete search space $\boldsymbol{u} \in \{0,1\}^{K \times n}$ into the set $[0,1]^{K \times n}$. In this formulation, the continuous variable $\boldsymbol{u}$ is seen as an indicator function, where $\boldsymbol{u}_{i,k} = 1$ means that the ith image pixel belongs to the kth class. Both [70] and [71] approach this problem through a dual formulation of a total variation term associated with E_s, differing in the definition of the constraint set of the dual variables. In [72], a thorough comparison between convex relaxation methods and discrete approximation methods is available.

The resulting minimizer in (3.6) is then binarized, thus providing a labeling. This binarized solution can be shown to be within a factor of the optimal solution of the discrete problem based on the energy difference between the binarized and nonbinarized convex solution [73], and to have an expected value that is within a factor of 2 of the optimal solution of the discrete problem [74].

3.7 Existing gaps

Whereas classification with context provides significant performance improvements to image classification, its formulation as a discrete optimization faces hurdles linked with the underlying discrete optimization problems and restricts prior selection. In fact, the prior is often selected because of the existence of efficient optimization methods for that prior.

3.8 Concluding remarks

Algorithms for computing classification with context are the key-stones of any robust classification system that combines context with rejection. Both the performance improvements associated with using rejection, and the computational complexity inherent to the application of context to classification, lead to a need of fast methods for the computation of context. However, increasing in the speed of the computation of context comes at the expense of approximating the problem, either by approximation methods or through the use of convex relaxations. Furthermore, the existing methods for fast computation of context often restrict the choice of prior information to the use of Potts-based models.

Part II

Classification with context and rejection

To create robust classification systems, we require not only methods to compute classification with context and classification with rejection, but also a framework to integrate them together. However, before we integrate classification with rejection and classification with context, there are gaps that need to be addressed both in classification with rejection and in classification with context. Here, we identify these gaps and lay the foundations for our contributions towards robust classification with context and rejection.

Existing gaps

Classification with rejection

Classification systems with rejection are resilient against ill-posed classification problems. By allowing the classifier to selectively abstain from classification in situations where misclassifications are expected, better performance can be achieved.

Despite the existence of significant work on classification systems with rejection and on the design of the rejection systems, the comparison between classification systems is still nontrivial. Comparing the performance of different classifiers with rejection is often limited by the existence of well-defined cost functions (which might not be available for every possible classification problem), and to trivial comparisons when the amount of rejection is the same.

Furthermore, even though classification with rejection provides a significant level of robustness to the classifiers through a process of selective abstention when misclassifications are expected, there is no framework to integrate the use of contextual cues to help the classification.

Classification with context

Classification systems with context are widely used in image classification and segmentation, as they allow significant performance improvements in tasks where context plays an important role. However, approaching the task of classification with context results in an integer optimization problem which, as discussed in Chapter 3, is often a NP-hard problem and unsolvable in feasible time.

Problem formulation

In order to design robust classification systems capable of dealing with nonideal classification situations, as the ill-posed classification problems described in Chapter 1, we need to address the existing gaps both in classification with rejection and in classification with context. To this extent, we will focus on the following points:

- Evaluation of performance for classification with rejection in Chapter 4;
- Formulation of classification with context as a convex problem using hidden fields in Chapter 5;
- Design of architectures for combining classification with rejection and classification with context into robust classification frameworks in Chapter 6.

Chapter 4

Performance measures

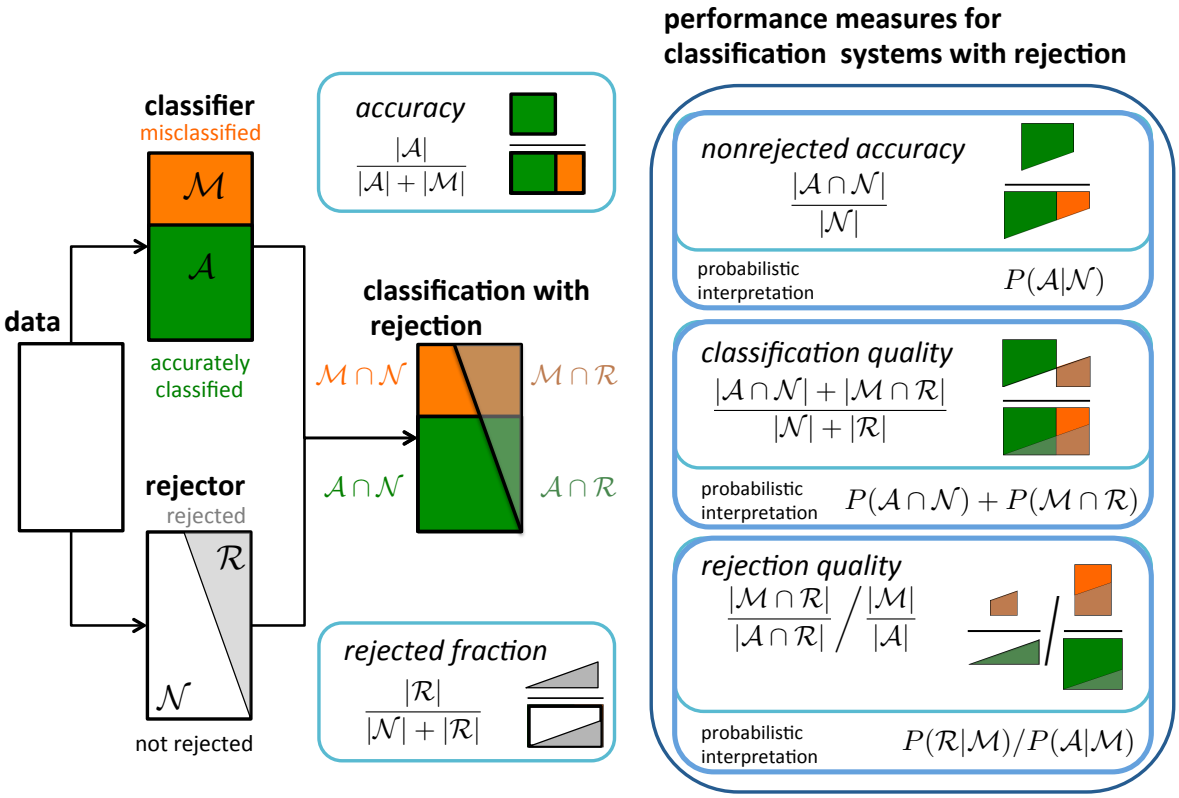


Figure 4.1: Graphical overview of performance measures for classification with rejection.

4.1 Introduction

As seen in Chapter 2, through the use of classification with rejection we are able to equip automatic classification systems with a significant degree of robustness. Unknown classes and high degrees of class overlap can be exceptionally handled as the classifier can abstain from hard classifications. However, we have seen that the performance measures currently used, not only are not standardized, but require many recomputations of classification with rejection at different rejection rates, which might not be feasible in some applications.

To fill the gap associated with the lack of performance measures for the evaluation of classification systems with rejection, we propose a set of properties for performance measures for classifiers with rejection, and a set of three performance measures that satisfy those properties.

Our starting point for the creation of reasonable performance measures is the definition of properties that such performance measures should hold for the evaluation classification systems with rejection, and consequently the evaluation of the the performance of the rejectors (or rejection mechanisms). We consider that the four following properties are necessary and sufficient for the definition of a performance measure:

- Property I — be a function of the fraction of rejected samples;
- Property II — be able to compare different rejection mechanisms working at the same fraction of rejected samples;
- Property III — be able to compare rejection mechanisms working at a different fractions of rejected samples when one rejection mechanism outperforms the other;
- Property IV — be maximum for a rejection mechanism that no other feasible rejection mechanism outperforms, and minimum for a rejection mechanism that all other feasible rejection mechanisms outperform.

These properties are based on the concept of outperformance and on the ability to state whether one rejection mechanism qualitatively outperforms the other. If a cost function exists that takes in account the cost of rejection and misclassification, the concept of outperformance is trivial, and this cost function not only satisfies the properties but is also the ideal performance measures for the problem in hand. However, as stated in Chapter 2, it might not be feasible to design a cost function for each individual classification problem. Thus, we derive a set of cases where the concept of outperformance holds for any cost function, provided that the cost of rejection is never greater than the cost of misclassification.

With the properties and the concept of outperformance in place, the following performance measures, illustrated in Fig. 4.1, satisfy the above stated properties:

- ***Nonrejected accuracy*** measures the ability of the classifier to accurately classify nonrejected samples (probability of a sample being accurately classified given that is was not rejected);
- ***Classification quality*** measures the ability of the classifier with rejection to accurately classify nonrejected samples and to reject misclassified samples (probability of a sample being correctly classified and not rejected or being misclassified and rejected);
- ***Rejection quality*** measures the ability of the classifier with rejection to make errors on rejected samples only (ratio between probability of a sample being misclassified given that

was rejected and the probability of a sample being misclassified).

With the three measures in place, we explore the best and worst case scenarios for each measure, for a given reference classifier with rejection. We denote the proximity of a classifier with rejection to its best and worst case scenarios, with regard to a reference classifier with rejection, as *relative optimality*. This allows us to easily connect performance measures to problem specific cost functions. For a classifier with rejection that rejects at two different numbers of rejected samples, the relative optimality defines the families of cost functions on which rejection at one number rejected samples is better, equal, or worse than rejection at the other number of rejected samples.

4.1.1 Notation

Following a plug-in approach, as described in Chapter 2, a classifier with rejection can be seen as a coupling of a classifier C with a rejection system R. The classification maps n d-dimensional feature vectors $\mathbf{x}$ into n labels $C : \mathbb{R}^{d \times n} \to \{1, \ldots, K\}^n$, such that

$$\widehat{\mathbf{y}} = C(\mathbf{x}),$$

where $\widehat{\mathbf{y}}$ denotes a labeling. The rejector R maps the classification (feature vectors $\mathbf{x}$ and associated labels $\widehat{\mathbf{y}} = C(\mathbf{x})$) into a binary rejection vector, $R : \mathbb{R}^{d \times n} \times \{1, \ldots, K\}^n \to \{0, 1\}^n$, such that

$$\boldsymbol{r} = R(\mathbf{x}, \widehat{\mathbf{y}}),$$

where $\boldsymbol{r}$ denotes the binary rejection vector. We define a classification with rejection $\widehat{\mathbf{y}}^R$ as

$$\widehat{\mathbf{y}}_i^R = \begin{cases} \widehat{\mathbf{y}}_i, & \text{if } \boldsymbol{r}_i = 0, \\ K + 1, & \text{if } \boldsymbol{r}_i = 1, \end{cases}$$

where $\widehat{\mathbf{y}}_i$ corresponds to the classification of the ith sample, $\boldsymbol{r}_i$ corresponds to the binary decision to reject ($r_i = 1$) or not ($r_i = 0$) the ith sample, and $\widehat{\mathbf{y}}_i^R = K + 1$ denotes rejection of the ith sample.

By comparing the classification $\widehat{\mathbf{y}}$ with its ground truth $\mathbf{y}$, we form a binary n-dimensional accuracy vector $\boldsymbol{a}$, such that $\boldsymbol{a}_i$ measures whether the ith sample is accurately classified or misclassified, respectively $\boldsymbol{a}_i = 1$ and $\boldsymbol{a}_i = 0$. The binary vector $\boldsymbol{a}$ imposes a partition of the set of samples in two subsets $\mathcal{A}$ and $\mathcal{M}$, namely the subset of accurately classified samples and the subset of misclassified samples. Let $\boldsymbol{c}$ be a confidence vector associated with the classification $\widehat{\mathbf{y}}$, such that

$$\boldsymbol{c}_i \geq \boldsymbol{c}_j \implies \boldsymbol{r}_i \leq \boldsymbol{r}_j,$$

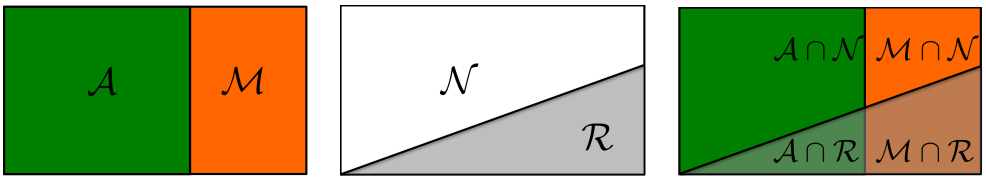
which implies that if sample i is rejected, then all the samples j with smaller confidence $\boldsymbol{c}_j < \boldsymbol{c}_i$ are also rejected. We have then the ground truth $\mathbf{y}$, the result of the classification $\widehat{\mathbf{y}}$, and the result of the classification with rejection $\widehat{\mathbf{y}}^R$.

Let $\boldsymbol{c}\downarrow$ denote the reordering of the confidence vector $\boldsymbol{c}$ in decreasing order. If we keep the k samples with the highest confidence and reject the rest $n - k$ samples, we obtain two subsets: k

nonrejected samples and $n-k$ rejected samples, $\mathcal{N}$ and $\mathcal{R}$[1] respectively. Our goal is to separate the accuracy vector $\boldsymbol{a}$ into two subvectors ($\boldsymbol{a}_{\mathcal{N}}$ and $\boldsymbol{a}_{\mathcal{R}}$), based on the confidence vector $\boldsymbol{c}$ such that all misclassifications are in the $\boldsymbol{a}_{\mathcal{R}}$ subvector, and all accurate classifications are in the $\boldsymbol{a}_{\mathcal{N}}$ subvector. We should note that, since $\mathcal{N}$ and $\mathcal{R}$ have disjoint supports,

$$\|\boldsymbol{a}\|_0 = \|\boldsymbol{a}_{\mathcal{N}}\|_0 + \|\boldsymbol{a}_{\mathcal{R}}\|_0, \tag{4.1}$$

for all $\mathcal{N}, \mathcal{R}$ such that $\mathcal{N} \cap \mathcal{R} = \emptyset$ and $\mathcal{N} \cup \mathcal{R} = \{1, \ldots, n\}$, meaning that the number of accurately classified samples $\|\boldsymbol{a}\|_0$ is equal to the sum of the number of accurately classified samples not rejected $\|\boldsymbol{a}_{\mathcal{N}}\|_0$ with the number of accurately classified samples rejected $\|\boldsymbol{a}_{\mathcal{R}}\|_0$. As we only work with the norm of binary vectors, we point that $\|\boldsymbol{a}\|_0 = \|\boldsymbol{a}\|_1$; for simplicity, we omit the subscript.



(a) (b) (c)

Figure 4.2: Partition of the sample space based on the performance of (a) classification only (partition space $\mathcal{A}$ and $\mathcal{M}$); (b) rejection only (partition space $\mathcal{R}$ and $\mathcal{N}$); and (c) classification with rejection. Green corresponds to accurately classified samples and orange to misclassified samples. Gray corresponds to rejected samples and white to nonrejected samples.

With the partitioning of the sample space into $\mathcal{A}$ and $\mathcal{M}$ according to the values of the binary vector $\boldsymbol{a}$, and the partitioning of the sample space into $\mathcal{N}$ and $\mathcal{R}$, we can thus partition the sample space as in Fig. 4.2:

- $\mathcal{A} \cap \mathcal{N}$: samples **a**ccurately classified and **n**ot rejected; the number of such samples is $|\mathcal{A} \cap \mathcal{N}| = \|\boldsymbol{a}_{\mathcal{N}}\|$
- $\mathcal{M} \cap \mathcal{N}$: samples **m**isclassified and **n**ot rejected; the number of such samples is $|\mathcal{M} \cap \mathcal{N}| = \|\mathbf{1} - \boldsymbol{a}_{\mathcal{N}}\|$
- $\mathcal{A} \cap \mathcal{R}$: samples **a**ccurately classified and **r**ejected; the number of such samples is $|\mathcal{A} \cap \mathcal{R}| = \|\boldsymbol{a}_{\mathcal{R}}\|$
- $\mathcal{M} \cap \mathcal{R}$: samples **m**isclassified and **r**ejected; the number of such samples is $|\mathcal{M} \cap \mathcal{R}| = \|\mathbf{1} - \boldsymbol{a}_{\mathcal{R}}\|$

4.2 Comparing classifiers with rejection

The comparison of the performance of two rejectors is nontrivial. It depends on the existence of a problem specific cost function that takes in account the trade-off between misclassification

[1]We note that R corresponds to a rejector, a function that maps classification into a binary rejection vector, whereas $\mathcal{R}$ denotes a set of samples that are rejected.

and rejection. If a cost function exists, the performance is linked to the comparison of the cost function evaluated on each rejector. However, as previously stated, the design of a problem specific cost function might not be feasible. Let ρ denote the trade-off between rejection and misclassification, thus defining a family of cost functions where a misclassification has a unitary cost and a rejection has a cost of ρ. This can be seen as a normalization of the cost function presented in Chapter 2, where W_A is set to 0, W_R is set to ρ, and W_M is set to 1. This family of cost functions can be expressed as,

$$|\mathcal{M} \cap \mathcal{N}| + \rho|\mathcal{R}| =$$  .

In a probabilistic interpretation, this cost function corresponds to an extended classification risk with cost $P(\mathcal{M} \cap \mathcal{N}) + \rho P(\mathcal{R})$.

There are three general cases where it is possible to perform comparisons between the performance of two rejectors for any loss function, independently of ρ:

- When the number of rejected samples is the same;
- When the number of accurately classified samples not rejected is the same;
- When the number of misclassified samples not rejected is the same.

This is true for all values of ρ, if we assume that $0 \leq \rho \leq 1$, which is a reasonable assumption, as $\rho < 0$ would lead to a rejection only problem (all samples rejected), and $\rho > 1$ would lead to a classification only problem (no samples are rejected). Let C denote a classifier with an accuracy vector $\boldsymbol{a}$, and R_1 and R_2 denote two different rejection mechanisms that partition the sample space in $\mathcal{N}_{R_1}, \mathcal{R}_{R_1}$ and $\mathcal{N}_{R_2}, \mathcal{R}_{R_2}$ respectively.

In the following cases we consider the most general concept of outperformance possible, when the cost function of R_1 is smaller than the cost function of R_2 for all values of ρ, such that $0 \leq \rho \leq 1$.

Equal number of rejected samples

If both rejectors reject the same number of samples, and if rejector R_1 has a larger number of accurately classified samples than R_2, then R_1 outperforms R_2. The rejector R_1 outperforms R_2

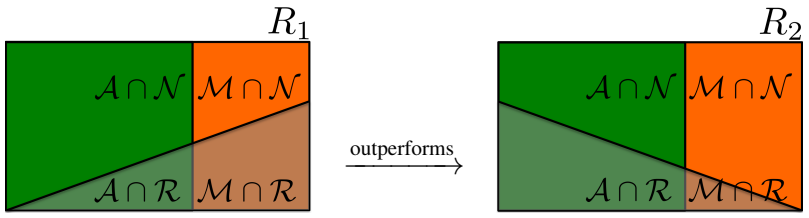


Figure 4.3: R_1 outperforms R_2 with equal number of rejected samples

when, for the same number of rejected samples

$$|\mathcal{R}_{R_1}| = |\mathcal{R}_{R_2}|,$$

R_1 rejects more misclassified samples than R_2 and, as a consequence, R_1 rejects less accurately classified samples than R_2,

$$\|a_{\mathcal{N}_{R_1}}\| > \|a_{\mathcal{N}_{R_2}}\|.$$

Equal number of nonrejected accurately classified samples

If both rejectors have the same number of accurately classified samples not rejected, and if rejector R_1 rejects more samples than R_2, then R_1 outperforms R_2. The rejector R_1 outperforms R_2

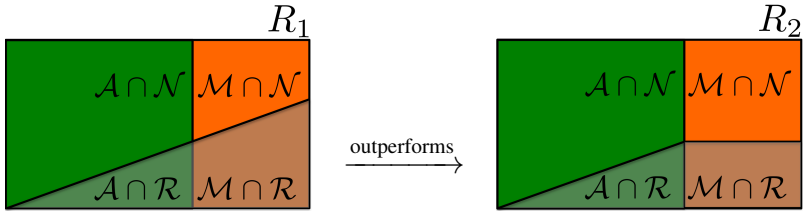


Figure 4.4: R_1 outperforms R_2 with equal number of nonrejected accurately classified samples

when, for the same number of accurately classified samples not rejected

$$\|a_{\mathcal{N}_{R_1}}\| = \|a_{\mathcal{N}_{R_2}}\|,$$

the rejector R_1 rejects a larger amount of samples than the rejector R_2,

$$|\mathcal{R}_{R_1}| > |\mathcal{R}_{R_2}|.$$

Equal number of nonrejected misclassified samples

If both rejectors have the same number of misclassified samples not rejected, and if rejector R_1 rejects fewer samples than R_2, then R_1 outperforms R_1. The rejector R_1 outperforms R_2 when,

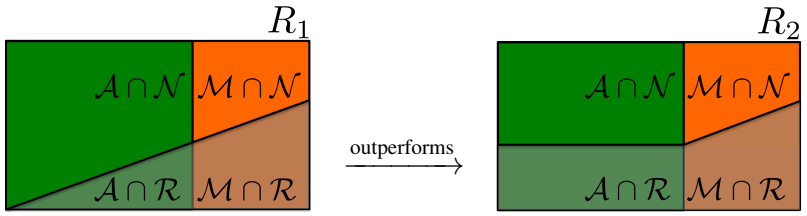


Figure 4.5: R_1 outperforms R_2 with equal number of nonrejected misclassified samples

for the same number of misclassified samples not rejected

$$\|\mathbf{1} - a_{\mathcal{N}_{R_1}}\| = \|\mathbf{1} - a_{\mathcal{N}_{R_2}}\|,$$

the rejector R_1 rejects a smaller amount of samples than the rejector R_2,

$$|\mathcal{R}_{R_1}| < |\mathcal{R}_{R_2}|.$$

4.3 Desired properties of performance measures

The definition of the rejection problem as the partitioning of the accuracy vector $\boldsymbol{a}$ based on two disjoint supports $\mathcal{N}$ and $\mathcal{R}$ is general and allows us to define desired characteristics for any generic performance measure α that evaluates the performance of classification with rejection.

We start by introducing the rejected fraction r (equivalent to the concept of rejection rate used in Chapter 2), as the ratio of rejected samples versus the overall number of samples,

$$r = \frac{n-k}{n} = \frac{|\boldsymbol{\mathcal{R}}|}{|\boldsymbol{\mathcal{R}}| + |\boldsymbol{\mathcal{N}}|} = \frac{\blacktriangle}{\blacksquare}. \tag{4.2}$$

4.3.1 Property I: Performance measure is a function of the rejected fraction

Given a performance measures α it should be a function of the fraction of rejected samples r:

$$\alpha = \alpha(r). \tag{4.3}$$

4.3.2 Property II: Performance measure is able to compare different rejector mechanisms working at the same rejected fraction

For the same classifier C, and for two different rejection mechanisms R_1 and R_2, the performance measures $\alpha(C, R_1, r)$ and $\alpha(C, R_2, r)$ should be able to compare the rejection mechanisms R_1 and R_2 when rejecting the same fraction:

$$\overbrace{\alpha(C, R_1, r)}^{\text{rejector } R_1} > \overbrace{\alpha(C, R_2, r)}^{\text{rejector } R_2} \iff R_1 \text{ outperforms } R_2. \tag{4.4}$$

4.3.3 Property III: Performance measure is able to compare different rejector mechanisms working at different rejected fractions

On the other hand, it is also desired that the performance measure be able to compare the performance of different rejection mechanisms R_1 and R_2 when they reject different fractions r_1 and r_2:

$$R_1 \text{ outperforms } R_2 \implies \overbrace{\alpha(C, R_1, r_1)}^{\text{rejector } R_1} > \overbrace{\alpha(C, R_2, r_2)}^{\text{rejector } R_2}. \tag{4.5}$$

4.3.4 Property IV: Maximum and minimum values for performance measures

Any performance measure should achieve its maximum when $\mathcal{N}$ coincides with $\mathcal{A}$ and thus $\mathcal{R}$ with $\mathcal{M}$, corresponding to simultaneously rejecting all misclassified samples and not rejecting any accurately classified sample. Similarly, the performance measure should achieve its minimum when $\mathcal{N}$ coincides with $\mathcal{M}$ and $\mathcal{R}$ with $\mathcal{A}$, corresponding to rejecting all accurately classified samples and not rejecting any misclassified sample.

4.4 Performance measures

We are now ready to define the three performance measures. First, we will show that the non-rejected accuracy, as used extensively in the literature, satisfies all our properties. We will then present two other measures that also satisfy the same properties: classification quality and rejection quality.

4.4.1 Nonrejected accuracy

The nonrejected accuracy measures the accuracy on the subset of nonrejected samples

$$A = \frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{n-k} = \frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{|\mathcal{N}|} = \frac{\text{■}}{\text{■■}}.$$

The nonrejected accuracy measures the proportion of samples that are accurately classified and not rejected compared to the samples that are not rejected. In a probabilistic interpretation, it is equivalent to the conditional probability of a sample being accurately classified given that it was not rejected.

Property I We can represent the nonrejected accuracy as a function of the rejected fraction,

$$A = \frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{|\mathcal{N}|} = \frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{n(1-r)} = A(r), \quad \square \tag{4.6}$$

satisfying Property I.

Property II For the same rejected fraction r, we have that if the nonrejected accuracy for R_1 is greater than the nonrejected accuracy for R_2, then

$$A_{R_1}(r) = \frac{\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|}{(1-r)n} > \frac{\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|}{(1-r)n} = A_{R_2}(r), \quad \square$$

meaning R_1 outperforms R_2.

Property III If R_1 outperforms R_2, for different rejected fractions $r_1 > r_2$, then $\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\| = \|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|$, leading to

$$A_{R_1}(r_1) = \frac{\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|}{(1-r_1)n} = \frac{\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|}{(1-r_1)n} > \frac{\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|}{(1-r_2)n} = A_{R_2}(r_2). \quad \square$$

If R_1 outperforms R_2, for different rejected fractions $r_1 < r_2$, then

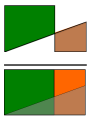
$$\begin{aligned}
\|\mathbf{1} - \boldsymbol{a}_{\mathcal{N}_{R_1}}\| = \|\mathbf{1} - \boldsymbol{a}_{\mathcal{N}_{R_2}}\| &\iff \\
(1 - r_1)n - \|\boldsymbol{a}_{\mathcal{N}_{R_1}}\| = (1 - r_2)n - \|\boldsymbol{a}_{\mathcal{N}_{R_2}}\| &\iff \\
(1 - A_{R_1}(r_1)) = \frac{(1 - r_2)}{(1 - r_1)} - \frac{(1 - r_2)\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|}{(1 - r_1)(1 - r_2)} &\iff \\
(1 - A_{R_1}(r_1)) = \frac{(1 - r_2)}{(1 - r_1)}(1 - A_{R_2}(r_2)) &\iff \\
1 - A_{R_1}(r_1) < 1 - A_{r_2}(r_2) &\iff \\
A_{R_1}(r_1) > A_{R_2}(r_2) \quad \square &
\end{aligned}$$

Property IV The nonrejected accuracy achieves its maximum, 1, when $\mathcal{N} = \mathcal{A}$ and $\mathcal{R} = \mathcal{M}$. This maximum is not unique however. Any selection of $\mathcal{N}$ such that $\mathcal{N} \subset \mathcal{A}$ achieves a maximum value of nonrejected accuracy. The minimum of the nonrejected accuracy, 0, is achieved when $\mathcal{N} = \mathcal{M}$ and $\mathcal{R} = \mathcal{A}$. Any selection of $\mathcal{N}$ such that $\mathcal{N} \subset \mathcal{M}$ achieves a minimum value of nonrejected accuracy.

4.4.2 Classification quality

The classification quality measures the correct decision making of the classifier-rejector, assessing both the performance of the classifier on the set of nonrejected samples and the performance of the rejector on the set of misclassified samples. This equates to measuring the number of accurately classified samples not rejected $\mathcal{A} \cap \mathcal{N}$ and the number of misclassified samples rejected $\mathcal{M} \cap \mathcal{R}$,

$$Q = \frac{\|\boldsymbol{a}_{\mathcal{N}}\| + \|\mathbf{1} - \boldsymbol{a}_{\mathcal{R}}\|}{|\mathcal{N}| + |\mathcal{R}|} = \frac{\|\boldsymbol{a}_{\mathcal{N}}\| + \|\mathbf{1} - \boldsymbol{a}_{\mathcal{R}}\|}{n} = \frac{}{}.$$



In a probabilistic interpretation, this is equivalent to the probability of a sample being accurately classified and not rejected or a sample being misclassified and rejected.

Property I To represent the classification quality Q as a function of the fraction of rejected samples r, we analyze separately the performance of the classifier on the subset of nonrejected samples and the performance of the rejector on the subset of misclassified samples. The performance of the classifier on the subset of nonrejected samples is the proportion of accurately classified samples not rejected to the total number of samples, which can be easily represented in terms of the nonrejected accuracy as follows,

$$\frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{n} = \frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{n(1 - r)}(1 - r) = A(r)(1 - r). \tag{4.7}$$

The performance of the rejector on the subset of misclassified samples is

$$\frac{\|\mathbf{1}-\boldsymbol{a}_{\mathcal{R}}\|}{n}=\frac{\|\mathbf{1}-\boldsymbol{a}\|-\|\mathbf{1}-\boldsymbol{a}_{\mathcal{N}}\|}{n}=1-A(0)-\frac{\|\mathbf{1}-\boldsymbol{a}_{\mathcal{N}}\|}{n}=$$
$$1-A(0)-\frac{k-\|\boldsymbol{a}_{\mathcal{N}}\|}{n}=1-A(0)-\frac{k}{n}+\frac{\|\boldsymbol{a}_{\mathcal{N}}\|}{n}=$$
$$1-A(0)-(1-r)+A(r)(1-r)=-A(0)+r+A(r)(1-r). \tag{4.8}$$

By combining (4.7) and (4.8), we can represent the classification quality as

$$Q(r)=2A(r)(1-r)+r-A(0), \quad \square \tag{4.9}$$

satisfying Property I.

Property II With representation of the classification quality in (4.9), we can note that, for the same rejected fraction r, if the classification quality for R_1 is higher than the classification quality for R_2, then

$$Q_{R_1}(r)>Q_{R_2}(r) \iff$$
$$2A_{R_1}(r)(1-r)-A(0)>2A_{R_2}(r)(1-r)-A(0) \iff$$
$$A_{R_1}>A_{R_1} \iff \|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|>\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|. \quad \square$$

Property III If R_1 outperforms R_2, for different rejected fractions $r_1>r_2$, then $\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|=\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|$, and

$$nQ_{R_1}(r_1)=\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|+\|1-\boldsymbol{a}_{\mathcal{R}_{R_1}}\|=\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|+|\mathcal{R}_{R_1}|-\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|=$$
$$\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|+r_1n-\|\boldsymbol{a}\|+|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|>\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|+r_2n-\|\boldsymbol{a}\|+|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|=$$
$$\|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|+r_1n-\|\boldsymbol{a}\|+|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|=nQ_{R_2}(r_2). \quad \square$$

If R_1 outperforms R_2, for different rejected fractions $r_1<r_2$, then $\|1-\boldsymbol{a}_{\mathcal{N}_{R_1}}\|=\|1-\boldsymbol{a}_{\mathcal{N}_{R_2}}\|$, and

$$nQ_{R_1}(r_1)=\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|+\|1-\boldsymbol{a}_{\mathcal{R}_{R_1}}\|=$$
$$\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|+|\mathcal{R}_{R_1}|-\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|=$$
$$|\mathcal{N}_{R_1}|+|\mathcal{R}_{R_1}|-\|1-\boldsymbol{a}_{\mathcal{N}_{R_1}}\|-(\|\boldsymbol{a}\|-\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\|)=$$
$$|\mathcal{N}_{R_1}|+|\mathcal{R}_{R_1}|-\|1-\boldsymbol{a}_{\mathcal{N}_{R_1}}\|-\|\boldsymbol{a}\|+|\mathcal{N}_{R_1}|-\|1-\boldsymbol{a}_{\mathcal{N}_{R_1}}\|=$$
$$n-A(0)+|\mathcal{N}_{R_1}|-2\|1-\boldsymbol{a}_{\mathcal{N}_{R_1}}\|>n-A(0)+|\mathcal{N}_{R_2}|-2\|1-\boldsymbol{a}_{\mathcal{N}_{R_1}}\|=$$
$$n-A(0)+|\mathcal{N}_{R_2}|-2\|1-\boldsymbol{a}_{\mathcal{N}_{R_2}}\|=nQ_{R_2}(r_2). \quad \square$$

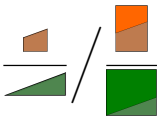
Property IV The classification quality achieves its unique maximum, 1, if $\mathcal{A}=\mathcal{N}$ and $\mathcal{M}=\mathcal{R}$, and achieves its unique minimum, 0, if $\mathcal{A}=\mathcal{R}$ and $\mathcal{M}=\mathcal{N}$. These correspond to the best and worst rejector behaviors possible, respectively.

We can use the classification as in (4.9) to compare the proportion of correct decisions between two different rejectors, for different values of rejected fractions. We note that as $Q(0)=A(0)$, we can also compare the proportion of correct decisions by using classification with rejection versus the use of no rejection at all.

4.4.3 Rejection quality

Finally, we present the rejection quality to evaluate the ability of the rejector to reject misclassified samples. This is measured through the ability to concentrate all misclassified samples onto the rejected portion of samples. The rejection quality is computed by comparing the proportion of misclassified to accurately classified samples on the set of rejected samples with the proportion of misclassified to accurately classified samples on the entire data set,

$$\phi = \frac{\|\mathbf{1} - \boldsymbol{a}_{\mathcal{R}}\|}{\|\boldsymbol{a}_{\mathcal{R}}\|} \Big/ \frac{\|\mathbf{1} - \boldsymbol{a}\|}{\|\boldsymbol{a}\|} =$$



As the rejection quality is not defined when there are no misclassified rejected samples, $\|\boldsymbol{a}_{\mathcal{R}}\| = 0$, we define $\phi \equiv \infty$ if any sample is rejected $|\mathcal{R}| > 0$, meaning that no accurately classified sample is rejected and some misclassified samples are rejected, and $\phi \equiv 1$ if no sample is rejected $|\mathcal{R}| = 0$.

Property I To express the rejection quality as a function of the rejected fraction, we note that, by (4.1), we can represent the accuracy on the rejected fraction as $\|\boldsymbol{a}_{\mathcal{R}}\| = \|\boldsymbol{a}\| - \|\boldsymbol{a}_{\mathcal{N}}\|$, and $\|1 - \boldsymbol{a}\|$ as $n(1 - A(0))$. This means that

$$\phi = \frac{r - A(0) + A(r)(1-r)}{A(0) - A(r)(1-r)} \frac{A(0)}{1 - A(0)} = \phi(r), \quad \square$$

satisfying Property I.

Property II For the same rejected fraction r, we have that if the rejection quality for R_1 is greater than the rejection quality for R_2, then

$$\begin{aligned}
\phi_{R_1}(r) > \phi_{R_2}(r) &\iff \\
\frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} > \frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} &\iff \\
\frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} > \frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|} \iff \frac{|\mathcal{R}_{R_1}| - \|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} > \frac{|\mathcal{R}_{R_2}| - \|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|} &\iff \\
\frac{|\mathcal{R}_{R_1}| - \|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} > \frac{|\mathcal{R}_{R_2}| - \|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|} \frac{|\mathcal{R}_{R_1}|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} - 1 > \frac{|\mathcal{R}_{R_2}|}{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|} - 1 &\iff \\
\frac{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{|\mathcal{R}_{R_1}|} < \frac{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{|\mathcal{R}_{R_1}|} \iff \|\boldsymbol{a}\| - \|\boldsymbol{a}_{\mathcal{N}_{R_1}}\| < \|\boldsymbol{a}\| - \|\boldsymbol{a}_{\mathcal{N}_{R_2}}\| &\iff \\
\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\| > \|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|. \quad \square &
\end{aligned}$$

Property III If R_1 outperforms R_2, for different rejected fractions $r_1 > r_2$, then $\|\boldsymbol{a}_{\mathcal{N}_{R_1}}\| = \|\boldsymbol{a}_{\mathcal{N}_{R_2}}\|$. As $\|\boldsymbol{a}\| = \|\boldsymbol{a}_{\mathcal{N}}\| + \|\boldsymbol{a}_{\mathcal{R}}\|$ and $r_1 > r_2$, we have $\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\| = \|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|$ and $|\mathcal{R}_{R_1}| > |\mathcal{R}_{R_2}|$

respectively, leading to

$$\phi_{R_1}(r_1) = \frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} =$$
$$\left(\frac{|\mathcal{R}_{R_1}|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} - 1 \right) \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} > \left(\frac{|\mathcal{R}_{R_2}|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} - 1 \right) \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} =$$
$$\frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_2}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} = \phi_{R_2}(r_2). \quad \square$$

If R_1 outperforms R_2, for different rejected fractions $r_1 < r_2$, *i.e.* $|\mathcal{N}_{R_1}| > |\mathcal{N}_{R_2}|$, then $\|1 - \boldsymbol{a}_{\mathcal{N}_{R_1}}\| = \|1 - \boldsymbol{a}_{\mathcal{N}_{R_2}}\|$. This means that $\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\| = \|1 - \boldsymbol{a}_{\mathcal{R}_{R_2}}\|$ and $|\mathcal{R}_1| < |\mathcal{R}_2|$,

$$\phi_{R_1}(r_1) = \frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{\|\boldsymbol{a}_{\mathcal{R}_{R_1}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} =$$
$$\frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{|\mathcal{R}_{R_1}| - \|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} > \frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|}{|\mathcal{R}_{R_2}| - \|1 - \boldsymbol{a}_{\mathcal{R}_{R_1}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} =$$
$$\frac{\|1 - \boldsymbol{a}_{\mathcal{R}_{R_2}}\|}{|\mathcal{R}_{R_2}| - \|1 - \boldsymbol{a}_{\mathcal{R}_{R_2}}\|} \frac{\|\boldsymbol{a}\|}{\|1 - \boldsymbol{a}\|} = \phi_{R_2}(r_2). \quad \square$$

Property IV The rejection quality achieves its maximum, ∞, when $\mathcal{N} = \mathcal{A}$ and $\mathcal{R} = \mathcal{M}$. This maximum is not unique. Any selection of $\mathcal{R}$ such that $\mathcal{R} \subset \mathcal{M}$ results in maximum values of rejection quality. Conversely, the rejection quality achieves its minimum, 0, when $\mathcal{R} = \mathcal{A}$ and $\mathcal{N} = \mathcal{M}$. This maximum is not unique. Any selection of $\mathcal{R}$ such that $\mathcal{R} \subset \mathcal{A}$ results in minimum values of rejection quality.

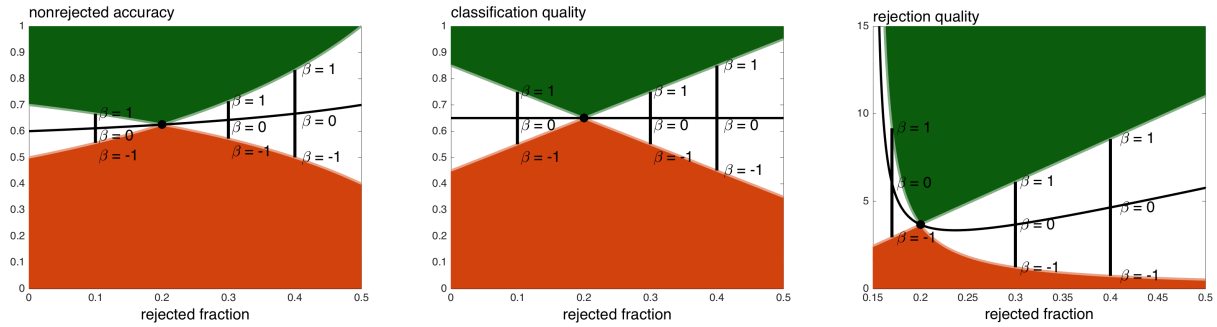
4.5 Quantifying performance

With the three performance measures defined, we can now compare the performance of classifiers with rejection. We illustrate this in Fig. 4.6, where we consider a general classifier with rejection. In the figure, black circles in the center correspond to a classifier with rejection that rejects 20% of the samples, with a nonrejected accuracy of 62.5%, a classification quality of 65%, and a rejection quality of 3.67; we call that black circle a reference *operating point*.

4.5.1 Reference operating point, operating point, and operating set

A set of performance measures and the associated rejected fraction r correspond to a *reference operating point* of the classifier with rejection. Given a reference operating point, we define the *operating set* as the set of achievable operating points as a function of the rejected fraction. This further means that for each operating point of a classifier with rejection there is an associated operating set.

Any point in the green region of each of the plots in Fig. 4.6 is an operating point of a classifier with rejection that outperforms the one at the reference operating point (black circle), and any operating point in the orange region is an operating point of a classifier with rejection that is outperformed by the one at the reference operating point (black circle), regardless of the cost function (assuming that the cost of rejection is never greater than the cost of misclassification). In white regions, performance *depends* on the trade-off between rejection and misclassification, and is thus dependent of the cost function. The borders of the green and orange regions correspond to the best and worst behaviors possible, respectively, of classifiers with rejection as compared to the reference operating point. Thus, given the reference operating point, its correspondent *operating set* is the union of the white regions including the borders.



(a) Nonrejected accuracy (b) Classification quality (c) Rejection quality

Figure 4.6: Performance measures with outperformance (green) under-performance (orange) regions for a reference operating point (black circle). Reference classifier rejects 20% of the samples and achieves a nonrejected accuracy of 62.5%, classification quality of 65%, and a rejection quality of 3.67. The parameter β, which we term *relative optimality*, measures the correctness of rejection; $\beta = 1$ corresponds to the best and $\beta = -1$ to the worst rejection behaviors possible, respectively.

4.5.2 Relative optimality

To compare the behavior of a classifier with rejection in the white region to that at the reference operating point, we measure how close that classifier is to the green and orange region borders, corresponding to the best and worst behaviors respectively. Let $\beta = 0$ denote the curve that corresponds to the middle point between the best and worst behaviors (black curve in Fig. 4.6), $\beta = 1$ to the best behavior (border with the green region), and $\beta = -1$ to the worst behavior (border with the orange region). We call β *relative optimality*, as it compares the behavior of a classifier with rejection relative to a given reference operating point.

Let us consider a reference operating point defined by a nonrejected accuracy A_0 at a rejected fraction r_0; we can now compare the performance at an arbitrary operating point (A_1, r_1) with that at a reference operating point (A_0, r_0) by computing the relative optimality

$$\beta = \begin{cases} 2\frac{A_1(1-r_1)-A_0(1-r_0)}{r_1-r_0} + 1, & \text{if } r_1 > r_0, \\ -2\frac{A_1(1-r_1)-A_0(1-r_0)}{r_1-r_0} - 1, & \text{if } r_1 < r_0. \end{cases} \tag{4.10}$$

4.5.3 Cost function

The relative optimality allows us to compare any two operating points of a classifier with rejection taking in account a cost function L which measures the relative cost of rejection versus misclassification. Let us consider the generic cost function

$$L_\rho(\hat{y}_i^R) = \begin{cases} 0, & \widehat{\mathbf{y}}_i^R \text{ accurately classified and not rejected;} \\ 1, & \widehat{\mathbf{y}}_i^R \text{ misclassified and not rejected;} \\ \rho, & \widehat{\mathbf{y}}_i^R \text{ rejected,} \end{cases} \tag{4.11}$$

where ρ is the cost of rejection and represents the trade-off between rejection and misclassification. We can compute the cost function (4.11) at an operating point (A, r) as a function of the nonrejected accuracy and the rejected fraction as

$$L_\rho(A, r) = (1 - r)(1 - A)n + \rho r n.$$

We now connect the concept of relative optimality with the generic cost function L as follows.

Theorem 1. For an operating point (A_1, r_1) with a relative optimality β relative to the reference operating point (A_0, r_0), and $r_1 > r_0$,

$$\operatorname{sgn}(\Delta L_\rho) = \operatorname{sgn}(L_\rho(A_0, r_0) - L_\rho(A_1, r_1)) = \operatorname{sgn}\left(\frac{\beta + 1}{2} - \rho\right), \tag{4.12}$$

where ΔL_ρ is the difference between the cost function at the reference operating point (A_0, r_0) and the cost function at the operating point (A_1, r_1).

Proof. Let $r_1 > r_0$, then we have that the cost function at a generic operating point (A, r) is

$$L_\rho(A, r) = (1 - r)(1 - A)n + \rho r n,$$

as we have $(1 - r)(1 - A)n$ misclassified samples, $(1 - r)An$ accurately classified samples, and rn rejected samples, and thus

$$\begin{aligned} \Delta L_\rho &= n\left((1 - r_0)(1 - A_0) + \rho r_0 - (1 - r_1)(1 - A_1) - \rho r_1\right) \\ &= n\left(r_1 - r_0 - (1 - r_0)A_0 + (1 - r_1)A_1 + \rho(r_0 - r_1)\right). \end{aligned} \tag{4.13}$$

On the other hand, from (4.10), we have that

$$A_1(1 - r_1) - A_0(1 - r_0) = \frac{\beta - 1}{2}(r_1 - r_0). \tag{4.14}$$

By combining (4.13) and (4.14), we have that

$$\Delta L_\rho = n(r_1 - r_0)\left(\frac{\beta + 1}{2} - \rho\right). \tag{4.15}$$

Because $r_1 - r_0$ and n are positive, ΔL_ρ and $(\beta + 1)/2 - \rho$ have the same sign. □

The previous discussion allows us to compare a classifier with rejection R_1 to the reference operating point R_0 as follows. Let the operating point (A_1, r_1) be at relative optimality β with respect to the reference operating point (A_0, r_0), then

$$\begin{cases} L_\rho(A_1, r_1) < L_\rho(A_0, r_0), & \text{for } \rho < (\beta+1)/2; \\ L_\rho(A_1, r_1) = L_\rho(A_0, r_0), & \text{for } \rho = (\beta+1)/2, \\ L_\rho(A_1, r_1) > L_\rho(A_0, r_0), & \text{for } \rho > (\beta+1)/2. \end{cases} \tag{4.16}$$

4.5.4 Comparing performance of classifiers with rejection

Let us consider a classifier C and two rejectors R_1 and R_0, with $r_1 > r_0$, and a cost function with a rejection-misclassification trade-off ρ. Let β be the relative optimality of the operating point of rejector R_1 at r_1 with respect to the reference operating point of R_0 at r_0.

From (4.16), and given the cost function with a rejection-misclassification trade-off ρ, we can connect the concept of outperformance, the relative optimality β and the rejection-misclassification trade-off ρ as follows,

- Rejector R_1 outperforms R_0 when $\beta > 2\rho - 1$;
- Rejector R_0 outperforms R_1 when $\beta < 2\rho - 1$;
- Rejector R_0 and R_1 are equivalent in terms of performance when $\beta = 2\rho - 1$.

This means that rejector R_1 outperforms R_0 when the following equivalent conditions are satisfied:

$$\begin{aligned} A_{R_1}(r_1) > A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + (\rho-1)\frac{r_1-r_0}{1-r_1} \iff \\ Q_{R_1}(r_1) > Q_{R_0}(r_0) + (2\rho-1)(r_1-r_0). \end{aligned}$$

Conversely, rejector R_0 outperforms R_1 when the following equivalent conditions are satisfied:

$$\begin{aligned} A_{R_1}(r_1) < A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + (\rho-1)\frac{r_1-r_0}{1-r_1} \iff \\ Q_{R_1}(r_1) < Q_{R_0}(r_0) + (2\rho-1)(r_1-r_0). \end{aligned}$$

Finally, rejectors R_0 and R_1 are equivalent in terms of performance when the following equivalent conditions are satisfied:

$$\begin{aligned} A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + (\rho-1)\frac{r_1-r_0}{1-r_1} \iff \\ Q_{R_1}(r_1) = Q_{R_0}(r_0) + (2\rho-1)(r_1-r_0). \end{aligned}$$

This conclusion can be obtained through a representation of the nonrejected accuracy and the classification quality across two different operating points based on the relative optimality. We can represent the nonrejected accuracy $A_{R_1}(r_1)$ as a function of $A_{R_0}(r_0)$ by noting that the best case scenario is

$$A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + \frac{r_1-r_0}{1-r_1},$$

corresponding to $\beta = 1$, and the worst case scenario is

$$A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1},$$

corresponding to $\beta = -1$. This results in a representation of the nonrejected accuracy $A_{R_1}(r_1)$ as

$$A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + \frac{\beta-1}{2}\frac{r_1-r_0}{1-r_1}.$$

Thus, rejector R_1 outperforms R_0 when

$$A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + \frac{\beta-1}{2}\frac{r_1-r_0}{1-r_1} > A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + (\rho-1)\frac{r_1-r_0}{1-r_1},$$

conversely, rejector R_0 outperforms R_1 when

$$A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + \frac{\beta-1}{2}\frac{r_1-r_0}{1-r_1} < A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + (\rho-1)\frac{r_1-r_0}{1-r_1},$$

and rejectors R_0 and R_1 are equivalent in terms of performance when

$$A_{R_1}(r_1) = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + \frac{\beta-1}{2}\frac{r_1-r_0}{1-r_1} = A_{R_0}(r_0)\frac{1-r_0}{1-r_1} + (\rho-1)\frac{r_1-r_0}{1-r_1}. \quad \square$$

The same line of though can be applied to the classification quality. We can represent $Q_{R_1}(r_1)$ as a function of $Q_{R_0}(r_0)$ by noting that the best case scenario is

$$Q_{R_1}(r_1) = Q_{R_0}(r_0) + (r_1 - r_0),$$

corresponding to $\beta = 1$, and the worst case scenario is

$$Q_{R_1}(r_1) = Q_{R_0}(r_0) - (r_1 - r_0),$$

corresponding to $\beta = -1$. This results in a representation of the classification quality $Q_{R_1}(r_1)$ as

$$Q_{R_1}(r_1) = Q_{R_0}(r_0) + \beta(r_1 - r_0).$$

Thus, rejector R_1 outperforms R_0 when

$$Q_{R_1}(r_1) = Q_{R_0}(r_0) + \beta(r_1 - r_0) > Q_{R_0}(r_0) + (2\rho - 1)(r_1 - r_0),$$

conversely, rejector R_0 outperforms R_1 when

$$Q_{R_1}(r_1) = Q_{R_0}(r_0) + \beta(r_1 - r_0) < Q_{R_0}(r_0) + (2\rho - 1)(r_1 - r_0),$$

and rejectors R_0 and R_1 are equivalent in terms of performance when

$$Q_{R_1}(r_1) = Q_{R_0}(r_0) + \beta(r_1 - r_0) = Q_{R_0}(r_0) + (2\rho - 1)(r_1 - r_0). \quad \square$$

4.6 Specifying the behavior of the rejector

A classifier with rejection can be seen as a coupling of two classifiers if we considered the rejector R to be a binary classifier on the output $\hat{y}$ of the classifier C, assigning to each sample a rejected or nonrejected label. Ideally, R should classify as rejected all samples misclassified by C and classify as nonrejected all the samples accurately classified by C.

In this binary classification formulation, the classification quality Q becomes the *accuracy* of the binary classifier R, the accuracy of the nonrejected samples A becomes the *precision* (positive predictive value) of the binary classifier R, and the rejection quality ϕ becomes the *positive likelihood ratio* (the ratio between the true positive rate and the false positive rate) of the binary classifier R. The rejected fraction becomes the ratio between the number of samples classified as rejected and the total number of samples.

This formulation allows us to show that the triplet $(A(r), Q(r), r)$ completely specifies the behavior of the rejector by relating the triplet to the confusion matrix associated with the binary classifier R. As we are able to reconstruct the confusion matrix from the triplet, we are thus able to show that the triplet $(A(r), Q(r), r)$ is sufficient to describe the behavior of the rejector.

Theorem 2. The set of measures $(A(r), Q(r), r)$ completely specifies the behavior of the rejector.

Proof. Let us consider the following confusion matrix associated with the interpretation of R as a binary classifier:

$$\begin{bmatrix} |\mathcal{A} \cap \mathcal{N}| & |\mathcal{M} \cap \mathcal{N}| \\ |\mathcal{A} \cap \mathcal{R}| & |\mathcal{M} \cap \mathcal{R}| \end{bmatrix},$$

where $|\mathcal{A} \cap \mathcal{N}|$ denotes the number of samples accurately classified and not rejected, $|\mathcal{M} \cap \mathcal{N}|$ the number of samples misclassified and not rejected, $|\mathcal{A} \cap \mathcal{R}|$ the number of samples accurately classified and rejected, and $|\mathcal{M} \cap \mathcal{R}|$ the number of samples misclassified and rejected. Given that n binary classifications classified n samples, the confusion matrix associated with R can be uniquely obtained from the following full rank system:

$$\begin{bmatrix} |\mathcal{A} \cap \mathcal{N}| \\ |\mathcal{M} \cap \mathcal{N}| \\ |\mathcal{A} \cap \mathcal{R}| \\ |\mathcal{M} \cap \mathcal{R}| \end{bmatrix} = n \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ r \\ Q(r) \\ A(r)\,(1-r) \end{bmatrix}.$$

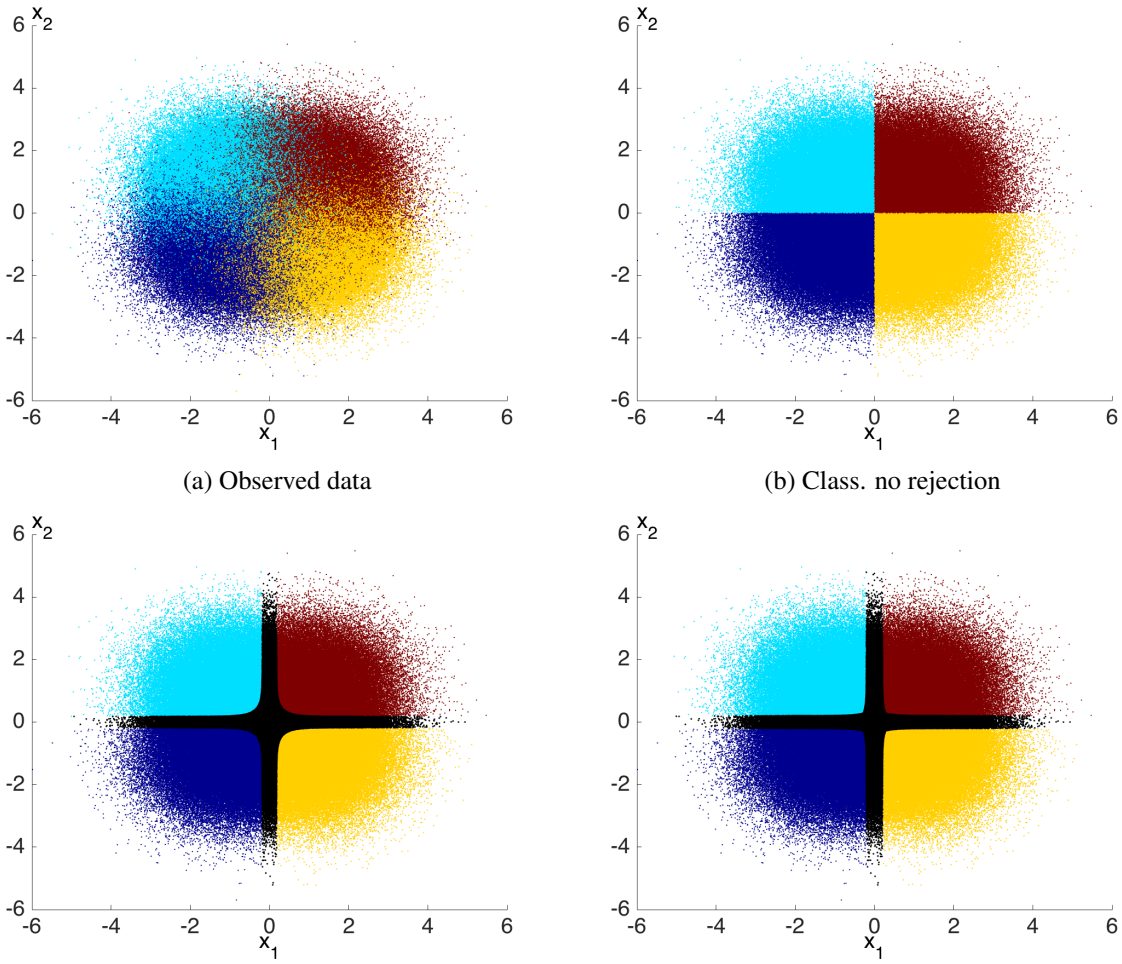
Therefore, as the set of measures and the confusion matrix are related by a full-rank system, the set of measures $(A(r), Q(r), r)$ completely specifies describes the behavior of the rejector. □

4.7 Experimental results

To illustrate the use of the proposed performance measures, we apply them to the analysis of the performance of classifiers with rejection applied to synthetic data here, and to real data in Part III. We use a simple synthetic problem to illustrate the problem of performance evaluation of classification with rejection.

4.7.1 Synthetic data

As a toy example, we consider a classification problem consisting of four two-dimensional Gaussians with identity covariance matrix and centers at $(\pm 1, \pm 1)$.



(a) Observed data

(b) Class. no rejection

(c) Class 20% rej. max. prob.

(d) Class 20% rej. breaking ties

Figure 4.7: Synthetic data example. Samples of four equally likely Gaussians with equal covariance (identity covariance matrix) and significant overlap (centered at $(\pm 1, \pm 1)$), classified with rejection (in black). (a) Observed data, (b) classification with no rejection, (c) classification with 20% rejection using maximum probability rejector, and (d) classification with 20% rejection using breaking ties rejector. The differences between the two rejectors are clear near the origin.

The Gaussians overlap significantly, as shown in Fig.4.7(a). This results in a simple classification decision using the Maximum Likelihood criterion: for each sample, assign the label of the class with the closest center as in Fig.4.7(b).

We illustrate our performance measures by comparing two simple rejection mechanisms:

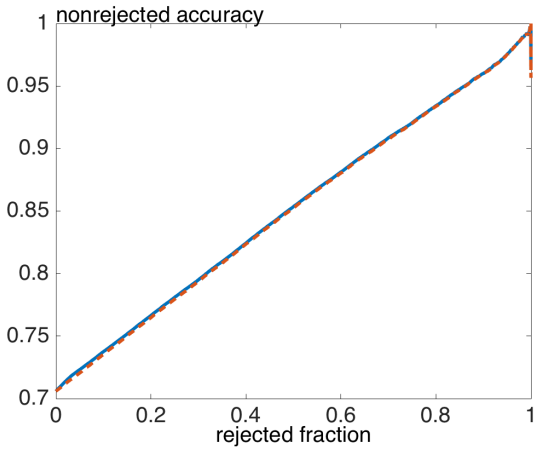
1. *Maximum probability rejector*, which, given a classifier and a rejected fraction, rejects the fraction of samples with lowest probability;
2. *Breaking ties rejector*, which, given a classifier and a rejected fraction, rejects the fraction of samples with lowest difference between the highest and second-highest class probabilities.

In Fig.4.8, we can see the performance measures computed for all possible rejected fractions for each of the two rejectors. It is clear that the with the accuracy-rejection curves alone, as shown in Fig.4.8(a), we are not able to single out any operating point of the classifier with rejection. On the other hand, with the classification quality in Fig.4.8(b), we can identify where the rejector is maximizing the number of correct decisions, and for which cases having a reject option outperforms not having a reject option. As illustrated in Fig.4.8(c), the rejection quality provides an easy way to discriminate between two different rejectors, as it focuses on the analysis of the ratios of incorrectly classified to correctly classified samples on the set of rejected samples.

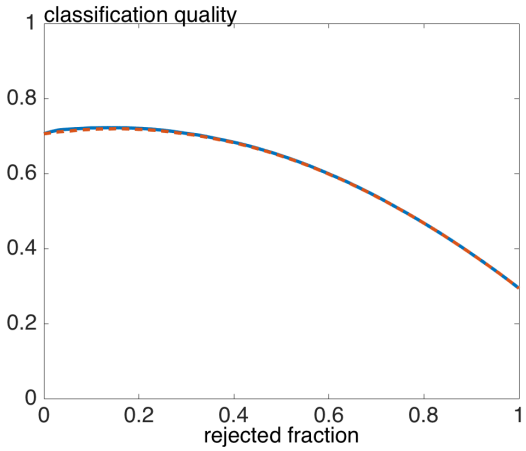
The relative optimality plots for both rejectors are present in Fig. 4.9. For each possible operating point of the rejector, for simplicity defined only by the rejected fraction, we compute the relative optimality of all other operating points of the rejector. We note that, for both rejectors, the operating point that corresponds to the maximum classification quality, has a nonnegative relative optimality with regards to all other operating points. This relative optimality plot is of particular interest for parameter selection.

4.8 Concluding remarks

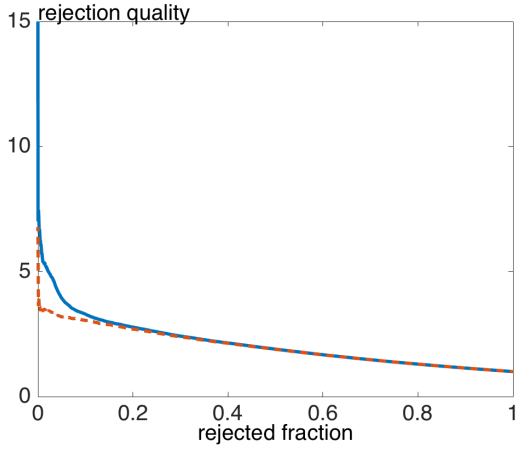
In this chapter, we introduced a set of measures to quantify performance of classifiers with rejection. We then applied these performance measures to classifiers with rejection on synthetic data. Furthermore, we connected the performance measures presented with general cost functions through the concept of relative optimality.



(a) Nonrejected accuracy

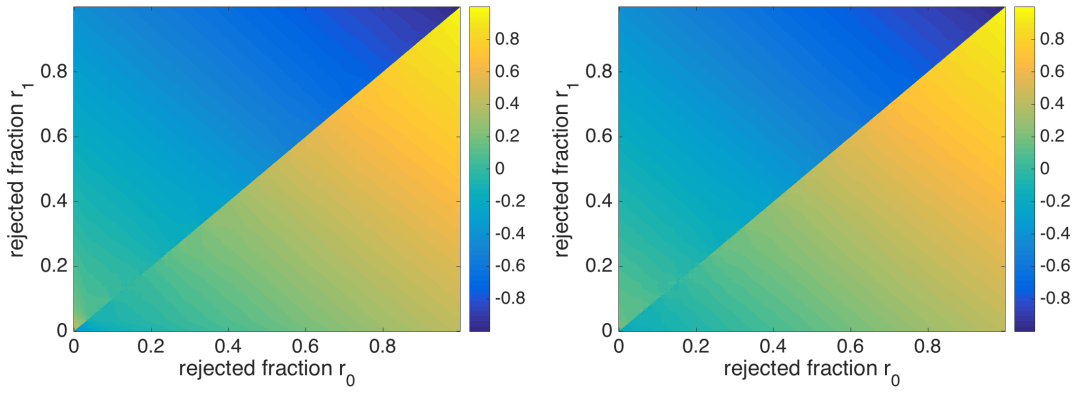


(b) Classification quality



(c) Rejection quality

Figure 4.8: Performance measures as a function of the rejected fraction for the synthetic example and the maximum probability rejector (solid blue line), and the breaking ties (dashed red line).



(a) *Maximum probability rejector* (b) *Breaking ties rejector*

Figure 4.9: Relative optimality computed for all possible pairs of operating points of (a) *maximum probability rejector* and (b) *breaking ties rejector*

Chapter 5

Classification with context

5.1 Introduction

As discussed in Chapter 3, by equipping classifiers with context, a degree of robustness can be provided to them. The use of classification with context is extensive and fruitful in image related applications. However, the problem of including context into the classification is of discrete nature, often leading to NP-hard problems. To mitigate this hurdle, approximations and relaxations are used, as seen in Chapter 3.

To avoid the discrete nature associated with the use of context, we propose a family of algorithms that formulates the problem of image classification with context in a continuous fashion based on the following key characteristics:

- Continuous hidden field driving the discrete labeling;
- Marginal maximum *a posteriori* (MMAP) estimate with marginalization across the discrete labels;
- Prior applied on the continuous hidden field;
- Convex optimization problem.

We name this family of algorithms **Seg**mentation via **S**plitting and **A**ugmented **L**agrangian **S**hrinkage **A**lgorithmn — **SegSALSA**.

This chapter is organized as follows. In Section 5.2, we describe the key components of the SegSALSA family of algorithms. In Section 5.3, we present three different priors, from which we instantiate three different algorithms based on SegSALSA. In Section 5.4, we formulate the SegSALSA family of algorithms as a convex optimization problem and derive the components of the algorithm that are common to all the members of the SegSALSA family. In Section 5.5, we present an instantiation of the SegSALSA with a vectorial total variation prior (SegSALSA-VTV) and the associated algorithm. In Section 5.6, we present an instantiation of the SegSALSA with structure tensor regularization (SegSALSA-STR) and the associated algorithm. In Section 5.7, we present an instantiation of the SegSALSA with a graph-based total variation prior (SegSALSA-GTV) and the associated algorithm. In Section 5.8, we study the parallelization potential of the SegSALSA family of algorithms. In Section 5.9, we illustrate the performance of the three instantiations of SegSALSA herein presented on three tasks: super-

vised natural image segmentation, supervised hyperspectral image classification, and supervised histopathology image classification. Section 5.10 concludes this chapter.

5.2 SegSALSA basics

5.2.1 MAP segmentation

The MAP formulations in (3.2) and (3.4) are integer optimization problems. As discussed in Chapter 3, the computation of the estimated labeling $\hat{\mathbf{y}}$ is a NP-hard problem for more than 2 classes, and only approximate or relaxed solutions can be found in reasonable time. Furthermore, the integer nature of MAP approach to the segmentation problem constraints the selection of the prior $p(\mathbf{y})$: the prior selection is dependent on the existence of efficient discrete optimization methods for the prior.

5.2.2 Hidden fields

The hidden field approach introduced in [75], allows the reformulation of the discrete segmentation problem in terms of a real-valued hidden field that drives the labeling. In [75], the hidden field is equipped with a Gaussian Markov Random Field prior that induces smoothness on the real-valued hidden field. The soft segmentation is obtained from the computation of a marginal MAP (MMAP) estimate of the hidden field, thus transforming the intractable discrete optimization problem into a convex segmentation problem. The use of hidden fields in image segmentation was also explored in [76], where wavelet-based priors are applied to the real-valued hidden field, and a MAP segmentation is performed via a generalized Expectation Maximization (EM) algorithm.

We represent the hidden field through a $K \times n$ matrix $\mathbf{z} \in \mathbb{R}^{K,n}$ that holds a collection of n hidden random vectors $\mathbf{z}_i \in \mathbb{R}^K$, one for each pixel $i \in \mathcal{S}$. The joint probability of the discrete labels $\mathbf{y}$ and hidden field $\mathbf{z}$ is

$$p(\mathbf{y}, \mathbf{z}) = p(\mathbf{y}|\mathbf{z})p(\mathbf{z}),$$

which, under assumption of conditional independence of the labels given the hidden field, results in

$$p(\mathbf{y}|\mathbf{z}) = \prod_{i \in \mathcal{S}} p(\mathbf{y}_i|\mathbf{z}_i).$$

5.2.3 Marginal maximum *a posteriori* segmentation

The joint probability of the features, labels, and hidden field $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is defined as

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y}|\mathbf{z})p(\mathbf{z}).$$

This joint probability can be marginalized with respect to the discrete labels $\mathbf{y}$ as

$$p(\mathbf{x}, \mathbf{z}) = \left(\prod_{i \in \mathcal{S}} \sum_{\mathbf{y}_i \in \mathcal{L}} p(\mathbf{x}_i|\mathbf{y}_i)p(\mathbf{y}_i|\mathbf{z}_i) \right) p(\mathbf{z}). \tag{5.1}$$

This marginalization on the discrete labels $\mathbf{y}$ transforms a discrete problem into a continuous one.

The MMAP estimate of the hidden field can be obtained from marginalized probabilities (5.1) as

$$\widehat{\mathbf{z}} \in \arg\max_{\mathbf{z}\in\mathcal{R}^{K\times n}} \left(\prod_{i\in\mathcal{S}} \sum_{\mathbf{y}_i\in\mathcal{L}} p(\mathbf{x}_i|\mathbf{y}_i)p(\mathbf{y}_i|\mathbf{z}_i) \right) p(\mathbf{z}) =$$

$$\arg\min_{\mathbf{z}\in\mathcal{R}^{K\times n}} \sum_{i\in\mathcal{S}} -\log \left(\sum_{\mathbf{y}_i\in\mathcal{L}} p(\mathbf{x}_i|\mathbf{y}_i)p(\mathbf{y}_i|\mathbf{z}_i) \right) - \log(p(\mathbf{z})). \tag{5.2}$$

Contrary to (3.2), this is no longer a discrete optimization problem, but rather a continuous optimization problem. As a result of the prior no longer being applied on the discrete labels, but on a continuous field $\mathbf{z}$, a wider family of priors can now be explored. Furthermore, if $-\log p(\mathbf{x}, \mathbf{z})$ is convex, then (5.2) is convex, which opens the door to the use of powerful solvers, namely those based on proximity calculus.

5.2.4 Link between class labels and hidden fields

Following closely the approach in [75], the following model for the hidden field is adopted:

$$[\mathbf{z}_i]_k \equiv p(\mathbf{y}_i = k|\mathbf{z}_i), \tag{5.3}$$

for each pixel $i \in \mathcal{S}$, an each label $k \in \mathcal{L}$, with $[\mathbf{z}_i]_k$ standing for the kth element of the vector $\mathbf{z}_i$. This means the components of the hidden field $\mathbf{z}$ represent a probability distribution, leading to the hidden vectors $\mathbf{z}_i$ being subject to two constraints:

- Nonnegativity constraint $\mathbf{z}_i \geq 0$;
- Sum-to-one constraint $\mathbf{1}^T\mathbf{z}_i = 1$.

The nonnegativity constraint is to be understood as a componentwise nonnegativity of each hidden vector, and the sum-to-one constraint means that the components on each hidden vector sum to one.

5.3 Priors

Using the MAP formulation (3.2) to obtain a segmentation with spatial context requires the use of spatial priors applied to the discrete labeling, resulting in a combinatorial optimization problem. Conversely, the use of hidden fields allows the use of a prior directly on the hidden field that is indirectly expressed on the labeling. Furthermore, a larger family of priors can be applied to the continuous hidden field than to the discrete labeling, thus increasing the flexibility in the choice of the prior.

A prior should promote a spatial consistency of the labeling. In image segmentation and in image classification, neighboring pixels are likely to belong to the same class.

We will now explore the use of three different prior-based regularizations that promote spatial consistency of the hidden field:

- Vectorial total variation regularization [77, 78];
- Structure tensor based regularization [79, 80];
- Graph-based total variation regularization [81, 82]

5.3.1 Vectorial total variation prior

The first prior that we will study is the Vectorial Total Variation (VTV) prior [77, 78]. The VTV prior is applied directly to the hidden field $\mathbf{z}$, promoting piecewise smoothness of the hidden field. It can be formulated as

$$-\ln p(\mathbf{z}) = \lambda_{TV} \sum_{i \in \mathcal{S}} \sqrt{\|(\boldsymbol{D}_h \mathbf{z})_i\|^2 + \|(\boldsymbol{D}_v \mathbf{z})_i\|^2} + c^{te}, \tag{5.4}$$

where $\boldsymbol{D}_h, \boldsymbol{D}_v : \mathbb{R}^{K \times n} \to \mathbb{R}^{K \times n}$ correspond to the circular horizontal difference and vertical difference operators, respectively. In addition to promoting piecewise smoothness of the hidden field, the VTV prior preserves discontinuities, leads to an alignment of discontinuities among class borders, and it is a convex (albeit nonsmooth) prior which can be optimized through proximal methods. The VTV prior can be extended to account for a spatially variable weight of each pixel:

$$-\ln p(\mathbf{z}) = \lambda_{TV} \sum_{i \in \mathcal{S}} \boldsymbol{w}_i \sqrt{\|(\boldsymbol{D}_h \mathbf{z})_i\|^2 + \|(\boldsymbol{D}_v \mathbf{z})_i\|^2} + c^{te}, \tag{5.5}$$

where $0 \leq \boldsymbol{w} \leq 1$ is a vector specifying the pixel specific weights. This extension allows us to, for example, attenuate the effect of the VTV prior on pixels that are likely to belong to boundaries between classes, where the effect of the VTV prior can be detrimental.

5.3.2 Structure tensor regularization

We now consider a generalization of the VTV prior based on structure tensor regularization (STR) priors [79]. The structure tensor prior is constructed from a patch-based Jacobian. The regularization via the structure tensor prior is based on a Schatten norm regularization, akin to [80].

Following closely the notation in [79], we define the patch-based Jacobian of the hidden field as

$$[\boldsymbol{J}\mathbf{z}]_i^T = \begin{bmatrix} (\boldsymbol{P}_1 \boldsymbol{D}_h \mathbf{z})_i^T & \dots & (\boldsymbol{P}_L \boldsymbol{D}_h \mathbf{z})_i^T \\ (\boldsymbol{P}_1 \boldsymbol{D}_v \mathbf{z})_i^T & \dots & (\boldsymbol{P}_L \boldsymbol{D}_v \mathbf{z})_i^T \end{bmatrix}, \tag{5.6}$$

where $[\boldsymbol{J}\mathbf{z}]_i$ denotes the components of the patch-based Jacobian on the ith pixel of the image, corresponding to a $(KL) \times 2$ matrix. The operators $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$ as the circular horizontal and vertical difference operators, as defined in (5.4), and $\boldsymbol{P}_j$ is as weighted shift operator that extracts a rectangular patch, with each operator being applied equally to the entire field: $\boldsymbol{D}_h, \boldsymbol{D}_v, \boldsymbol{P}_j : \mathbb{R}^{K \times n} \to \mathbb{R}^{K \times n}$. Assuming rectangular patches of size $(2M+1) \times (2M+1)$, with $L = (2M+1)^2$ pixels, $\boldsymbol{P}_j$ corresponds to the jth possible shift within the patch (from the L possible shifts) weighted by a Gaussian window centered at the center of the patch and with a standard deviation γ.

From the patch-based Jacobian of the hidden field (5.6), the structure tensor $\boldsymbol{S}_L$ is defined, for the ith pixel, as the 2×2 matrix

$$[\boldsymbol{S}_L \mathbf{z}]_i = [\boldsymbol{J}\mathbf{z}]_i^T [\boldsymbol{J}\mathbf{z}]_i. \tag{5.7}$$

The minimization of the eigenvalues of the structure tensor $\boldsymbol{S}_L$ in (5.7) leads to the penalization of variations of the field among the pixels in patch. This can be achieved with the following prior:

$$-\ln p(\mathbf{z}) \equiv \lambda \sum_{i \in \mathcal{S}} \|\sigma([\boldsymbol{S}_L \mathbf{z}]_i)\|_p + c^{te}, \tag{5.8}$$

where $\sigma(\boldsymbol{S}_L)$ represents the eigenvalues of $\boldsymbol{S}_L$. As there is an intrinsic connection between the eigenvalues of the structure tensor (5.7) (λ_+, λ_-) and the singular values of the patch-based Jacobian $(\sqrt{\lambda_+}, \sqrt{\lambda_-})$, we can minimize the singular values instead. This means that the minimization (5.8) is equivalent [79] to the minimization of the singular values of the patch-based Jacobian.

Let $\|[\boldsymbol{J}\mathbf{z}]_i\|_{S_p}$ denote the Schatten p norm of the patch-based Jacobian

$$\|[\boldsymbol{J}\mathbf{z}]_i\|_{S_p} = \|\sigma([\boldsymbol{J}\mathbf{z}]_i)\|_p,$$

where $\sigma([J\mathbf{z}]_i)$ represent the singular values of $[J\mathbf{z}]_i$. The discrete structure tensor prior can be constructed through the minimization of the singular values of the patch-based Jacobian

$$-\ln p(\mathbf{z}) \equiv \lambda \sum_{i \in \mathcal{S}} \|[\boldsymbol{J}\mathbf{z}]_i\|_{S_p} + c^{te}. \tag{5.9}$$

It should be noted that for 1×1 patches ($L = 1$) and $p = 2$, the minimization of the Schatten norm of the structure tensor is equivalent to the minimization of the VTV.

Through the minimization of the Schatten norm of the patch-based Jacobian, a generalization of the VTV prior, we are imposing smoothness across the hidden field, thus promoting the spatial consistency of the hidden field.

Similarly to the VTV prior, the discrete structure tensor prior can also be extended to account for a spatially varying weight,

$$-\ln p(\mathbf{z}) \equiv \lambda \sum_{i \in \mathcal{S}} \boldsymbol{w}_i \|[\boldsymbol{J}\mathbf{z}]_i\|_{S_p} + c^{te}, \tag{5.10}$$

where $0 \leq \boldsymbol{w} \leq 1$ is a vector containing the pixel specific weights. The weights allow us to control the strength of the prior in zones where the effect of the prior can be detrimental.

5.3.3 Graph total variation

Finally, we also explore the use of graph-based total variation priors that can harness existing unsupervised segmentations.

Unsupervised oversegmentation techniques, such as superpixelization techniques [83,84], are computationally efficient. The unsupervised oversegmentations can provide useful contextual cues, and can improve significantly the segmentation performance [81]. By transforming an

oversegmented partition of the image into a graph, we are able to impose label consistency on pixels that belong to the same partition element through the minimization of the total variation of the hidden field across the graph [82]. We note that there is no restriction to use only local segmentations of the image. This means that we can use nonlocal partitions of the image derived from the use of fast methods for computing similarity, such as the Winner Take All hash [85], to provide us with nonlocal contextual cues.

Let us consider an image with the pixels indexed by the set of pixels $\mathcal{S}$, and a partition $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_L)$ of $\mathcal{S}$ that divides the image in L nonoverlapping elements, where

$$\bigcup_{i=1}^{L} \mathcal{P}_i = \mathcal{S},$$
$$\mathcal{P}_i \cap \mathcal{P}_j = \emptyset \text{ if } i \neq j.$$

The partition $\mathcal{P}$ can be represented by an adjacency matrix $\mathbf{A}_{\mathcal{P}}$ of a graph as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected graph, where $\mathcal{V}$ and $\mathcal{E}$ correspond to the set of nodes and set of edges, respectively.

We begin by connecting directly the graph nodes with the image pixels, $\mathcal{S} = \mathcal{V}$. Then, an edge connects any two nodes that correspond to image pixels belonging to the same partition element. If v_1 and v_2 are nodes that correspond to pixels belonging to the same partition element, then there is an edge that connects them , *i.e.*, $(v_1, v_2) \in \mathcal{E}$. As there are no edges connecting nodes that correspond to pixels belonging to different partition elements, the graph $\mathcal{G}$ is an union of L disjoint subgraphs,

$$\mathcal{G} = \bigcup_{i=1}^{L} \mathcal{G}_i,$$

one for each partition element, where each subgraph $\mathcal{G}_i$ is fully connected. As illustrated in Fig.5.1, there is a one-to-one correspondence between the following entities: graph $\mathcal{G}$ and partition $\mathcal{P}$; fully connected subgraph $\mathcal{G}_i$ and partition element $\mathcal{P}_i$; and image pixel (indexed by $\mathcal{S}$) and graph node (indexed by $\mathcal{V}$).

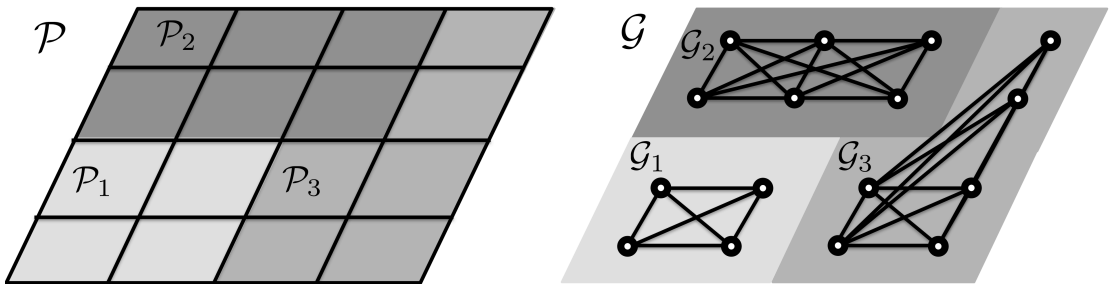


Figure 5.1: Example of partition $\mathcal{P}$ (left) and associated graph $\mathcal{G}$ (right).

With the connection between graph and partition established, we can now build an adjacency matrix $\mathbf{A}_{\mathcal{P}}$ that represents the partition. Considering the ith node, that corresponds to the ith

pixel of the image, we build $\mathcal{N}_i$ as the union of the set of nodes that share an edge with the ith node, the neighborhood of i, with i itself,

$$\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\} \cup \{i\}.$$

This corresponds to the set of nodes (pixels) that belong to the same fully connected subgraph (partition) element as the ith node (pixel). We can build the adjacency matrix $\boldsymbol{A}^{\mathcal{P}}$ as follows. The ith row $[\boldsymbol{A}^{\mathcal{P}}]_i$ is defined as

$$[\boldsymbol{A}^{\mathcal{P}}]_{i,j} = \begin{cases} \frac{1}{|\mathcal{N}_i|}, & \text{if } j \in |\mathcal{N}_i|, \\ 0, & \text{otherwise,} \end{cases} \tag{5.11}$$

where $|\mathcal{N}_i|$ denotes the number of neighbors of i (counting with i itself). This means that the ith row of the adjacency matrix $\boldsymbol{A}^{\mathcal{P}}$ indicates the pixels that belong to the same partition element as the ith pixel, normalized by the size of the partition element.

With the adjacency matrix $\boldsymbol{A}^{\mathcal{P}}$ in (5.11), corresponding to the partitioning of the image, we can define the total variation of the hidden field $\mathbf{z}$ on the partitioning graph as

$$-\ln p(\mathbf{z}) = \lambda_G \|(\boldsymbol{A}^{\mathcal{P}} - \boldsymbol{I})\mathbf{z}\|_F^2 + c^{te}, \tag{5.12}$$

where $\boldsymbol{I}$ denotes a $n \times n$ identity matrix. This graph total variation prior promotes piecewise smoothness of the hidden field inside the partition elements. Each hidden vector tends to approximate to the hidden vectors belonging to the same partition element.

Multiple partitions

The formulation in (5.12), which corresponds to a single unsupervised segmentation, can be easily extended to include multiple segmentations. Let us consider a set of multiple segmentations represented by m different partitions $\{\mathcal{P}^1, \ldots, \mathcal{P}^m\}$ of the set of pixels $\mathcal{S}$, we denote by $\boldsymbol{A}^{\mathcal{P}_j}$ the adjacency matrix that corresponds to the jth partition.

The multiple segmentation graph total variation prior can be obtained as

$$-\ln p(\mathbf{z}) = \lambda_G \sum_{j=1}^{m} \boldsymbol{q}_j \|(\boldsymbol{A}^{\mathcal{P}_j} - \boldsymbol{I})\mathbf{z}\|_F^2 + c^{te}, \tag{5.13}$$

where $0 \leq \boldsymbol{q} \leq 1$ is a vector containing the partition specific weights. This prior allows us to use simultaneously different unsupervised segmentations of the image, with different associated weights.

5.4 SegSALSA

5.4.1 Formulation

With connection between the hidden field and the class probabilities established we can now pose the general SegSALSA formulation as a convex optimization problem. Let us consider a

supervised segmentation problem defined by a probability field $\boldsymbol{p} \in \mathbb{R}^{K \times n}$, corresponding to a collection of posterior probabilities $p(\mathbf{y}_i|\mathbf{x}_i)$ (resulting either from the knowledge of the class models, or from the use of a supervised classifier) for each pixel $i \in \mathcal{S}$, we have that the MMAP estimate of the hidden field is given by

$$\begin{aligned} \widehat{\mathbf{z}}_{\text{MMAP}} \in \arg \min_{\mathbf{z} \in \mathcal{R}^{K \times n}} & \overbrace{\sum_{i \in \mathcal{S}} - \ln(\boldsymbol{p}_i^T \mathbf{z}_i)}^{\text{data term}} \overbrace{- \ln p(\mathbf{z})}^{\text{prior}} \\ \text{subject to:} \quad & \underbrace{\mathbf{z} \geq 0}_{\text{non-negativity}} \quad , \quad \underbrace{\mathbf{1}_K^T \mathbf{z} = \mathbf{1}_n}_{\text{sum-to-one}} . \end{aligned} \tag{5.14}$$

As long as the prior $-\ln p(\mathbf{z})$ is convex, the optimization 5.14 is convex, as the data term is convex and the non-negative and sum-to-one constraints are convex. The convexity of the data term results from the fact that the Hessian of $-\ln(\mathbf{p}_i^T \mathbf{z}_i)$ is semidefinite positive.

5.4.2 Reformulating the problem for SALSA

Following closely the approach in [86], we start the optimization by rewriting the optimization (5.14) in a formulation more suitable to SALSA:

$$\min_{\mathbf{z} \in \mathbb{R}^{K \times n}} \sum_{j=1}^{3} f_j(\mathbf{z}) + \overbrace{\sum_{j=1}^{m} g_j(\boldsymbol{H}_j^g \mathbf{z})}^{\text{prior}}, \tag{5.15}$$

where the convex functions f_j, for $j = 1, \ldots, 3$, correspond to the data term, sum-to-one constraint and nonnegativity constraints, and g_j for $j = 1, \ldots, m$ are convex functions, $\boldsymbol{H}_j^g$ for $j = 1, \ldots, m$ are linear operators, and $\sum_{j=1}^{m} g_j(\boldsymbol{H}_j^g \mathbf{z})$ corresponds to a prior which is a summation of m terms.

We define the convex functions f_j as

$$\begin{aligned} f_1(\boldsymbol{\zeta}) =& \sum_{i \in \mathcal{S}} - \ln(\boldsymbol{p}_i^T \boldsymbol{\zeta}_i)_+, \\ f_2(\boldsymbol{\zeta}) =& \iota_+(\boldsymbol{\zeta}), \\ f_3(\boldsymbol{\zeta}) =& \iota_1(\boldsymbol{\zeta}), \end{aligned} \tag{5.16}$$

where $\boldsymbol{\zeta}$ are dummy variables with dimensions dependent of the functions f_j. The operator $(\boldsymbol{\zeta})_+$ corresponds to nonnegative part of $\boldsymbol{\zeta}$, and we define $\ln(0) \equiv +\infty$. The function ι_+ is an indicator on the nonnegative orthant ($\mathbb{R}_0^+$), with $\iota_+ = 0$ if and only if $\boldsymbol{\zeta} \in \mathbb{R}_+^{K \times n}$, and $+\infty$ otherwise. The function ι_0 is an indicator on the set $\{\mathbf{1}_n\}$, with $\iota_1 = 0$ if and only if $\boldsymbol{\zeta} \in \{\mathbf{1}_n\}$, and $+\infty$ otherwise.

To reformulate the optimization (5.15) into a constrained optimization problem, we introduce the following variable splittings,

$$\begin{aligned}\boldsymbol{u}_j^f =& \mathbf{z}, \text{ for } j = 1, \ldots, 3, \\ \boldsymbol{u}_j^g =& \boldsymbol{H}_j^g \mathbf{z}, \text{ for } j = 1, \ldots, m,\end{aligned} \tag{5.17}$$

We stack columnwise the identity operators $\boldsymbol{I}$ into single operator $\boldsymbol{G}^f : \mathbb{R}^{K\times n} \rightarrow \mathbb{R}^{K\times 3n}$ and define $\boldsymbol{G}^g$ as a columnwise stacking of the operators $\boldsymbol{H}_j^g$ associated with the prior term, allowing us to reformulate (5.14) as the following constrained formulation

$$\begin{aligned}\min_{\boldsymbol{u}^f, \boldsymbol{u}^g, \mathbf{z}} \sum_{j=1}^{3} f_j(\boldsymbol{u}_j^f) + \sum_{j=1}^{m} g_j(\boldsymbol{u}_j^g) \\ \text{s.t. } \begin{bmatrix}\boldsymbol{u}^f \\ \boldsymbol{u}^g\end{bmatrix} = \begin{bmatrix}\boldsymbol{G}^f \\ \boldsymbol{G}^g\end{bmatrix} \mathbf{z},\end{aligned} \tag{5.18}$$

with $\boldsymbol{u}_1^f, \boldsymbol{u}_2^f, \boldsymbol{u}_3^f, \in \mathbb{R}^{K\times n}$, and the dimension of $\boldsymbol{u}_j^g$ being dependent of the prior selection.

5.4.3 SALSA formulation

With the formulation in (5.18), we are now able to apply the C-SALSA methodology [86], which basically formulates the problem as an instance of the Alternated Direction Method of Multipliers (ADMM) [87–89].

We denote the scaled Lagrange multipliers associated with the constraints $\boldsymbol{u}^f = \mathbf{G}^f\mathbf{z}$ and $\boldsymbol{u}^g = \mathbf{G}^g\mathbf{z}$ as $\boldsymbol{d}^f = [\boldsymbol{d}_1^{fT}, \ldots, \boldsymbol{d}_3^{fT}]$ and $\boldsymbol{d}^g = [\boldsymbol{d}_1^{gT}, \ldots, \boldsymbol{d}_m^{gT}]$, respectively. We thus have the following C-SALSA based formulation for (5.18),

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \left\| \begin{bmatrix}\boldsymbol{G}^f \\ \boldsymbol{G}^g\end{bmatrix} \mathbf{z} - \begin{bmatrix}\boldsymbol{u}^{f,k} \\ \boldsymbol{u}^{g,k}\end{bmatrix} - \begin{bmatrix}\boldsymbol{d}^{f,k} \\ \boldsymbol{d}^{g,k}\end{bmatrix} \right\|_F^2, \tag{5.19}$$

$$\begin{aligned}\boldsymbol{u}_i^{f,k+1} =& \arg\min_{\boldsymbol{u}^f} f_j(\boldsymbol{u}_j^f) + \frac{\mu}{2}\|\boldsymbol{G}^f\mathbf{z}^{k+1} - \boldsymbol{u}_j^f - \boldsymbol{d}_j^{f,k}\|_F^2, \\ \boldsymbol{u}_i^{g,k+1} =& \arg\min_{\boldsymbol{u}^g} g_j(\boldsymbol{u}_j^g) + \frac{\mu}{2}\|\boldsymbol{G}^g\mathbf{z}^{k+1} - \boldsymbol{u}_j^g - \boldsymbol{d}_j^{g,k}\|_F^2,\end{aligned} \tag{5.20}$$

$$\begin{aligned}\boldsymbol{d}^{f,k+1} =& \boldsymbol{d}^{f,k} - \left[\boldsymbol{G}^f\mathbf{z}^{k+1} - \boldsymbol{u}^{f,k+1}\right], \\ \boldsymbol{d}^{g,k+1} =& \boldsymbol{d}^{g,k} - \left[\boldsymbol{G}^g\mathbf{z}^{k+1} - \boldsymbol{u}^{g,k+1}\right],\end{aligned} \tag{5.21}$$

where $\mu > 0$ is a parameter of the optimization controlling the variable splitting.

5.4.4 Convergence of SegSALSA

As the feasibility set of (5.18) is compact, if $\boldsymbol{p}_i^T \mathbf{z}_i \geq 0$, for $i \in \mathcal{S}$, then the objective function in (5.14) is continuous on the feasibility set and thus has minimum point. Let

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^f \\ \mathbf{G}^g \end{bmatrix}, \quad \boldsymbol{u}^k = \begin{bmatrix} \boldsymbol{u}^{f,k} \\ \boldsymbol{u}^{g,k} \end{bmatrix}, \quad \boldsymbol{d}^k = \begin{bmatrix} \boldsymbol{d}^{f,k} \\ \boldsymbol{d}^{g,k} \end{bmatrix}.$$

As the linear operator $\mathbf{G}$ has null$(\mathbf{G}) = \{\mathbf{0}\}$, and the objective functions are closed, proper, and convex, this means that if a solution $\mathbf{z}$ exists for (5.18), then the sequence $\{\mathbf{z}^k, k = 0, 1, \ldots\}$ converges to $\mathbf{z}$, for all $\mu > 0$. If no solution $\mathbf{z}$ exists for (5.18), then at least on of the following sequences diverges: $\{\boldsymbol{u}^k, k = 0, 1, \ldots\}$, $\{\boldsymbol{d}^k, k = 0, 1, \ldots\}$. This results from the application of a theorem on the convergence of ADMM [87] applied to the convergence of the SALSA methodology.

5.4.5 Optimization with respect to the hidden field $\mathbf{z}$

The solution for (5.19) is

$$\mathbf{z}^{k+1} = (\boldsymbol{G}^* \boldsymbol{G})^{-1} \boldsymbol{G}^* (\boldsymbol{u}^k - \boldsymbol{d}^k) =$$
$$\boldsymbol{F}^{-1} \left(3\boldsymbol{I}(\boldsymbol{u}_j^{f,k} - \boldsymbol{d}_j^{f,k}) + \sum_{j=1}^{m} (\boldsymbol{H}_j^g)^* (\boldsymbol{u}_j^{g,k} - \boldsymbol{d}_j^{g,k}) \right), \tag{5.22}$$

where

$$\boldsymbol{F} = 3\boldsymbol{I} + \sum_{j=1}^{m} (\boldsymbol{H}_j^g)^* \boldsymbol{H}_j^g,$$

and $(.)^*$ corresponds to the adjoint operator with respect to the standard Euclidean norm. If the linear operators $\boldsymbol{H}_j^g$ can be represented through cyclic convolution operators, solving (5.19) with respect to the hidden field $\mathbf{z}$ can be implemented through cyclic convolution operations, thus diagonalizable in the frequency domain and consequently easily performed in the frequency domain [90].

5.4.6 Optimization with respect to the split variables

The optimization problems associated with (5.20) can be solved through proximal methods, by computing the associated Moreau proximity operators (MPO) [91] of each of the convex functions. We now present the closed form expressions of these operators for the data fit term, and sum-to-one and nonnegativity constraints.

The Moreau proximity operator for the data fit f_1 is

$$\psi_{f_1/\mu}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{\zeta}} \left(\sum_{i \in \mathcal{S}} -\ln(\boldsymbol{p}_i^T \boldsymbol{\zeta}_i) \right) + \frac{\mu}{2} \|\boldsymbol{\zeta} - \boldsymbol{\nu}\|_F^2,$$

where $\boldsymbol{\nu} \equiv [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_n] \in \mathbb{R}^{K \times n}$, and $\boldsymbol{\zeta} \equiv [\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_n] \in \mathbb{R}^{K \times n}$. This optimization is decoupled (pixelwise) with respect to $\boldsymbol{\zeta}_i$ for $i \in \mathcal{S}$, meaning

$$\psi_{f_1/\mu}(\boldsymbol{\nu}) = (\psi_{f_1/\mu}(\nu_{\mathbf{1}}), \ldots, \psi_{f_1/\mu}(\nu_{\mathbf{n}})).$$

Furthermore, we have that

$$\psi_{f_1/\mu}(\boldsymbol{\nu}_i) = \arg\min_{\boldsymbol{\zeta}_\mathbf{i}} -\ln(\boldsymbol{p}_i^T \boldsymbol{\zeta}_i) + \frac{\mu}{2}\|\boldsymbol{\zeta}_i - \boldsymbol{\nu}_i\|_F^2.$$

We find the proximal operator by finding the positive root of $\boldsymbol{p}_i^T \nabla \psi_{f_1/\mu} = 0$ with respect to $\boldsymbol{p}_i^T \boldsymbol{\zeta}_i$. This root is then used in $\nabla \psi_{f_1/\mu} = 0$. After some manipulation, we thus have

$$\psi_{f_1/\mu}(\nu_{\mathbf{i}}) = \boldsymbol{\nu}_i + \frac{\boldsymbol{p}_i}{\mu a_i},$$

where

$$a_i \equiv \frac{\boldsymbol{p}_i^T \boldsymbol{\nu}_i + \sqrt{(\boldsymbol{p}_i^T \boldsymbol{\nu}_i)^2 + \|\boldsymbol{p}_i\|^2/\mu}}{2}.$$

This operator has a complexity of $O(Kn)$, the number of classes times the number of pixels.

The Moreau proximity operator for the sum-to-one constraint f_2 is

$$\psi_{f_2/\mu}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{\zeta}} \iota_1(\boldsymbol{\zeta}) + \frac{\mu}{2}\|\boldsymbol{\zeta} - \boldsymbol{\nu}\|_F^2 = \left(\mathbf{I} - \frac{\mathbf{1}_K \mathbf{1}_K^T}{K}\right)\nu + \frac{\mathbf{1}_K \mathbf{1}_n^T}{K},$$

where $\nu, \boldsymbol{\zeta} \in \mathbb{R}^{K \times n}$. This operator is the projection in the probability simplex, and has a complexity of $O(Kn)$, the number of classes times the number of pixels.

The Moreau proximity operator for nonnegativity constraint f_3 is

$$\psi_{f_3/\mu}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{\zeta}} \iota_+(\boldsymbol{\zeta}) + \frac{\mu}{2}\|\boldsymbol{\zeta} - \boldsymbol{\nu}\|_F^2 = \max\{\mathbf{0}, \boldsymbol{\nu}\},$$

where $\nu, \boldsymbol{\zeta} \in \mathbb{R}^{K \times n}$. This operator is the projection in the first orthant, and has a complexity of $O(Kn)$, the number of classes times the number of pixels.

5.4.7 Algorithm — SegSALSA

In Alg. 1, we show the pseudocode for the SegSALSA family of algorithms. The stopping condition can be set either by imposing that both the primal and the dual residues are smaller than a given threshold [92], or by a fixed number of iterations. Setting the stop condition is a problem dependent, and should be approached case by case. In the practical setting, results show that the SegSALSA family of algorithms converges for roughly 100 iterations. Let k_{STOP} denote the final iteration of the algorithm, determined by a valid stop condition, either by threshold on the residuals or by fixed number of iterations.

Algorithm 1: SegSALSA family of algorithms

initialization:
choose $(\boldsymbol{u}_j^{f,0}, \boldsymbol{d}_j^{f,0}) \in \mathbb{R}^{n\times K}$, $j = 1, \ldots, 3$
choose $(\boldsymbol{u}_j^{g,0}, \boldsymbol{d}_j^{g,0})$, $j = 1, \ldots, m$
define $\boldsymbol{F} = 3\mathbf{I} + \sum_{j=1}^{m} \boldsymbol{H}_j^{g*}\boldsymbol{H}_j^{g}$
set $\mu \in]0, +\infty[$
for $k = 0, 1, \ldots, k_{STOP}$ **do**
  $\boldsymbol{z}^{k+1} = \boldsymbol{F}^{-1}\big(\sum_{j=1}^{3}(\boldsymbol{u}_j^{f,k} - \boldsymbol{d}_j^{f,k}) + \sum_{j=1}^{m}(\boldsymbol{H}_j^{g})^{*}(\boldsymbol{u}_j^{g,k} - \boldsymbol{d}_j^{g,k})\big)$
  for $j = 1$ **to** 3 **do**
    $\boldsymbol{u}_j^{f,k+1} = \text{prox}_{f_j/\mu}(\boldsymbol{z}^{k+1} - \boldsymbol{d}_j^{f,k})$
    $\boldsymbol{d}_j^{f,k+1} = \boldsymbol{d}_j^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_j^{f,k+1})$
  for $j = 1$ **to** m **do**
    $\boldsymbol{u}_j^{g,k+1} = \text{prox}_{g_j/\mu}(\boldsymbol{z}^{k+1} - \boldsymbol{d}_j^{g,k})$
    $\boldsymbol{d}_j^{g,k+1} = \boldsymbol{d}_j^{g,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_j^{g,k+1})$
return $\boldsymbol{z}^{k+1}$

5.4.8 Extending of SegSALSA

The optimization 5.14 defines the family of SegSALSA algorithms for classification with context. At this point, we are able to instantiate the three algorithms (defined by their different priors) simply through the definition of the prior as a sum of convex functions g_j and associated linear operators H_j^g, and respective Moreau proximity operators With the three different families of priors in hand, we can now reformulate three instances of algorithms of the SegSALSA family

- SegSALSA-VTV [1, 2] in Section 5.5— using a VTV prior (5.5);
- SegSALSA-STR [3] in Section 5.6 — using structure tensor regularization (5.10);
- SegSALSA-GTV [5] in Section 5.7 — using a a graph-based TV (5.13).

With the general SegSALSA optimization formulated in (5.14), we are able to derive the algorithm for the SegSALSA family of algorithns.

5.5 SegSALSA-VTV

The first instantiation of the SegSALSA family is through the use of a VTV prior that promotes piecewise smooth hidden fields, SegSALSA-VTV.

5.5.1 Formulation

By combining the SegSALSA formulation in (5.14) with the prior in (5.5) we obtain the following formulation.

$$\widehat{\mathbf{z}}_{\text{MMAP}} = \arg\min_{\mathbf{z}\in\mathcal{R}^{K\times n}} \sum_{i\in\mathcal{S}} -\ln(\boldsymbol{p}_i^T\mathbf{z}_i) + \lambda_{TV}\sum_{i\in\mathcal{S}} \boldsymbol{w}_i\sqrt{\|(\boldsymbol{D}_h\mathbf{z})_i\|^2 + \|(\boldsymbol{D}_v\mathbf{z})_i\|^2}, \quad (5.23)$$
$$\text{subject to:} \quad \mathbf{z}\geq 0, \quad \mathbf{1}_K^T\mathbf{z} = \mathbf{1}_n,$$

where λ_{TV} is the relative weight between the data term and the prior, and $\boldsymbol{w}_i$ is a pixel specific weight that controls the influence of the prior. This formulation is based on the definition of the VTV prior as a sum of convex functions and associated linear operators. The optimization in (5.23) is convex as the VTV prior is convex with respect to $\mathbf{z}$.

5.5.2 Optimization

The inclusion of the VTV prior in (5.15) results in

$$\min_{\mathbf{z}\in\mathbb{R}^{K\times n}} \sum_{j=1}^{3} f_j(\mathbf{z}) + \overbrace{\sum_{j=1}^{m} g_j(\boldsymbol{H}_j^g\mathbf{z})}^{\text{prior}} \iff ,$$
$$\min_{\mathbf{z}\in\mathbb{R}^{K\times n}} \sum_{j=1}^{3} f_j(\mathbf{z}) + \underbrace{\lambda_{\text{TV}}\left(\sum_{i\in\mathcal{S}} \boldsymbol{w}_i\sqrt{\|(\boldsymbol{D}_h\mathbf{z})_i\|^2 + \|(\boldsymbol{D}_v\mathbf{z})_i\|^2}\right)}_{\text{prior}},$$

which means that our prior has a single term, $m = 1$.

We define the linear operator $\boldsymbol{H}^g : \mathbb{R}^{K\times n} \to \mathbb{R}^{2K\times n}$ as

$$\boldsymbol{H}^g = \begin{pmatrix} \boldsymbol{D}_h \\ \boldsymbol{D}_v \end{pmatrix}, \quad (5.24)$$

where $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$ correspond to the circular horizontal difference operators previously defined. The convex function g is defined as

$$g(\boldsymbol{\zeta}) = \lambda_{\text{TV}}\sum_{i\in\mathcal{S}} \boldsymbol{w}_i\sqrt{\|\boldsymbol{\zeta}_n^h\|^2 + \|\boldsymbol{\zeta}_n^v\|^2},$$

where $\boldsymbol{\zeta} = [\boldsymbol{\zeta}^h \quad \boldsymbol{\zeta}^v] \in \mathbb{R}^{2K\times n}$, and $\boldsymbol{\zeta}^h$ and $\boldsymbol{\zeta}^v$ belong to the range of the horizontal and vertical difference operators $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$, respectively.

The Moreau proximity operator for the VTV prior is thus

$$\psi_{g/\mu}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{\zeta}} \left(\sum_{i\in\mathcal{S}} \boldsymbol{w}_i\sqrt{\|\boldsymbol{\zeta}_i^h\|^2 + \|\boldsymbol{\zeta}_i^v\|^2}\right) + \frac{\mu}{2\lambda_{\text{TV}}}\|\boldsymbol{\zeta} - \boldsymbol{\nu}\|_F^2,$$

where $\nu, \boldsymbol{\zeta} \in \mathbb{R}^{2K\times n}$ and $\boldsymbol{\zeta}^h, \boldsymbol{\zeta}^v \in \mathbb{R}^{K\times n}$. This optimization can be pixelwise decoupled and can be solved with the vector soft thresholding operator [91]

$$\psi_{g/\mu}(\boldsymbol{\nu}_i) = \max\left\{\mathbf{0}, \|\boldsymbol{\nu}_i\| - \lambda_{\text{TV}}\mathbf{w}_i/\mu\right\}\frac{\boldsymbol{\nu}_i}{\|\boldsymbol{\nu}_i\|}.$$

This operator has a complexity of $O(Kn)$.

5.5.3 Algorithm

Let $\boldsymbol{H}^g$ denote the stacking of the circular difference operators, as seen (5.24), we have that the SegSALSA-VTV algorithm is

Algorithm 2: SegSALSA-VTV

initialization:

choose $(\boldsymbol{u}_j^{f,0}, \boldsymbol{d}_j^{f,0}) \in \mathbb{R}^{n\times K}$, $j = 1, \ldots, 3$

choose $(\boldsymbol{u}^{g,0}, \boldsymbol{d}^{g,0}) \in \mathbb{R}^{n\times 2K}$

define $a_i = \frac{\boldsymbol{p}_i^T\boldsymbol{\nu}_i + \sqrt{(\boldsymbol{p}_i^T\boldsymbol{\nu}_i)^2 + \|\boldsymbol{p}_i\|^2/\mu}}{2}$, for $i \in \mathcal{S}$

define $\boldsymbol{B} = (\boldsymbol{H}^g)^*\boldsymbol{H}^g$

define $\boldsymbol{C} = (3\mathbf{I} + \boldsymbol{B})^{-1}$

set $\mu \in]0, +\infty[$

for $k = 0, 1, \ldots, k_{STOP}$ **do**

/* update **z** */

$\boldsymbol{z}^{k+1} = \boldsymbol{C}\big(\sum_{j=1}^{3}(\boldsymbol{u}_j^{f,k} - \boldsymbol{d}_j^{f,k}) + (\boldsymbol{H}^g)^*(\boldsymbol{u}^{g,k} - \boldsymbol{d}^{g,k})\big)$

/* update split variables */

$[\boldsymbol{u}_1^{f,k+1}]_i = [\boldsymbol{z}^{k+1} - \boldsymbol{d}_1^{f,k}]_i + \frac{\boldsymbol{p}_i}{\mu a_i}$, for $i \in \mathcal{S}$

$\boldsymbol{d}_1^{f,k+1} = \boldsymbol{d}_1^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_1^{f,k+1})$

$\boldsymbol{u}_2^{f,k+1} = \left(\mathbf{I} - \frac{\mathbf{1}_K\mathbf{1}_K^T}{K}\right)(\boldsymbol{z}^{k+1} - \boldsymbol{d}_2^{f,k}) \frac{\mathbf{1}_K\mathbf{1}_n^T}{K}$, $\boldsymbol{d}_2^{f,k+1} = \boldsymbol{d}_2^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_2^{f,k+1})$

$\boldsymbol{u}_3^{f,k+1} = \max\{\mathbf{0}, (\boldsymbol{z}^{k+1} - \boldsymbol{d}_3^{f,k})\}$,

$\boldsymbol{d}_3^{f,k+1} = \boldsymbol{d}_3^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_3^{f,k+1})$

/* VTV prior */

for $i \in \mathcal{S}$ **do**

$[\boldsymbol{u}^{g,k+1}]_i = \max\left\{\mathbf{0}, \|[(\boldsymbol{H}^g\boldsymbol{z}^{k+1} - \boldsymbol{d}^{g,k})]_i\| - \lambda_{\text{TV}}\mathbf{w}_i/\mu\right\} \frac{[(\boldsymbol{H}^g\boldsymbol{z}^{k+1} - \boldsymbol{d}^{g,k})]_i}{\|[(\boldsymbol{H}^g\boldsymbol{z}^{k+1} - \boldsymbol{d}^{g,k})]_i)\|}$

$\boldsymbol{d}^{g,k+1} = \boldsymbol{d}^{g,k} - (\boldsymbol{H}^g\boldsymbol{z}^{k+1} - \boldsymbol{u}^{g,k+1})$

return $\boldsymbol{z}_{k+1}$

5.6 SegSALSA-STR

The second instantiation of the SegSALSA family of algorithms is obtained through the use structure tensor regularization associated with the minimization of the Schatten norm of patch-based Jacobian matrices (5.10), SegSALSA-STR. The use of this prior promotes low-rank patch-based Jacobian, leading to the smoothness of the hidden field.

5.6.1 Formulation

By combining the SegSALSA formulation in (5.14) with the prior in (5.10), we obtain the following formulation

$$\widehat{\mathbf{z}}_{\text{MMAP}} = \arg \min_{\mathbf{z} \in \mathcal{R}^{K \times n}} \sum_{i \in \mathcal{S}} -\ln(\boldsymbol{p}_i^T \mathbf{z}_i) + \lambda_{\text{ST}} \sum_{i \in \mathcal{S}} \boldsymbol{w}_i \|[\boldsymbol{J}\mathbf{z}]_i\|_{S_p}, \quad \text{subject to:} \quad \mathbf{z} \geq 0, \quad \mathbf{1}_K^T \mathbf{z} = \mathbf{1}_n, \tag{5.25}$$

where λ_{ST} is the relative weight between the data term and the prior, and $\boldsymbol{w}_i$ is a pixel specific weight that controls the influence of the prior. This formulation is based on the definition of the Schatten norm of the patch-based Jacobians as a sum of convex functions and associated linear operators. The optimization (5.25) is convex as $\|[\boldsymbol{J}\mathbf{z}]_i\|_{S_p}$ is convex in $\mathbf{z}$, it is a composition of norms [80].

5.6.2 Optimization

By including the structure tensor prior in (5.15), we have

$$\min_{\mathbf{z} \in \mathbb{R}^{K \times n}} \sum_{j=1}^{3} f_j(\mathbf{z}) + \overbrace{\sum_{j=1}^{m} g_j(\boldsymbol{H}_j^g \mathbf{z})}^{\text{prior}} \iff ,$$

$$\min_{\mathbf{z} \in \mathbb{R}^{K \times n}} \sum_{j=1}^{3} f_j(\mathbf{z}) + \underbrace{\lambda_{\text{STR}} \left(\sum_{i \in \mathcal{S}} \mathbf{w}_i \|[\boldsymbol{J}\boldsymbol{z}]_i\|_{S_p} \right)}_{\text{prior}},$$

which means that our prior has a single term, $m = 1$.

We denote the linear operator $\boldsymbol{H}^g$ as the stacking of the patch-based Jacobian matrices $\boldsymbol{J}$ defined in (5.6), with $\boldsymbol{H}^g : \mathbb{R}^{K \times n} \to \mathbb{R}^{2LK \times n}$.

The convex function g is defined as

$$g(\boldsymbol{\zeta}) = \lambda_{\text{STR}} \sum_{i \in \mathcal{S}} \boldsymbol{w}_i \|[\boldsymbol{\zeta}]_i\|_{S_p},$$

where $\boldsymbol{\zeta} \in \mathbb{R}^{2LK \times n}$ corresponds to the patch-based Jacobian of the ith pixel, as defined in (5.6).

The Moreau proximity operator can be found by solving

$$\psi_{g/\mu}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{\zeta}} \sum_{i \in \mathcal{S}} \mathbf{w}_i \|[\boldsymbol{\zeta}]_i\|_{S_p} + \frac{\mu}{2\lambda_{\text{STR}}} \|\boldsymbol{\zeta} - \boldsymbol{\nu}\|_F^2,$$

where $\boldsymbol{\nu}, \boldsymbol{\zeta} \in \mathbb{R}^{(2LK)\times n}$. This optimization can be pixelwise decoupled and can be solved, for $p = 1$, with a vector soft thresholding operator on the singular values of $\boldsymbol{\nu}$ [80],

$$\begin{aligned}\psi_{g/\mu}(\boldsymbol{\nu}_i) =& \boldsymbol{U}_i \max\{0, \boldsymbol{S}_i - \lambda_{\text{STR}}/\mu\} \boldsymbol{V}_i^T, \\ \boldsymbol{U}_i \boldsymbol{S}_i \boldsymbol{V}_i^T =& \boldsymbol{\nu}_i.\end{aligned}$$

The time complexity of solving this Moreau proximity operator is dominated by the computation of n single value decompositions of matrices $(LK) \times 2$, which amounts to a time complexity of $O(n(LK)^3)$ [93].

5.6.3 Algorithm

Let $\boldsymbol{H}^g : \mathbb{R}^{K\times n} \to \mathbb{R}^{2LK\times n}$ denote the stacking of the patch-based Jacobian operators, we have that SegSALSA-STR can be expressed as follows.

Algorithm 3: SegSALSA-STR

initialization:

choose $(\boldsymbol{u}_j^{f,0}, \boldsymbol{d}_j^{f,0}) \in \mathbb{R}^{n\times K}$, $j = 1, \ldots, 3$

choose $(\boldsymbol{u}^g, \boldsymbol{d}^g) \in \mathbb{R}^{n\times 2LK}$

define $a_i = \frac{\boldsymbol{p}_i^T\boldsymbol{\nu}_i + \sqrt{(\boldsymbol{p}_i^T\boldsymbol{\nu}_i)^2 + \|\boldsymbol{p}_i\|^2/\mu}}{2}$, for $i \in \mathcal{S}$

define $\boldsymbol{B} = (\boldsymbol{H}^g)^*\boldsymbol{H}^g$

define $\boldsymbol{C} = (3\mathbf{I} + \boldsymbol{B})^{-1}$

set $\mu \in]0, +\infty[$

for $k = 0, 1, \ldots, k_{STOP}$ **do**

/* update **z** */

$\boldsymbol{z}^{k+1} = \boldsymbol{C}\big(\sum_{j=1}^{3}(\boldsymbol{u}_j^{f,k} - \boldsymbol{d}_j^{f,k}) + \boldsymbol{H}^{g*}(\boldsymbol{u}^{g,k} - \boldsymbol{d}^{g,k})\big)$

/* update split variables */

$[\boldsymbol{u}_1^{f,k+1}]_i = [\boldsymbol{z}^{k+1} - \boldsymbol{d}_1^{f,k}]_i + \frac{\boldsymbol{p}_i}{\mu a_i}$, for $i \in \mathcal{S}$

$\boldsymbol{d}_1^{f,k+1} = \boldsymbol{d}_1^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_1^{f,k+1})$

$\boldsymbol{u}_2^{f,k+1} = \left(\mathbf{I} - \frac{\mathbf{1}_K\mathbf{1}_K^T}{K}\right)(\boldsymbol{z}^{k+1} - \boldsymbol{d}_2^{f,k})\frac{\mathbf{1}_K\mathbf{1}_n^T}{K}$, $\boldsymbol{d}_2^{f,k+1} = \boldsymbol{d}_2^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_2^{f,k+1})$

$\boldsymbol{u}_3^{f,k+1} = \max\{\mathbf{0}, (\boldsymbol{z}^{k+1} - \boldsymbol{d}_3^{f,k})\}$

$\boldsymbol{d}_3^{f,k+1} = \boldsymbol{d}_3^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_3^{f,k+1})$

/* structure tensor regularization */

for $i \in \mathcal{S}$ **do**

/* single value decomposition */

$\boldsymbol{U}_i\boldsymbol{S}_i\boldsymbol{V}_i^T = [(\boldsymbol{H}^g\boldsymbol{z}^{k+1} - \boldsymbol{d}^{g,k})]_i$

/* singular value soft thresholding */

$\boldsymbol{S}_i' = \max\{\mathbf{0}, \boldsymbol{S}_i - \lambda_{\text{STR}}/\mu\}$

$[\boldsymbol{u}^{g,k+1}]_i = \boldsymbol{U}_i\boldsymbol{S}_i'\boldsymbol{V}_i^T$

$\boldsymbol{d}^{g,k+1} = \boldsymbol{d}^{g,k} - (\boldsymbol{H}^g\boldsymbol{z}^{k+1} - \boldsymbol{u}^{g,k+1})$

return $\boldsymbol{z}_{k+1}$

5.7 SegSALSA-GTV

The final instantiation of the SegSALSA family of algorithms is obtained with a sum of graph total variation (GTV) priors that impose the minimization of the GTV on multiple graphs reflecting multiple unsupervised segmentations (5.26), SegSALSA-GTV.

5.7.1 Formulation

By combining the SegSALSA formulation in (5.14) with the prior in (5.13), we obtain the following formulation

$$\widehat{\mathbf{z}}_{\text{MMAP}} = \arg \min_{\mathbf{z} \in \mathcal{R}^{K \times n}} \sum_{i \in \mathcal{S}} -\ln(\boldsymbol{p}_i^T \mathbf{z}_i) + \sum_{j=1}^{m} \lambda_G \boldsymbol{q}_j \|(\boldsymbol{A}^{\mathcal{P}_j} - \boldsymbol{I})\mathbf{z}\|_F^2, \qquad (5.26)$$
$$\text{subject to:} \quad \mathbf{z} \geq 0, \quad \mathbf{1}_K^T \mathbf{z} = \mathbf{1}_n,$$

where $\boldsymbol{A}^{\mathcal{P}_j}$ is the adjacency matrix that corresponds to the jth unsupervised segmentation $\mathcal{P}_j$, λ_{G} is the relative weight between the data term and the GTV prior, and $\boldsymbol{q}_j$ is a partition specific weight controlling the relative weight of each partition in the segmentation. This formulation is based on the definition of the GTV prior as sums of convex functions and associated linear operators. The optimization (5.26) is convex as both the GTV priors is convex on $\mathbf{z}$.

5.7.2 Optimization

The Moreau proximity operator for graph TV g_j is

$$\psi_{g_j/\mu}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{\zeta}} \lambda_{\text{G}} \mathbf{q}_j \|(\boldsymbol{A}_j - \boldsymbol{I})\boldsymbol{\zeta}\|_F^2 + \frac{\mu}{2}\|\boldsymbol{\zeta} - \boldsymbol{\nu}\|_F^2,$$

where $\boldsymbol{\nu}, \boldsymbol{\zeta} \in \mathbb{R}^{K \times n}$. This is a optimization is a quadratic problem, and we can compute the Moreau proximity operator as

$$\psi_{g_j/\mu}(\boldsymbol{\nu}) = \frac{\mu}{2\lambda_{\text{G}}\mathbf{q}_j} \underbrace{\left((\mathbf{A}_j - \mathbf{I})^T (\mathbf{A}_j - \mathbf{I}) + \frac{\mu}{2\lambda_{\text{G}}\mathbf{q}_j} \mathbf{I} \right)^{-1}}_{\mathbf{C}_j} \boldsymbol{\nu},$$

where, by construction of the partitioning graphs and respective adjacency matrices, $\mathbf{C}_j$ is a permuted block diagonal matrix, this is, $\mathbf{C}_j = \mathbf{P}_j \mathbf{B}_j \mathbf{P}_j^T$, where $\mathbf{P}_j$ is a permutation matrix that encodes the pixel membership on the partition elements, and $\mathbf{B}_j$ is a block diagonal matrix. Furthermore, we have that $\mathbf{C}_j^{-1} = (\mathbf{P}_j \mathbf{B}_j \mathbf{P}_j^T)^{-1} = \mathbf{P}_j \mathbf{B}_j^{-1} \mathbf{P}_j^T$, as $\mathbf{P}_j^{-1} = \mathbf{P}_j^T$, and that the inverse of $\mathbf{B}_j$ is the diagonal stacking of the inverse of each of the diagonal blocks.

Each diagonal block $\mathbf{B}_{j,t}$ of $\mathbf{B}_j$ corresponds to a fully connected subgraph of the partition graph with n_t nodes, a partition element with n_t pixels, and is a $n_t \times n_t$ matrix with the following

form

$$[\mathbf{B}_{j,t}] = \begin{bmatrix} b_{\mathrm{d}} & b_{\mathrm{o}} & \dots & b_{\mathrm{o}} \\ b_{\mathrm{o}} & b_{\mathrm{d}} & \dots & b_{\mathrm{o}} \\ \vdots & \vdots & \ddots & \dots \\ b_{\mathrm{o}} & b_{\mathrm{o}} & \dots & b_{\mathrm{d}} \end{bmatrix},$$

where

$$b_{\mathrm{d}} = 1 + \frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j} - \frac{1}{n_t}, \quad b_{\mathrm{o}} = -\frac{1}{n_t}.$$

This means that $\mathbf{B}^{-1}$ corresponds to

$$[\mathbf{B}^{-1}_{j,t}] = \begin{bmatrix} b'_{\mathrm{d}} & b'_{\mathrm{o}} & \dots & b'_{\mathrm{o}} \\ b'_{\mathrm{o}} & b'_{\mathrm{d}} & \dots & b'_{\mathrm{o}} \\ \vdots & \vdots & \ddots & \dots \\ b'_{\mathrm{o}} & b'_{\mathrm{o}} & \dots & b'_{\mathrm{d}} \end{bmatrix},$$

where

$$b'_{\mathrm{d}} = \frac{n_t \frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j} + 1}{n_t \frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j}(\frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j} + 1)}, \quad b'_{\mathrm{o}} = \frac{1}{n_t \frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j}(\frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j} + 1)}.$$

Thus, the proximal operator for the graph total variation,

$$\psi_{g_j/\mu}(\boldsymbol{\nu}) = \frac{\mu}{2\lambda_{\mathrm{G}}\mathbf{q}_j}\mathbf{C}_j^{-1}\boldsymbol{\nu},$$

can be pixelwise decoupled and is

$$\psi_{g_j/\mu}(\boldsymbol{\nu}_i) = \frac{1}{1 + \frac{2\lambda_{\mathrm{G}}\mathbf{q}_j}{\mu}}\boldsymbol{\nu}_i + \frac{\frac{2\lambda_{\mathrm{G}}\mathbf{q}_j}{\mu}}{1 + \frac{2\lambda_{\mathrm{G}}\mathbf{q}_j}{\mu}}\frac{\sum_{k\in\mathcal{N}_i^j}\boldsymbol{\nu}_k}{|\mathcal{N}_i^j|}, \tag{5.27}$$

where $|\mathcal{N}_i^j|$ corresponds to the number of pixels that belong to the same partition element as the ith pixel. The first term of (5.27) corresponds to the value on the ith node itself, the value of the ith pixel, and the second term corresponds to the mean of ν on the fully connected subgraph the ith node belongs to, the mean value of all pixels belonging to the same partition element the ith pixel. This operator has a complexity dominated by $O(Kn)$.

5.7.3 Algorithm

With the GTV prior, we have m terms, one for each of the graph generating unsupervised segmentations guiding the supervised segmentation.

Algorithm 4: SegSALSA-GTV

initialization:

choose $(\boldsymbol{u}_j^{f,0}, \boldsymbol{d}_j^{f,0}) \in \mathbb{R}^{n\times K}$, $j = 1, \ldots, 3$

choose $(\boldsymbol{u}_j^{g}, \boldsymbol{d}_j^{g}) \in \mathbb{R}^{n\times K}$, $j = 1, \ldots, m$

define $a_i = \frac{\boldsymbol{p}_i^T\boldsymbol{\nu}_i + \sqrt{(\boldsymbol{p}_i^T\boldsymbol{\nu}_i)^2 + \|\boldsymbol{p}_i\|^2/\mu}}{2}$, for $i \in \mathcal{S}$

define $\boldsymbol{C} = (3\mathbf{I} + m\mathbf{I})^{-1}$

set $\mu \in]0, +\infty[$

for $k = 0, 1, \ldots, k_{STOP}$ **do**

/* update **z** */

$\boldsymbol{z}^{k+1} = \boldsymbol{C}\big(\sum_{j=1}^{3}(\boldsymbol{u}_j^{f,k} - \boldsymbol{d}_j^{f,k}) + \sum_{j=1}^{m}(\boldsymbol{u}_j^{g,k} - \boldsymbol{d}_j^{g,k})\big)$

/* update split variables */

$[\boldsymbol{u}_1^{f,k+1}]_i = [\boldsymbol{z}^{k+1} - \boldsymbol{d}_1^{f,k}]_i + \frac{\boldsymbol{p}_i}{\mu a_i}$, for $i \in \mathcal{S}$

$\boldsymbol{d}_1^{f,k+1} = \boldsymbol{d}_1^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_1^{f,k+1})$

$\boldsymbol{u}_2^{f,k+1} = \left(\mathbf{I} - \frac{\mathbf{1}_K\mathbf{1}_K^T}{K}\right)(\boldsymbol{z}^{k+1} - \boldsymbol{d}_2^{f,k})\frac{\mathbf{1}_K\mathbf{1}_n^T}{K}$, $\boldsymbol{d}_2^{f,k+1} = \boldsymbol{d}_2^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_2^{f,k+1})$

$\boldsymbol{u}_3^{f,k+1} = \max\{\mathbf{0}, (\boldsymbol{z}^{k+1} - \boldsymbol{d}_3^{f,k})\}$

$\boldsymbol{d}_3^{f,k+1} = \boldsymbol{d}_3^{f,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_3^{f,k+1})$

/* graph-TV prior */

for $j = 1, \ldots, m$ **do**

for $i \in \mathcal{S}$ **do**

$[\boldsymbol{u}_j^{g,k+1}]_i = \frac{1}{1+\frac{2\lambda_{\mathrm{G}}\mathbf{q}_j}{\mu}}[\boldsymbol{z}^{k+1} - \boldsymbol{d}_j^{g,k}]_i + \frac{\frac{2\lambda_{\mathrm{G}}\mathbf{q}_j}{\mu}}{1+\frac{2\lambda_{\mathrm{G}}\mathbf{q}_j}{\mu}}\frac{\sum_{l\in\mathcal{N}_i^j}[\boldsymbol{z}^{k+1} - \boldsymbol{d}_j^{g,k}]_l}{|\mathcal{N}_i^j|}$,

$\boldsymbol{d}_j^{g,k+1} = \boldsymbol{d}_j^{g,k} - (\boldsymbol{z}^{k+1} - \boldsymbol{u}_j^{g,k+1})$

return $\boldsymbol{z}_{k+1}$

5.8 Parallelization

The SegSALSA family of algorithms is built upon two major steps:

- Update of the hidden field $\mathbf{z}$ and its associated quadratic problem;
- Update of the split variables $\boldsymbol{u}$ and its associated Moreau proximity operators.

The parallelization potential of the SegSALSA family of algorithms is dependent on the possibility to efficiently parallelize these two steps.

The step associated with the update of the hidden field is

$$\boldsymbol{z}^{k+1} = \boldsymbol{F}^{-1}\left(\sum_{j=1}^{3}(\boldsymbol{u}_j^{f,k} - \boldsymbol{d}_j^{f,k}) + \sum_{j=1}^{m}(\boldsymbol{H}_j^g)^*(\boldsymbol{u}_j^{g,k} - \boldsymbol{d}_j^{g,k})\right),$$

with

$$\boldsymbol{F} = 3\mathbf{I} + \sum_{j=1}^{m}(\boldsymbol{H}_j^g)^*\boldsymbol{H}_j^g.$$

Thus, the parallelization of this step is intrinsically connected to the structure of $\boldsymbol{F}^{-1}$, and $\boldsymbol{F}^{-1}(\boldsymbol{H}_j^g)^*$, for $j = 1, \ldots, m$. As long as these structures can be represented by filtering operations, their parallelization is trivial.

The steps associated with the computation of the Moreau proximity operators, as seen in both the formulation of SegSALSA and on the instantiations with the three priors, are can be parallelizable as the Moreau proximity operators can be pointwise decoupled.

5.9 Experimental results

To illustrate the behavior of the SegSALSA family of algorithms, we apply them to three different image classification tasks:

- Natural image segmentation;
- Hyperspectral image classification;
- Biomedical image classification.

5.9.1 Natural image segmentation

The first task corresponds to natural image segmentation. To this extent, we apply SegSALSA with VTV, GTV, and STR regularization to the task of supervised image segmentation of the Graz data set [94]. Furthermore, we also study the combination of regularizers, VTV+GTV, STR+GTV, and STR+VTV.

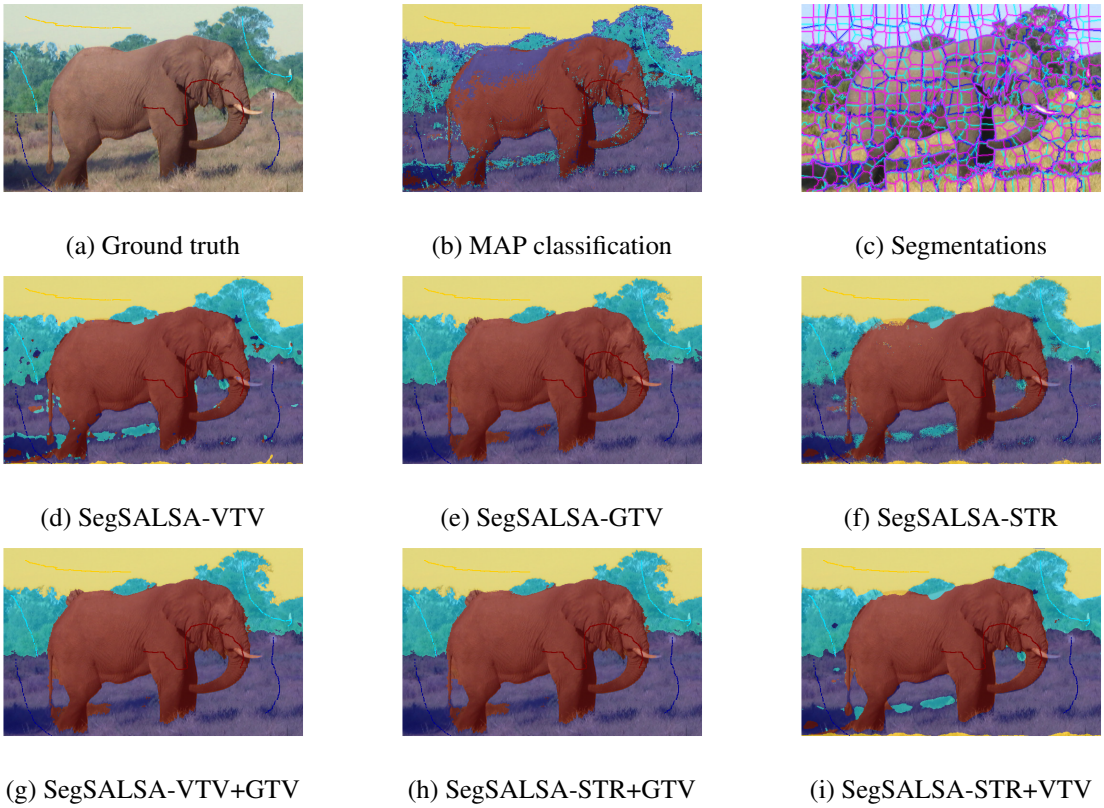
Experimental setup

The Graz data set is a collection of natural images and associated interactive multilabel segmentation. The interactive multilabel segmentations are obtained through the process of drawing class-defining scribbles over natural images. Each element of the Graz data set is composed of

three associated entities (as seen in Fig.5.2 (a) and Fig.5.3 (c)): the RGB image, the scribble, and the ground truth.

Following the approach used on [72] for comparing different methods on the Graz data set, we use as features a spatially varying estimation of the color distributions [95]. These features are tailored for interactive segmentation as they take in account the distance of each image pixel to the closest scribble.

To obtain the graph defining oversegmentations for SegSALSA-GTV, we use the SLIC [84] algorithm to obtain oversegmentations. As seen in Fig.5.2 (c) and Fig.5.3 (c), we obtain oversegmentations at multiple scales, meaning that we have multiple graphs, each at a different scale. This means that we can provide the GTV prior a multiscale flavor through the combination of multiple segmentations at different scales. We obtained the oversegmentations at multiple scales through a parameter sweep on the size of the partition elements on the SLIC algorithm.

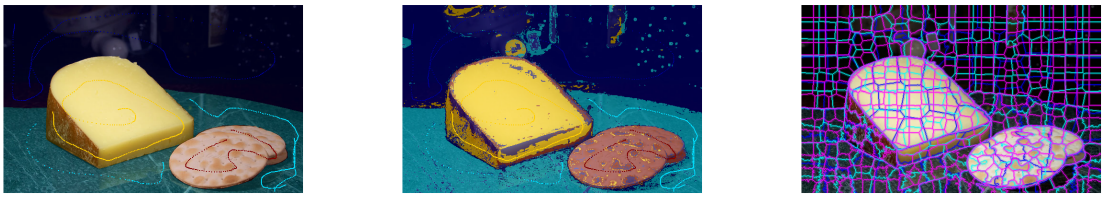


(a) Ground truth (b) MAP classification (c) Segmentations

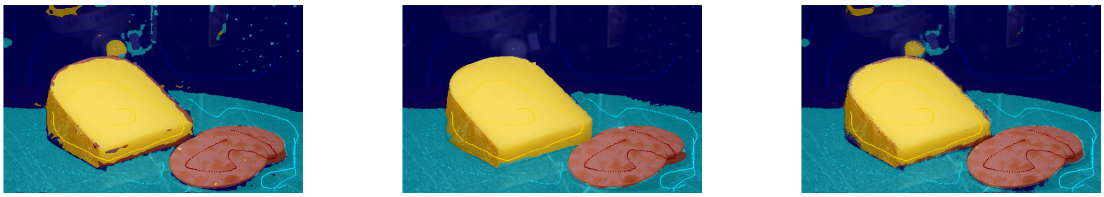
(d) SegSALSA-VTV (e) SegSALSA-GTV (f) SegSALSA-STR

(g) SegSALSA-VTV+GTV (h) SegSALSA-STR+GTV (i) SegSALSA-STR+VTV

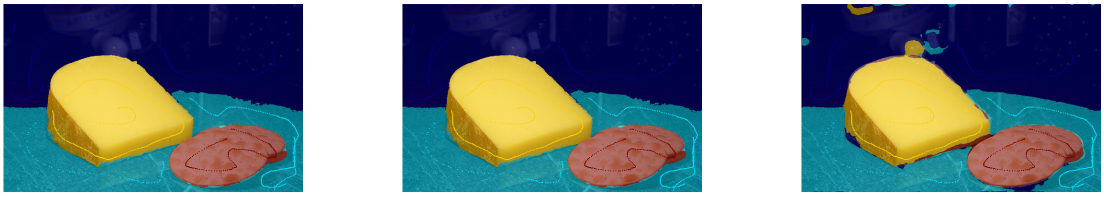
Figure 5.2: Example of supervised segmentation of natural images from the Graz dataset—elephant. (a) Original image and ground truth, (b) MAP classification (83.12% accuracy), and (c) multiple oversegmentations using SLIC. (d) Classification with VTV (93.18% accuracy), (e) classification with GTV (95.41% accuracy), and (f) classification with STR (95.41% accuracy). (g) Classification with VTV and GTV (95.61% accuracy), (h) classification with STR and GTV (95.57% accuracy), (i) classification with STR and VTV (93.70% accuracy).



(a) Ground truth (b) MAP classification (c) Segmentations



(d) SegSALSA-VTV (e) SegSALSA-GTV (f) SegSALSA-STR



(g) SegSALSA-VTV+GTV (h) SegSALSA-STR+GTV (i) SegSALSA-STR+VTV

Figure 5.3: Example of supervised segmentation of natural images from the Graz dataset — cheese. (a) Original image and ground truth, (b) MAP classification (88.77% accuracy), and (c) multiple oversegmentations using SLIC. (d) Classification with VTV (94.70% accuracy), (e) classification with GTV (98.59% accuracy), and (f) classification with STR (95.67% accuracy). (g) Classification with VTV and GTV (98.59% accuracy), (h) classification with STR and GTV (98.54% accuracy), (i) classification with STR and VTV (96.23% accuracy).

5.9.2 Hyperspectral images

The second task corresponds to hyperspectral image classification. To this extent, we apply SegSALSA-(VTV, STR, GTV) to the classification of the ROSIS Pavia University hyperspectral scene [1].

Experimental setup

The Pavia University scene (false color composition of the image in Fig. 5.4 (a)) was acquired with the ROSIS hyperspectral sensor in Pavia, Italy. The scene consists of a 610×340 pixel hyperspectral image with 103 spectral bands containing 9 not mutually exclusive classes, with the classification accuracy and classification quality being measured on those 9 classes.

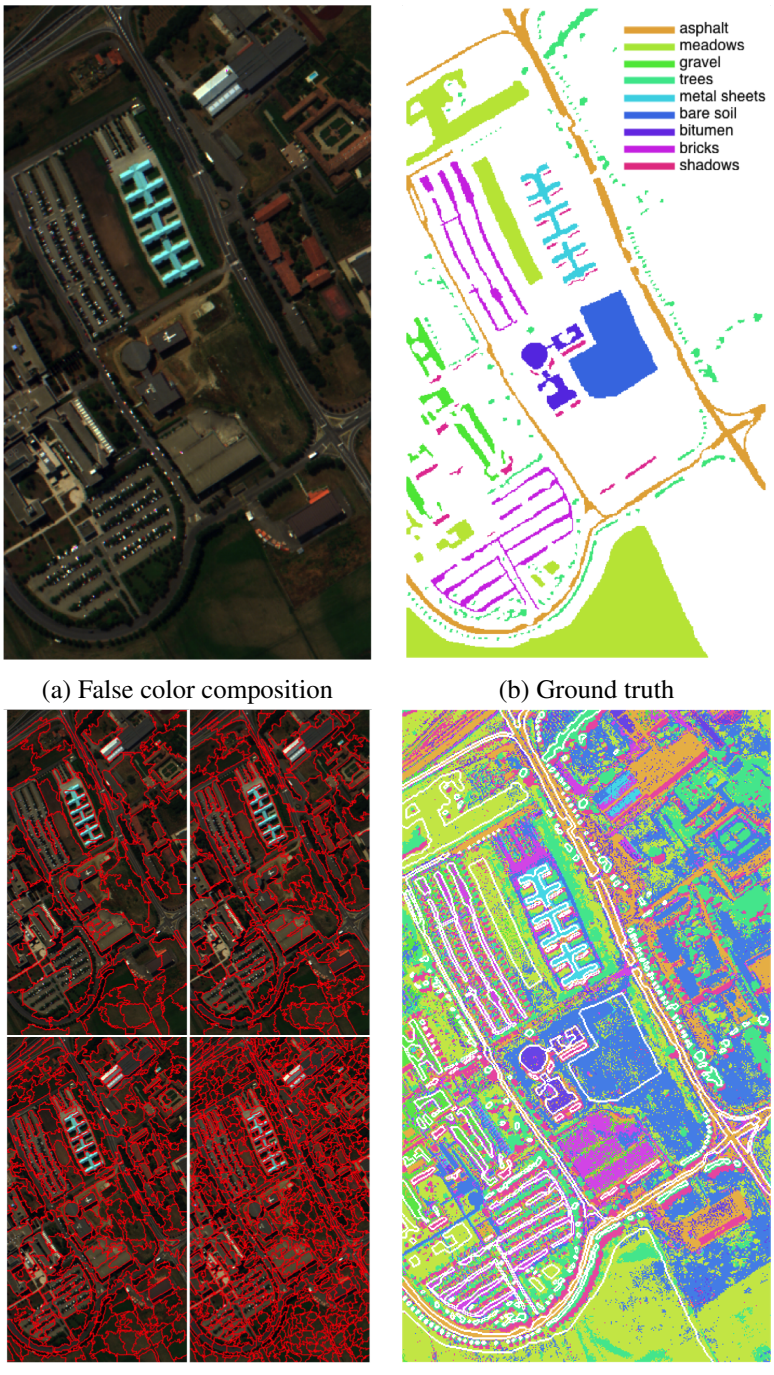
We use model the classes with a sparse Multinomial Logistic Regression (MLR) using the LORSAL algorithm [96] to estimate the class probabilities and using a Radial Basis Function (RBF) kernel, following the approach in [96]. Our training set is composed of 50 samples per class, and is used to train the LORSAL classifier. The result of the classification is present in Fig. 5.4 (d), were we achieve 83.80% overall accuracy.

As the most common superpixelization methods are tailored for segmentation of RGB images, the superpixelization is applied to the false color composition of the hyperspectral image (Fig. 5.4 (a)) instead of the original 103 spectral band hyperspectral image. To obtain the graph-defining oversegmentations for SegSALSA-GTV, we use the SLIC [84] algorithm to obtain oversegmentations. As seen in Fig.5.4 (c), we obtain oversegmentations at multiple scales, meaning that we have multiple graphs, each at a different scale.

Results and comparison

In Fig. 5.5, we show the application of the family of SegSALSA algorithms to hyperspectral image classification. We use, separately, VTV, STR, and GTV priors, and also combinations of the priors, where we have simultaneously two priors in action VTV+GTV, STR+GTV, and STR+VTV.

[1] We thank Prof. Gamba for providing the ROSIS Pavia data set to the hyperspectral community.



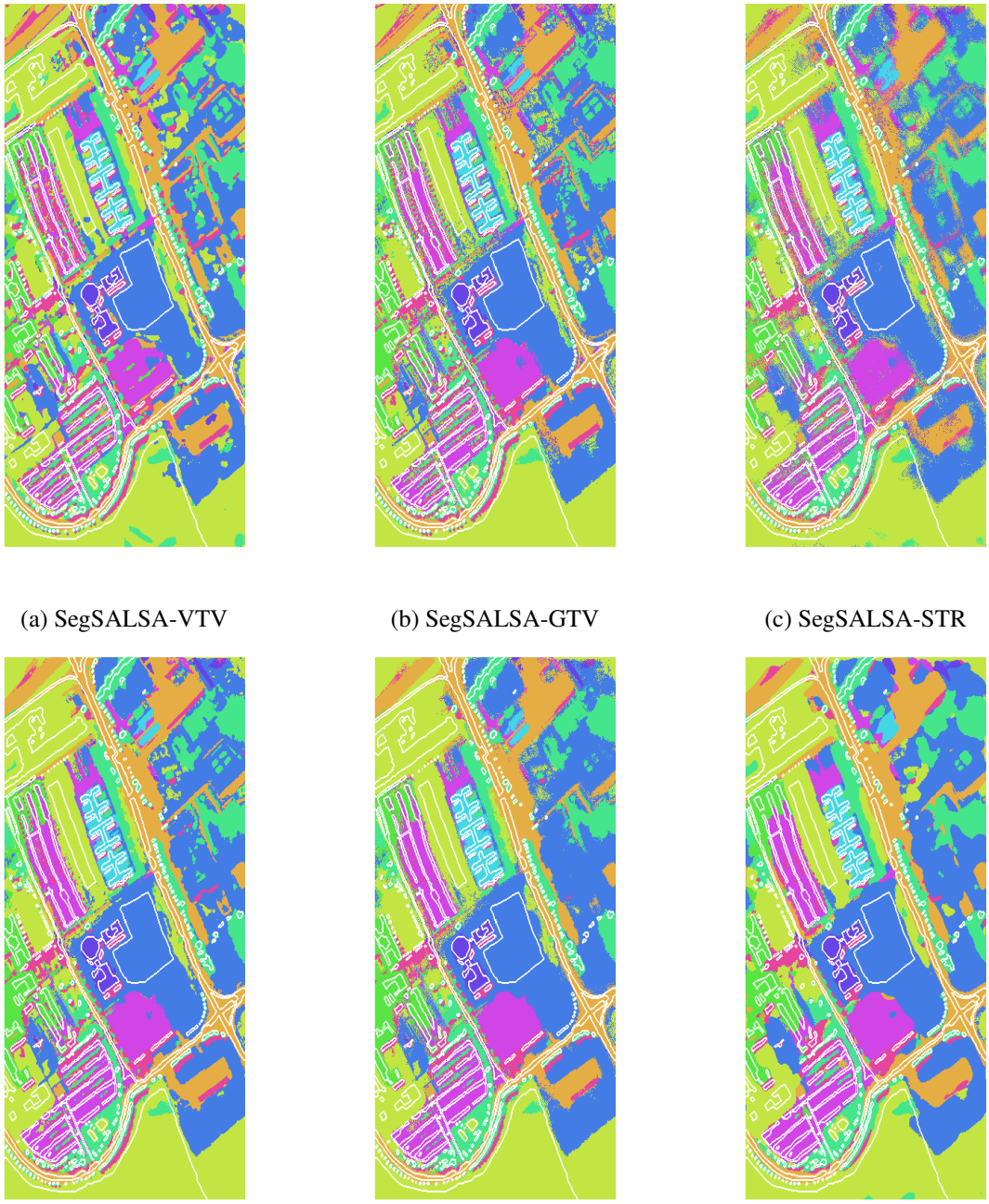
(a) False color composition

(b) Ground truth

(c) Segmentations

(d) MAP classification

Figure 5.4: Example of supervised segmentation of hyperspectral images. (a) False color composition, (b) ground truth, (c) multiple oversegmentations using SLIC, and MAP classification (83.80% accuracy).



(a) SegSALSA-VTV (b) SegSALSA-GTV (c) SegSALSA-STR

(d) SegSALSA-VTV+GTV (e) SegSALSA-STR+GTV (f) SegSALSA-STR+VTV

Figure 5.5: Application of SegSALSA to hyperspectral image classification. (a) SegSALSA-VTV (93.35% accuracy), (b) SegSALSA-GTV (94.38% accuracy), (c) SegSALSA-STR (91.45% accuracy), (d) SegSALSA-VTV+GTV (95.30% accuracy), (e) SegSALSA-GTV+STR (94.05% accuracy), (f) SegSALSA-VTV+STR (92.81% accuracy).

5.9.3 Histopathology images

The third task corresponds to the classification of histopathology images. To this extent, we apply SegSALSA-VTV, SegSALSA-GTV and a combination of both, SegSALSA-VTV+GTV, to the classification of Hematoxylin and Eosin (H&E) stained teratoma images, while exploring local and nonlocal similarity graphs.

Experimental setup

H&E stained teratoma images are acquired by use of light-field microscopy on slices of teratoma tissue. Teratomas are tumors that spawn from unregulated cell-growth through inhibition of cell signaling processes. This results in a large variety of tissues that originate from the three germ-layers (endoderm, mesoderm and ectoderm). These slices are stained with two dyes, Hematoxylin and Eosin, that have different affinities to the chemical composition of the tissues. Hematoxylin stains the tissues in a dark violet or blue color and has a high affinity to acidic and negatively charged chemical components, whereas Eosin stains the tissues in a red or pink color and has a high affinity for basic and positively charged components.

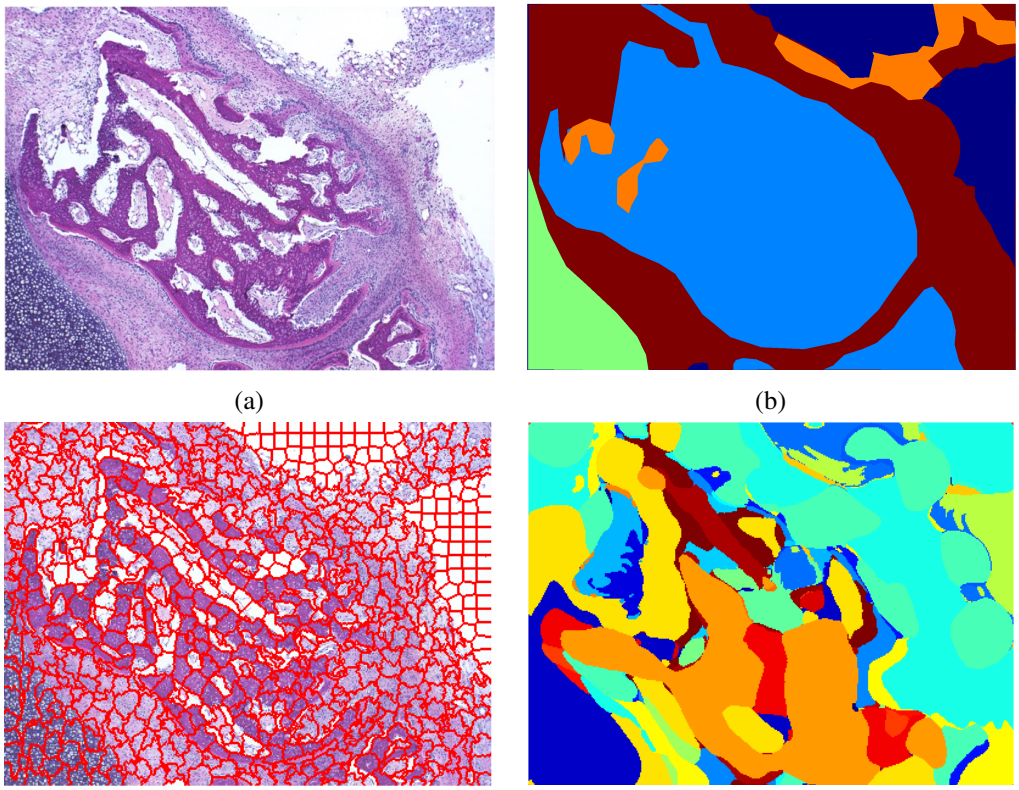
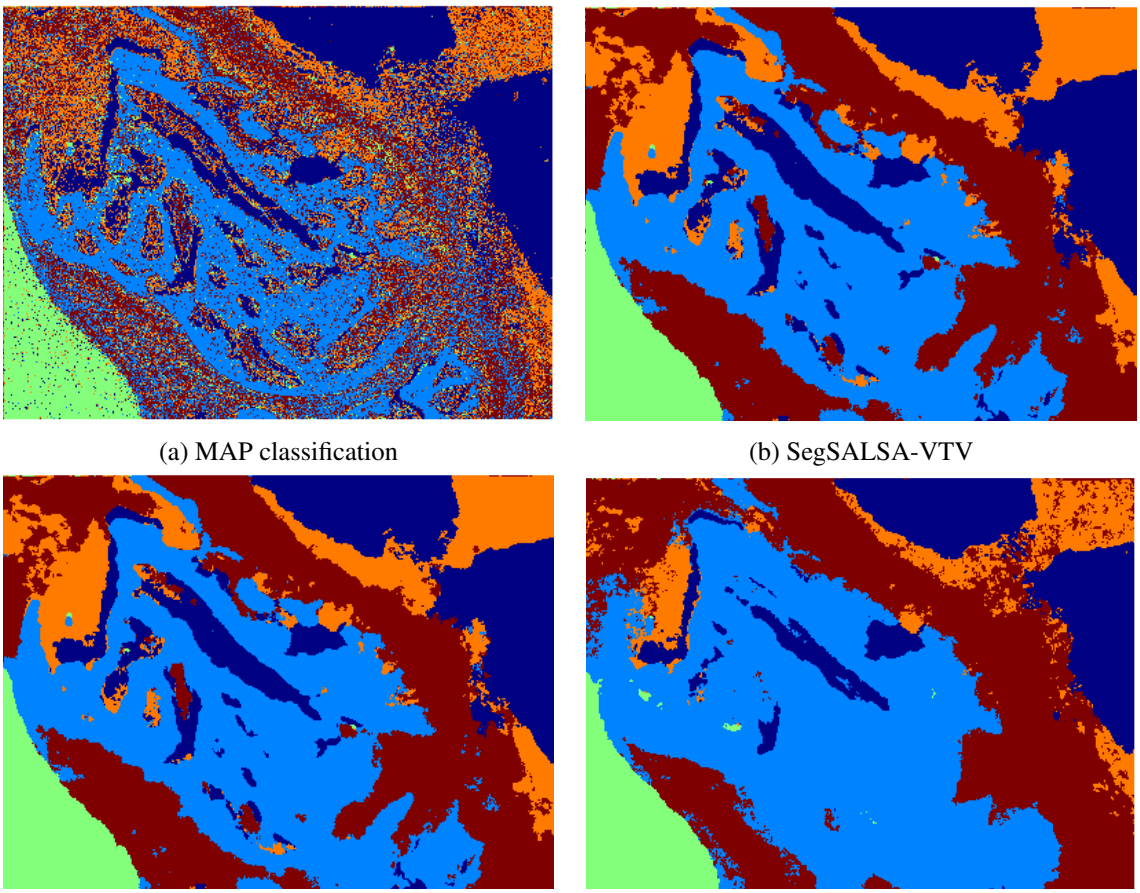


Figure 5.6: Example of supervised segmentation of histopathology data. (a) H&E stained image and (b) ground truth: background (dark blue), bone (light blue), mesenchyme (dark red), cartilage (green), and fat (orange). (c) Local segmentation with SLIC, and (d) nonlocal segmentation using the Winner Take All hash on the H&E vocabulary features.

The H&E data set we use consists of 36 images, 1200×1600 pixels, imaged at $40\times$ magnification, with the ground truth delineated by histopathologists [7][2].

Whereas significant work exists on the use of features that translate the expert knowledge of histopathologists, an H&E vocabulary [97–99], these features suffer from lack of spatial resolution. Their computation is based on very strong filtering processes that remove a significant portion of spatial information. To circumvent the loss of spatial information associated with use of the H&E vocabulary, we use as features 2×2 patches of the image. This means that instead of working with a $1200 \times 1600 \times 3$ image, we are working with a $600 \times 800 \times 12$ image. We use model the classes with a sparse Multinomial Logistic Regression (MLR) using the LORSAL algorithm [96] to estimate the class probabilities and using a Radial Basis Function (RBF) kernel, following closely the approach in [96]. Our training set is composed of 200 patches per class (from a total of 480000 patches per image), and is used to train the LORSAL classifier.



(a) MAP classification

(b) SegSALSA-VTV

(c) SegSALSA-VTV+(local)GTV

(d) SegSALSA-VTV+(local&nonlocal)GTV

Figure 5.7: Example of application of SegSALSA family of algorithms in H&E image classification. (a) MAP classification (53.35% accuracy), (b) SegSALSA-VTV (68.94%), (c) SegSALSA-VTV+(local)GTV (74.39%), and (d) SegSALSA-VTV+(nonlocal&local)GTV (76.25%).

[2]We thank Dr. John Ozolek and Dr. Carlos Castro for providing the H&E data set.

To obtain the graph-defining local oversegmentations for SegSALSA-GTV (Fig.5.6 c), we use the SLIC [84] algorithm to obtain oversegmentations.

To obtain a graph-defining nonlocal oversegmentation for SegSALSA-GTV (Fig.5.6 d), we use the H&E vocabulary features [97–99] and extract similarity using a Winner Take All (WTA) hash process [85]. With the WTA, significant performance in similarity-based methods can be achieved [100] at a very reduced computational cost. Furthermore, the WTA hash has been successfully used on unsupervised segmentation of H&E data [101].

Results and comparison

In Fig 5.7, we show the application of SegSALSA-VTV and also combinations of the VTV and GTV priors using local and nonlocal similarity graphs. From a 53.35% pixelwise accuracy obtained by the MAP classification, with no context, we are able to achieve 76.25% by using SegSALSA with VTV and GTV (local and nonlocal) priors. This illustrates the flexibility of the SegSALSA family, namely the SegSALSA-GTV algorithm, to include multiple concepts of context through the use of local and nonlocal similarity graphs.

5.10 Concluding remarks

In this chapter, we presented family of algorithms for classification with context that sidestep from the discrete nature of the optimization problems associated with context. This can be achieved with the reformulation of the MAP problem as a convex marginal MAP problem on a continuous field that drives the discrete labels. The use of hidden fields provides an additional degree of freedom on the selection of the priors. We also show three members of the SegSALSA family, according to the concept of prior applied on the hidden field: SegSALSA-VTV, SegSALSA-STR, and SegSALSA-GTV; and their application to multiple image classification applications.

Chapter 6

Classification with context and rejection

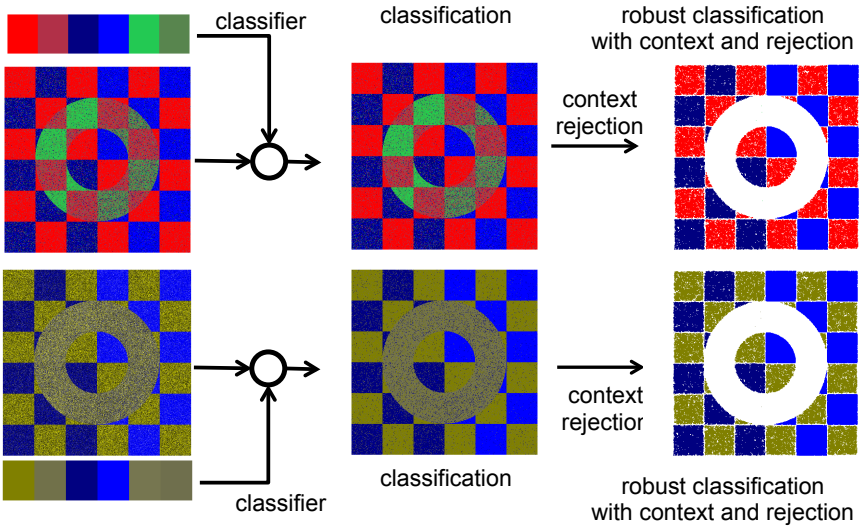


Figure 6.1: Robust classification with context and rejection of the well-posed companion problem (top) and of the ill-posed companion problem (bottom).

With classification with context described in Chapters 3 and 5, and classification with rejection described in Chapters 2 and 4, we are now ready to combine both context and rejection to achieve robust classification systems, as illustrated in Fig 6.1. In this chapter we present the general architecture for robust classification systems with context and rejection. The key feature of the architectures for robust classifiers resides in the way context and rejection interact. There are interesting trade-offs in the different architectures in terms of computational speed and ease to change the rejected fraction of robust classification, as well as in the optimality of the classifications themselves.

In Section 6.1, we present the general architecture for classification with context and rejection. In Sections 6.2 and 6.3, we describe architectures based on the joint and sequential computation of context and rejection, respectively. In Section 6.4, we reformulate the joint and sequential architecture in terms of an energy minimization problem defined on graphs and char-

acterize the behavior of the solutions. Section 6.5 concludes this chapter.

6.1 General architecture for classification with context and rejection

We illustrate the general architecture for classification with context and rejection in Fig 6.2. This

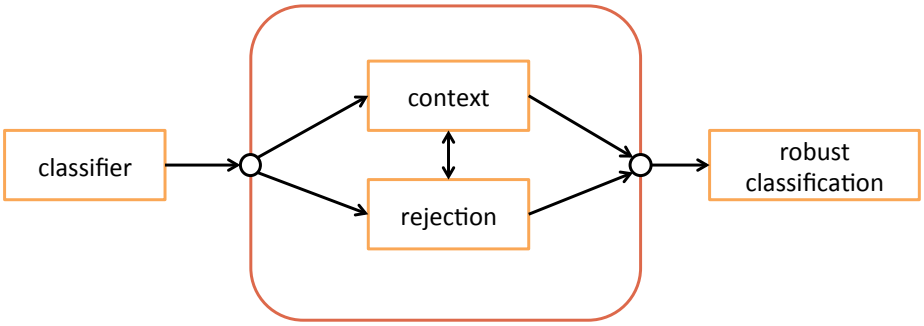


Figure 6.2: General architecture for classification with context and rejection.

general architecture can be formulated as general energy minimization problem defined on a graph that represents the structure of the image. Let $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ denote a graph, where $\mathcal{S}$ is the set of nodes (we consider at this point a one-to-one connection between graph nodes and image pixels) $\mathcal{E}$ the set of edges (encoding the structure of the image), and $\mathcal{L}'$ denote the extended set of labels that includes rejection. We can formulate the general architecture for classification with context and rejection as the following optimization

$$\arg \min_{\mathbf{y}_\mathcal{G} \in \mathcal{L}'^{|\mathcal{S}|}} E(\mathbf{y}_\mathcal{G}) = \arg \underbrace{\min_{\mathbf{y}_\mathcal{G} \in \mathcal{L}'^{|\mathcal{S}|}} E_d(\mathbf{y}_\mathcal{G})}_{\text{rejection}} + \overbrace{E_s(\mathbf{y}_\mathcal{G})}^{\text{context}}, \tag{6.1}$$

where E_d is a data fit term that associates the labeling with the output of the classifier and corresponding rejector, and E_s an interaction term that promotes context-aware labellings, for example through the promotion of spatially consistent labellings.

6.2 Joint context and rejection

The simplest form to combine context and rejection is through their joint computation, as illustrated in Fig. 6.3. This problem can be formulated as the optimization problem in 6.1. Rejection is included as an extra class, $K+1$, belonging to the extended label set $\mathcal{L}'$, and the data term is such that the energy associated with the ith node being assigned the kth label is

$$E_d(\mathbf{y}_i = k) = \begin{cases} -\ln(p(\mathbf{y}_i = k|\mathbf{x}_i)) & \text{if } k \in \mathcal{L}, \\ -\ln(\rho_i) & \text{if } k = K+1, \end{cases} \tag{6.2}$$

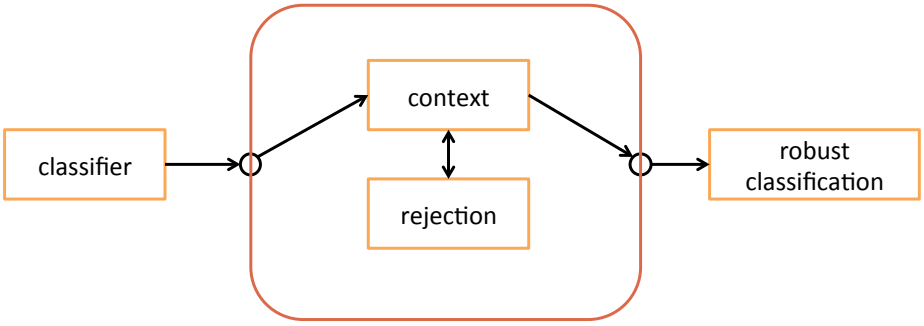


Figure 6.3: Joint architecture for classification with context and rejection.

where ρ_i denotes the likelihood that the ith node is rejected. This means that rejection is modeled as a probability of classifier failure.

As we will see in section 6.4, this formulation achieves an optimal labeling of the problem (6.1). However, a significant drawback exists: it is not possible to define *a priori* how much rejection is obtained. This means that any change in the value of rejection requires the recomputation of context. In Chapters 7 and 8 we present examples of robust classification systems based on a joint context and rejection architecture.

6.3 Sequential context and rejection

As discussed previously, the joint context and rejection approach has a significant drawback with respect to the inability to define *a priori* the amount of rejection obtained. To mitigate this, we can combine context and rejection sequentially, as illustrated in Fig. 6.4. This problem can be

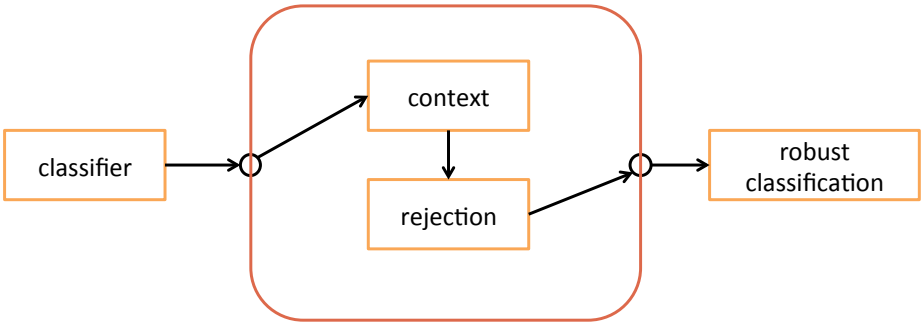


Figure 6.4: Sequential architecture for classification with context and rejection.

formulated as the following two-step optimization,

$$\hat{\mathbf{y}}_C \in \arg \min_{\mathbf{y}_\mathcal{G} \in \mathcal{L}^{|\mathcal{S}|}} E_d(\mathbf{y}_\mathcal{G}) + E_s(\mathbf{y}_\mathcal{G}), \tag{6.3}$$

$$\hat{\mathbf{y}}_{C \to R} \in \arg \min_{\mathbf{y}_\mathcal{G} \in \mathcal{M}} E_d(\mathbf{y}_\mathcal{G}) + E_s(\mathbf{y}_\mathcal{G}), \tag{6.4}$$

where $\mathcal{M} = \{\hat{\mathbf{y}}_C\} \cup \{K+1\}^{|\mathcal{S}|}$. This means that (6.3) amounts to solving the context only problem (with K labels), and (6.4) amounts to solving a rejection problem only, a binary problem where each node i can either be rejected, or maintain the label assigned in (6.3). The data fit term can be defined exactly as in (6.2).

This formulation approaches the optimization (6.1) in an approximate way, first solving a K label problem of classification with context, and then solving a binary rejection problem with no context. We derive, in section 6.4, approximation bounds that relate the sequential and the joint approaches. A significant advantage of this architecture is that the rejection amount can be defined *after* the computation of context, which means that context only has to be computed once. However, it requires a method to obtain a degree of confidence associated with classification with context for each node, which might not be achievable in many algorithms for computing classification with context. In Chapter 8, we present an algorithm for robust classification based on a sequential context and rejection architecture that uses SegSALSA (described in Chapter 5) to compute context, with the hidden field providing the confidence associated with classification with context.

6.4 Theoretical results

There is an interesting trade-off between the use of joint context and rejection architectures and sequential context and rejection architectures in terms of how the optimization (6.1) is solved. Whereas joint context and rejection solves it in a single problem, sequential context and rejection solves it in two sequential problems: first context, then rejection. In this section we derive properties of these solutions and compare their structure to the structure of a classification with context only solution.

We revisit the energy based formulation in (6.1), and define an energy function E with regard to a labeling $\mathbf{y}_{\mathcal{G}}$ on the graph $\mathcal{G}$ as:

$$E(\mathbf{y}_{\mathcal{G}}) = E_d(\mathbf{y}_{\mathcal{G}}) + E_s(\mathbf{y}_{\mathcal{G}}) = \sum_{i \in \mathcal{S}} E_d(\mathbf{y}_i) + \sum_{\{i,j\} \in \mathcal{E}} E_I(\mathbf{y}_i, \mathbf{y}_j), \tag{6.5}$$

where $E_d(\mathbf{y}_i) \geq 0$ denotes the data fit term, and $E_I(\mathbf{y}_i, \mathbf{y}_j) \geq 0$ denotes the contextual interactions between pixels i and j.

Let us consider a partition of the graph $\mathcal{G}$ in two subgraphs $\mathcal{A}$ and $\mathcal{B}$, as seen in Fig. 6.5, such that $\mathcal{G} = (\mathcal{S}, \mathcal{E}) = (\mathcal{S}_{\mathcal{A}} \cup \mathcal{S}_{\mathcal{B}}, \mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}} \cup \mathcal{E}_{\mathcal{A},\mathcal{B}})$. We have that $\mathcal{S}_{\mathcal{A}}$ corresponds to the set of nodes that belong to $\mathcal{A}$, $\mathcal{E}_{\mathcal{A}}$ is the set of edges such that both nodes belong to the set $\mathcal{S}_{\mathcal{A}}$, and $\mathcal{E}_{\mathcal{A},\mathcal{B}}$ denotes the set of all edges such that one of the nodes belongs to the $\mathcal{S}_{\mathcal{A}}$ and the other to $\mathcal{S}_{\mathcal{B}}$. We can represent the energy of a labeling $\mathbf{y}_{\mathcal{G}}$ defined on the graph $\mathcal{G}$ by the sum of the energies of its subgraphs $\mathcal{A}$ and $\mathcal{B}$,

$$\begin{aligned} E(\mathbf{y}_{\mathcal{G}}) = E(\mathbf{y}_{\mathcal{A}\cup\mathcal{B}}) = \sum_{i \in \mathcal{A}\cup\mathcal{B}} E_d(\mathbf{y}_i) + \sum_{\{i,j\} \in \mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}} \cup \mathcal{E}_{\mathcal{A},\mathcal{B}}} E_I(\mathbf{y}_i, \mathbf{y}_j) = \sum_{i \in \mathcal{A}} E_d(\mathbf{y}_i) + \sum_{i \in \mathcal{B}} E_d(\mathbf{y}_i) + \\ \sum_{\{i,j\} \in \mathcal{E}_{\mathcal{A}}} E_I(\mathbf{y}_i, \mathbf{y}_j) + \sum_{\{i,j\} \in \mathcal{E}_{\mathcal{B}}} E_I(\mathbf{y}_i, \mathbf{y}_j) + \sum_{\{i,j\} \mathcal{E}_{\mathcal{A},\mathcal{B}}} E_I(\mathbf{y}_i, \mathbf{y}_j). \end{aligned}$$

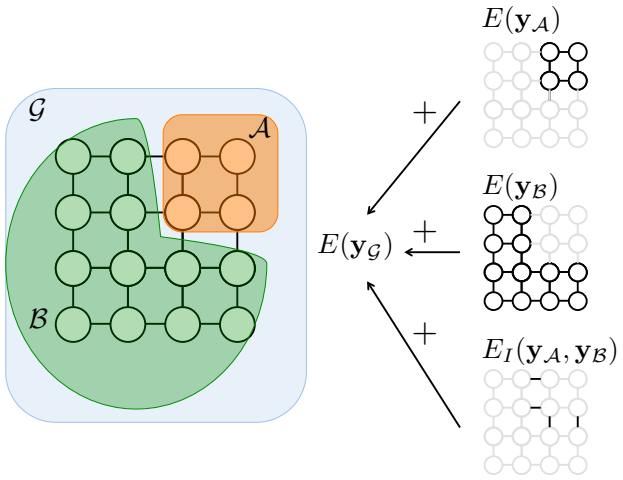


Figure 6.5: Decomposition of energy function of labeling defined on a graph $E(\mathbf{y}_{\mathcal{G}})$. The energy is the sum of the energies in subgraphs $E(\mathbf{y}_{\mathcal{A}}), E(\mathbf{y}_{\mathcal{B}})$ and in subgraph interface.

This energy can be further rearranged as a sum of the energy of the labeling on each of the subgraphs and an interface term

$$E(\mathbf{y}_{\mathcal{A}\cup\mathcal{B}}) = \underbrace{\sum_{i\in\mathcal{A}} E_d(\mathbf{y}_i) + \sum_{\{i,j\}\in\mathcal{E}_{\mathcal{A}}} E_I(\mathbf{y}_i, \mathbf{y}_j)}_{E(\mathbf{y}_{\mathcal{A}})} + \underbrace{\sum_{i\in\mathcal{B}} E_d(\mathbf{y}_i) + \sum_{\{i,j\}\in\mathcal{E}_{\mathcal{B}}} E_I(\mathbf{y}_i\, \mathbf{y}_j)}_{E(\mathbf{y}_{\mathcal{B}})} +$$
$$+ \underbrace{\sum_{\{i,j\}\mathcal{E}_{\mathcal{A},\mathcal{B}}} E_I(\mathbf{y}_i, \mathbf{y}_j)}_{E_I(\mathbf{y}_{\mathcal{A}}, \mathbf{y}_{\mathcal{B}})} = \underbrace{E(\mathbf{y}_{\mathcal{A}})}_{\text{energy subgraph } \mathcal{A}} + \underbrace{E(\mathbf{y}_{\mathcal{B}})}_{\text{energy subgraph } \mathcal{B}} + \underbrace{E_I(\mathbf{y}_{\mathcal{A}}, \mathbf{y}_{\mathcal{B}})}_{\text{interface subgraphs } \mathcal{A} \text{ and } \mathcal{B}} .$$

Let $\mathbf{y}^*$ denote a global optimal labeling of the graph $\mathcal{G} = (\mathcal{A} \cup \mathcal{B}, \mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}} \cup \mathcal{E}_{\mathcal{A},\mathcal{B}})$,

$$\mathbf{y}^* \in \arg \min_{\mathbf{y}_{\mathcal{G}} \in \mathcal{L}^{|\mathcal{A}\cup\mathcal{B}|}} E(\mathbf{y}_{\mathcal{G}}), \tag{6.6}$$

and $E(\mathbf{y}_{\mathcal{B}}|\mathbf{y}^*_{\mathcal{A}})$ denote the energy of a labeling $\mathbf{y}_{\mathcal{B}}$ on the subgraph $\mathcal{B}$ given that on the subgraph $\mathcal{A}$ we have the labeling $\mathbf{y}^*_{\mathcal{A}}$. We define a solution $\mathbf{y}^*_{\mathcal{B}}$ to be *locally optimal* in the subgraph $\mathcal{B}$ with regard to the subgraph $\mathcal{A}$ and its labeling $\mathbf{y}^*_{\mathcal{A}}$ if

$$\mathbf{y}^*_{\mathcal{B}} =\in \arg \min_{\mathbf{y}_{\mathcal{B}} \in \mathcal{L}^{|\mathcal{B}|}} E(\mathbf{y}_{\mathcal{B}}|\mathbf{y}^*_{\mathcal{A}}) = \arg \min_{\mathbf{y}_{\mathcal{B}} \in \mathcal{L}^{|\mathcal{B}|}} E(\mathbf{y}_{\mathcal{B}}) + E_I(\mathbf{y}_{\mathcal{B}}, \mathbf{y}^*_{\mathcal{A}}) \tag{6.7}$$

By decomposing the energy function across the subgraphs and the interfaces between subgraphs, we have that any globally optimal solution $\mathbf{y}^*_{\mathcal{G}}$ on a graph is also a locally optimal solution on any of its subgraph.

Lemma 1. Given a global optimal labeling $\mathbf{y}^*_{\mathcal{G}}$ on the graph $\mathcal{G}$, for any subgraph $(\mathcal{A}, \mathcal{E}_{\mathcal{A}})$ of $\mathcal{G}$, $\mathbf{y}^*_{\mathcal{G}}$ is also locally optimal in $(\mathcal{A}, \mathcal{E}_{\mathcal{A}})$ given the labeling $\mathbf{y}^*_{\mathcal{G}}$ on the rest of the graph.

Proof. Let us consider a partition of the graph $\mathcal{G}$ in two subgraphs containing the set of nodes $\mathcal{A}$ and $\mathcal{B}$, such that $\mathbf{y}_\mathcal{G} = \mathbf{y}_\mathcal{A} \cup \mathbf{y}_\mathcal{B}$. As $\mathbf{y}^*_\mathcal{G} = \mathbf{y}^*_\mathcal{A} \cup \mathbf{y}^*_\mathcal{B}$ is a globally optimal labeling, we have that

$$\begin{aligned} E(\mathbf{y}^*_\mathcal{A} \cup \mathbf{y}^*_\mathcal{B}) =& \overbrace{E(\mathbf{y}^*_\mathcal{A}) + E_I(\mathbf{y}^*_\mathcal{A}, \mathbf{y}^*_\mathcal{B})}^{E(\mathbf{y}^*_\mathcal{A}|\mathbf{y}^*_\mathcal{B})} + E_(\mathbf{y}^*_\mathcal{B}) \\ =& E(\mathbf{y}^*_\mathcal{A}|\mathbf{y}^*_\mathcal{B}) + E(\mathbf{y}^*_\mathcal{B}) \leq E(\mathbf{y}), \text{ for all } \mathbf{y} \in \mathcal{L}^{|\mathcal{A}\cup\mathcal{B}|}. \end{aligned} \tag{6.8}$$

If $E(\mathbf{y}^*_\mathcal{A}|\mathbf{y}^*_\mathcal{B})$ is **not** a local optimal labeling, then there exists a labeling $\boldsymbol{w}_\mathcal{A}$ such that

$$E(\boldsymbol{w}_\mathcal{A}|\mathbf{y}^*_\mathcal{B}) < E(\mathbf{y}^*_\mathcal{A}|\mathbf{y}^*_\mathcal{B}).$$

This means that

$$E(\boldsymbol{w}_\mathcal{A}|\mathbf{y}^*_\mathcal{B}) + E(\mathbf{y}^*_\mathcal{B}) < E(\mathbf{y}^*_\mathcal{A}|\mathbf{y}^*_\mathcal{B}) + E(\mathbf{y}^*_\mathcal{B}) \iff E(\boldsymbol{w}_\mathcal{A} \cup \mathbf{y}^*_\mathcal{B}) < E(\mathbf{y}^*_\mathcal{A} \cup \mathbf{y}^*_\mathcal{B}),$$

which contradicts (6.8). □

6.4.1 Classification with context and rejection as energy functions on a graph

Considering the set of labels $\mathcal{L} = \{1, \ldots, K\}$ and the extended set of labels $\mathcal{L}' = \{1, \ldots, K, K+1\}$ including the $K+1$ rejection class, we have that the following labellings minimize the energy function E on the graph $\mathcal{G}$.

- Classification with context

$$\hat{\mathbf{y}}_C \in \arg \min_{\mathbf{y}_\mathcal{G} \in \mathcal{L}^{|\mathcal{S}|}} E(\mathbf{y}_\mathcal{G}), \tag{6.9}$$

- Joint context and rejection

$$\hat{\mathbf{y}}_{C+R} \in \arg \min_{\mathbf{y}_\mathcal{G} \in \mathcal{L}'^{|\mathcal{S}|}} E(\mathbf{y}_\mathcal{G}). \tag{6.10}$$

Given the classification with context $\hat{\mathbf{y}}_C$ in (6.9), let $\mathcal{M} = \{\hat{\mathbf{y}}_C\}^{|\mathcal{S}|} \cup \{K+1\}^{|\mathcal{S}|}$ denote a binary search space (for each node i, either assign the label of the ith node of $\hat{\mathbf{y}}_C$ or $K+1$), we can obtain an approximation for the solution in $\mathcal{L}'$ in (6.10) by extending the solution $\hat{\mathbf{y}}_C$ as follows,

- Sequential context and rejection

$$\hat{\mathbf{y}}_{C\to R} \in \arg \min_{\mathbf{y}_\mathcal{G} \in \mathcal{M}} E(\mathbf{y}_\mathcal{G}), \tag{6.11}$$

this is, for each node i, either the label of the classification with context $\hat{\mathbf{y}}_C$ is assigned, or the node is rejected ($K+1$).

Ordering of solutions

Because the problem of classification with context and the problem of classification with context and rejection (with joint and sequential architectures) correspond to optimizations with the same objective function and nested feasibility sets, we can order the solutions according to the energy value of their optimal solution.

Theorem 3. Let $\hat{\mathbf{y}}_C$, $\hat{\mathbf{y}}_{C+R}$, and $\hat{\mathbf{y}}_{C\to R}$ denote the labellings resulting from classification with context, classification with joint context and rejection, and classification with sequential context and rejection, respectively. If the interaction between rejected and nonrejected labels is class-blind,

$$E_I(\mathbf{y}_i, K+1) = E_I(\mathbf{y}_j, K+1), \text{ for all } \mathbf{y}_i, \mathbf{y}_j \neq K+1,$$

then we have that

$$E(\hat{\mathbf{y}}_{C+R}) \leq E(\hat{\mathbf{y}}_{C\to R}) \leq E(\hat{\mathbf{y}}_C). \tag{6.12}$$

Proof. We start by showing that $E(\hat{\mathbf{y}}_{C+R}) \leq E(\hat{\mathbf{y}}_{C\to R})$. As $\hat{\mathbf{y}}_{C+R}$ is a globally optimal labeling, we have that

$$E(\hat{\mathbf{y}}_{C+R}) \leq E(\mathbf{y}), \text{ for any } \mathbf{y} \in \mathcal{L}'^{|\mathcal{S}|},$$

This means that, because $\hat{\mathbf{y}}_{C\to R} \in \mathcal{L}'^{|\mathcal{S}|}$,

$$E(\hat{\mathbf{y}}_{C+R}) \leq E(\hat{\mathbf{y}}_{C\to R}).$$

On the other hand, we have that $\hat{\mathbf{y}}_C \in \mathcal{M}^{|\mathcal{S}|}$ as $\mathcal{M} = \{\hat{\mathbf{y}}_C\}^{|\mathcal{S}|} \cup \{K+1\}^{|\mathcal{S}|}$, and that

$$E(\hat{\mathbf{y}}_{C\to R}) \leq E(\mathbf{y}), \text{ for any } \mathbf{y} \in \mathcal{M}^{|\mathcal{S}|}, \tag{6.13}$$

which leads to

$$E(\hat{\mathbf{y}}_{C\to R}) \leq E(\hat{\mathbf{y}}_C).$$

□

6.4.2 Graph partitioning - context only versus joint context and rejection

Let us consider the following partitions of the graph $\mathcal{G}$ based on the relation between the classification with context $\hat{\mathbf{y}}_C$ and the classification with joint context and rejection $\hat{\mathbf{y}}_{C+R}$:

$$\begin{aligned}
\alpha =& \{i : \hat{\mathbf{y}}_{(C+R)_i} = \hat{\mathbf{y}}_{C_i}\},\\
\beta =& \{i : \hat{\mathbf{y}}_{(C+R)_i} = K+1\},\\
\gamma =& \{i : \hat{\mathbf{y}}_{(C+R)_i} \neq K+1 \cap \hat{\mathbf{y}}_{(C+R)_i} \neq \hat{\mathbf{y}}_{C_i}\},
\end{aligned}$$

which means that α is the subset of nodes where the labels are the same for classification with context and for classification with joint context and rejection, β is the subset of nodes that are rejected by classification with joint context and rejection, and γ is the subset of nodes where

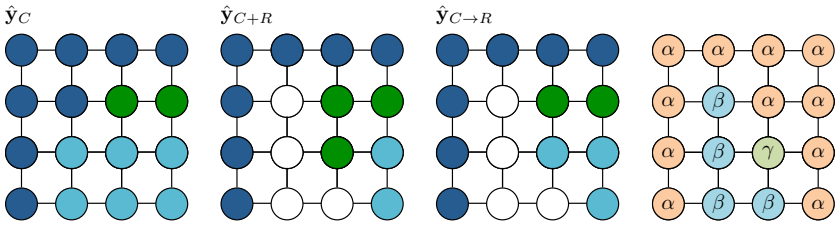


Figure 6.6: Example of the partitions of graph in (α, β, γ) according to the differences between classification with context and classification with joint context and rejection. The green node (γ) is influenced by the light-blue label in $\hat{\mathbf{y}}_C$ and in $\hat{\mathbf{y}}_{C\to R}$ and so it remains light blue, whereas in $\hat{\mathbf{y}}_{C+R}$ it is no longer influenced by the light-blue label and it changes to the green label.

the labels change between classification with joint context and classification with context and rejection, without being rejected.

The structure of the subgraph defined by the subset γ is of particular interest, as it represents the core difference between classification with joint context and rejection and classification with sequential context and rejection, as seen in Fig. 6.6. The nodes in γ correspond to nodes where rejection changes the contextual result. Whereas in classification with sequential context and rejection the nodes that are rejected influence the neighboring nodes with their previous label, in classification with joint context and rejection the nodes that are rejected influence the neighboring nodes with the rejected label.

6.4.3 Approximation bounds for classification with sequential context and rejection

The partition of the graph $\mathcal{G}$ on (α, β, γ) subgraphs allows us to bound the energy of classification with sequential context and rejection.

Theorem 4. The difference in the energy of the labellings $\hat{\mathbf{y}}_{C\to R}$ and $\hat{\mathbf{y}}_{C+R}$ is bounded by

$$0 \leq E(\hat{\mathbf{y}}_{C\to R}) - E(\hat{\mathbf{y}}_{C+R}) \leq \min\{E(\hat{\mathbf{y}}_{C_\gamma}|\hat{\mathbf{y}}_{(C+R)_\alpha}) - E(\hat{\mathbf{y}}_{(C+R)_\gamma}|\hat{\mathbf{y}}_{(C+R)_\alpha}), \\ E(\hat{\mathbf{y}}_{C_{\beta,\gamma}}|\hat{\mathbf{y}}_{(C)_\alpha}) - E(\hat{\mathbf{y}}_{(C+R)_{\beta,\gamma}}|\hat{\mathbf{y}}_{(C+R)_\alpha})\}.$$

Proof. From (6.12), we have that

$$0 \leq E(\hat{\mathbf{y}}_{C\to R}) - E(\hat{\mathbf{y}}_{C+R}) \leq E(\hat{\mathbf{y}}_C) - E(\hat{\mathbf{y}}_{C+R}) = \\ E(\hat{\mathbf{y}}_{C_{\beta,\gamma}}|\hat{\mathbf{y}}_{(C)_\alpha}) + E(\hat{\mathbf{y}}_{(C)_\alpha}) - E(\hat{\mathbf{y}}_{(C+R)_{\beta,\gamma}}|\hat{\mathbf{y}}_{(C+R)_\alpha}) - E(\hat{\mathbf{y}}_{(C+R)_\alpha}) = \\ E(\hat{\mathbf{y}}_{C_{\beta,\gamma}}|\hat{\mathbf{y}}_{(C)_\alpha}) - E(\hat{\mathbf{y}}_{(C+R)_{\beta,\gamma}}|\hat{\mathbf{y}}_{(C+R)_\alpha}).$$

Let $\hat{\mathbf{y}}'_{CR}$ denote an hybrid solution between classification with context and classification with

joint context and rejection such that

$$\hat{\mathbf{y}}'_{CR,i} = \begin{cases} \hat{\mathbf{y}}_{C+R,i} & \text{if } i \in \alpha, \\ \hat{\mathbf{y}}_{C+R,i} & \text{if } i \in \beta, \\ \hat{\mathbf{y}}_{C,i} & \text{if } i \in \gamma. \end{cases}$$

As $\hat{\mathbf{y}}'_{CR} \in \mathcal{M}^{|\mathcal{S}|}$, we have by (6.13) that

$$E(\hat{\mathbf{y}}_{C\to R}) \leq E(\hat{\mathbf{y}}'_{CR}).$$

This means that

$$\begin{aligned} E(\hat{\mathbf{y}}_{C\to R}) - E(\hat{\mathbf{y}}_{C+R}) \leq E(\hat{\mathbf{y}}'_{CR}) - E(\hat{\mathbf{y}}_{C+R}) = \\ \overbrace{E(\hat{\mathbf{y}}'_{(CR)_\gamma}|\hat{\mathbf{y}}'_{(CR)_{\alpha\cup\beta}}) + E(\hat{\mathbf{y}}'_{(CR)_{\alpha\cup\beta}})}^{E(\hat{\mathbf{y}}'_{CR})} \\ \overbrace{-E(\hat{\mathbf{y}}_{(C+R)_\gamma}|\hat{\mathbf{y}}_{(C+R)_{\alpha\cup\beta}}) - E(\hat{\mathbf{y}}_{(C+R)_{\alpha\cup\beta}})}^{-E(\hat{\mathbf{y}}_{C+R})} = \\ E(\hat{\mathbf{y}}'_{(CR)_\gamma}|\hat{\mathbf{y}}'_{(CR)_{\alpha\cup\beta}}) - E(\hat{\mathbf{y}}_{(C+R)_\gamma}|\hat{\mathbf{y}}_{(C+R)_{\alpha\cup\beta}}) = \\ E(\hat{\mathbf{y}}'_{(CR)_\gamma}|\hat{\mathbf{y}}'_{(CR)_\alpha}) + E_I(\hat{\mathbf{y}}'_{(CR)_\gamma}, \hat{\mathbf{y}}'_{(CR)_\beta}) \\ -E(\hat{\mathbf{y}}_{(C+R)_\gamma}|\hat{\mathbf{y}}_{(C+R)_\alpha}) - E_I(\hat{\mathbf{y}}_{(C+R)_\gamma}, \hat{\mathbf{y}}_{(C+R)_\beta}) = \\ E(\hat{\mathbf{y}}'_{(CR)_\gamma}|\hat{\mathbf{y}}'_{(CR)_\alpha}) - E(\hat{\mathbf{y}}_{(C+R)_\gamma}|\hat{\mathbf{y}}_{(C+R)_\alpha}), \end{aligned} \tag{6.14}$$

where (6.14) results from the definition of β as the subgraph composed of rejected nodes and from the assumption that the interaction between rejected and nonrejected labels is class-blind. □

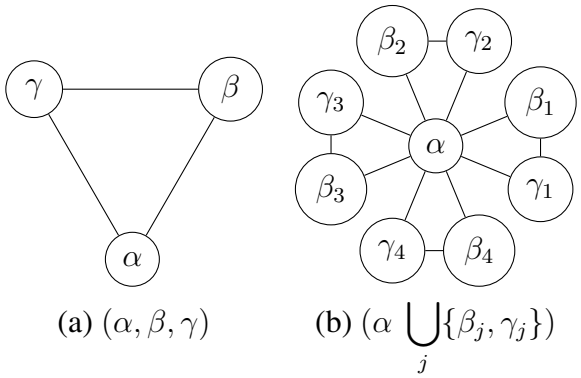
6.4.4 Structure of the (α, β, γ) partition

Let us consider a rearranging of the (α, β, γ) partition of the graph into

$$\alpha \bigcup_j \{\beta_j \, \gamma_j\},$$

as illustrated in Fig.6.7. This graph is such that $\gamma = \bigcup_j \{\gamma_j\}$ and $\beta = \bigcup_j \{\beta_j\}$. Furthermore, for all pairs of nodes (k_1, k_2) such that $k_1 \in \{\beta_i, \gamma_i\}$ and $k_2 \in \{\beta_j, \gamma_j\}$, if $i \neq j$, then there is no path between any node from k_1 and any node from k_2 that does not contain a node from α, as illustrated in Fig.6.7 (b).

With this reformulation of the (α, β, γ) we can show that, by going from classification with context $\hat{\mathbf{y}}_C$ to classification with joint context and rejection $\hat{\mathbf{y}}_{C+R}$, nodes / pixels can *only* change their label (between different labels corresponding to nonrejection) if they are in contact with nodes / pixels that will become rejected.



(a) (α, β, γ) (b) $(\alpha \bigcup_j \{\beta_j, \gamma_j\})$

Figure 6.7: Restructuring of the (α, β, γ) partition.

Theorem 5. For each $\{\beta_i, \gamma_i\}$, if γ_i is nonempty, then β_i is nonempty.

Proof. Let us assume that it exists a j such that γ_j is nonempty and β_j is empty. This means that,

$$\hat{\mathbf{y}}_{(C+R)_{\gamma_j}} \in \arg \min_{\mathbf{y}_{\gamma_j} \in \mathcal{L}'^{|\gamma_j|}} E(\mathbf{y}_{\gamma_j} | \hat{\mathbf{y}}_{(C+R)_\alpha}) =$$
$$\arg \min_{\mathbf{y}_{\gamma_j} \in \mathcal{L}'^{|\gamma_j|}} E(\mathbf{y}_{\gamma_j} | \hat{\mathbf{y}}_{(C)_\alpha}) = \hat{\mathbf{y}}_{(C)_{\gamma_j}}.$$

By definition of γ, for all node $i \in \gamma$, $\hat{\mathbf{y}}_{(C)_{\gamma_i}} \neq \hat{\mathbf{y}}_{(C+R)_{\gamma_i}}$, which is a contradiction. □

6.5 Concluding remarks

In this chapter, we presented the general architecture for robust classification system with context and rejection. The behavior of such system depends on the interaction between context and rejection. We presented two architectures for classification with context and rejection based on the joint computation of context and rejection, and based on the sequential computation of context and rejection. There are drawbacks and advantages associated with each of the formulations: a joint computation can achieve better solutions at the expense of an increase in the computation burden and difficulty to define precise values of rejected fraction, whereas a sequential computation can achieve faster solutions at the expense of solving an approximate problem. By formulating this architectures are energy optimization problems defined on a graph, we were able to provide approximation bounds for the sequential architecture, and derive the general structure of labellings resulting from the joint computation of context and rejection when compared to labellings resulting from the computation of context only.

Part III

Algorithms

Chapter 7

ICRCI algorithm

7.1 Introduction

As discussed in Chapter 1, in many classification problems, the cost of creating a training set that is statistically representative of the input dataset is often high. This is due to the required size of the training set, and the difficulty of obtaining a correct labeling resulting from unclear class separability and the possibility of presence of unknown classes. In this chapter, we are motivated by the need for automated tissue identification (classification) in images from H&E stained histopathological slides [6, 97–99]. H&E staining is used both for diagnosis as well as to gain a better understanding of the diseases and their processes, consisting of the sequential staining of a tissue with two different stains that have different chemical affinities to different tissue components.

The following characteristics of H&E image classification make this problem an ideal candidate the use of robust classification using context and rejection:

- The classification is not directly based on the observation of pixel values but on higher-level features;
- The characteristics of the image make it impossible to have access to pixelwise ground truth, leading to small, unbalanced, noisy, or incomplete training sets;
- The pixels may belong to unknown classes;
- The classification accuracy at pixels belonging to interesting or known classes is more important than the classification accuracy at pixels belonging to uninteresting or unknown classes;
- The need for high accuracy surpasses the need to classify all the samples.

7.1.1 Classification with rejection Using contextual information

The proposed framework, shown in Fig. 7.1, combines classification with rejection with classification with contextual information. This approach allows for not only rejecting a sample when the information is insufficient to classify, but also for not rejecting a sample when an "educated guess" is possible based on neighboring labels (local and nonlocal from the spatial point

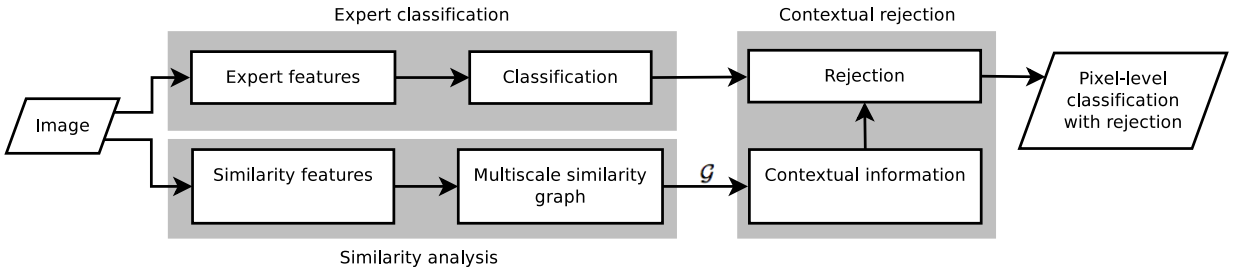


Figure 7.1: General diagram of classification system with rejection using contextual information. Each gray block is discussed in a separate section: similarity analysis in Section 7.3, expert classification in Section 7.4, and contextual rejection in Section 7.5.

of view). We do so by transforming the soft classification (posterior distributions) obtained by an expert classifier into a hard classification (labels) that considers both rejection and contextual information.

An expert classifier is designed based on application-specific features and a similarity graph is constructed representing the underlying multiscale structure of the data. The classification risk from the expert classifier is computed and the rejection is introduced as a simple classification risk threshold rule in an extended risk formulation. This formulation consists in a MAP inference problem defined on the similarity graph, thus combining rejection and contextual information.

Compared with classification with rejection only, our approach has an extra degree of complexity: the rejection depends not only on a rejection threshold for the classification but also on a rejection consistency parameter. By imposing a higher rejection consistency, the rejected samples become rejection areas (that is, a nonrejected sample surrounded by rejected samples will tend to be rejected too), which is meaningful in the task of image classification.

Compared with classification with contextual information only, this problem is of the same complexity, as the rejection can be treated as an extra class, and class-specific transitions can be easily modeled. However, as this is an approach based on the joint computation of context and rejection, changes in the amount of rejection lead to a need to recompute context.

7.1.2 Chapter outline

In Section 7.2, we describe the background for this robust classification framework: partitioning, feature extraction, and classification. In Section 7.3, we explore the similarity analysis block of the framework and the design of a multilevel similarity graph that represents the underlying structure of the data. In Section 7.4, we describe the elements of the expert classification block of the framework not described in the background. We introduce the rejection as a mechanism for handling the inability of the classifier to correctly classify all the samples. In Section 7.5, we combine the expert classification and the multiscale similarity graph in an energy minimization formulation to obtain robust classification with rejection using contextual information. In Section 7.6, we apply the framework to the classification of H&E-stained teratoma tissue images. Section 7.7 concludes this chapter.

7.2 Background

Let $\mathcal{S} = \{1, \ldots, m\}$ denote the set of pixel locations, $\mathbf{x}_i \in \mathbb{R}^d$ denote an observed vector at pixel $i \in \mathcal{S}$, $I = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \in \mathbb{R}^{d \times m}$ denote an observed image, $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ denote a partition of $\mathcal{S}$, $\mathcal{V} = \{1, \ldots, n\}$ denote a set indexing the elements of the partition $\mathcal{P}$ termed *superpixels*, and $\mathcal{E}$ denote a set indexing pairs of neighboring superpixels. Given that $\mathcal{P}$ is a partition of $\mathcal{S}$, then $\boldsymbol{p}_i \subset \mathcal{S}$, for $i \in \mathcal{V}$, $\boldsymbol{p}_i \cap \boldsymbol{p}_j = \emptyset$ for $i \neq j \in \mathcal{N}$, and $\cup_{i=1}^{n} \boldsymbol{p}_i = \mathcal{S}$. We note that in this formulation, we do not have a one-to-one connection between image pixels and graph nodes, such connection exists only between the partition elements and the graph nodes.

7.2.1 Partitioning

To decrease the dimensionality of the problem, and thus the computational burden, we partition the set of pixel locations $\mathcal{S}$ into a partition $\mathcal{P}$, allowing for the efficient use of graph-based methods. The partitioning of the image is performed by oversegmentation creating superpixels as described in [83]. This method, as is typical in most oversegmentation techniques, aims at maintaining a high level of similarity inside each superpixel and high dissimilarity between different superpixels.

Because of how the superpixels are created (measuring the evidence of a boundary between two regions), there is a high degree of inner similarity in each partition element; the elements of a superpixel will very likely belong to the same class. The major drawback of using this partitioning method is that the partition elements are highly nonuniform in terms of shape. This results of a trade-off between the regularity of the shape and the difference between the similarity inside the superpixel and dissimilarity between different superpixels.

7.2.2 Features

We use two kinds of features: (1) application-specific features encode expert knowledge and are used to classify each partition element, and (2) generic similarity features represent low-level similarities of the image and are used to assess the similarity among the partition elements. From each partition element $\boldsymbol{p}_i$, we extract statistics of the application-specific features and of the similarity features (from all pixels belonging to the same partition element), mapping from features defined on an image pixel space to features defined on an image partition space.

As application-specific features we use the *histopathology vocabulary (HV)* [97–99]. These features emulate the visual cues used by expert histopathologists, and are thus physiologically relevant. From the HV, we use nucleus size (1D), nucleus eccentricity (1D), nucleus density (1D), nucleus color (3D), red blood cell coverage (1D), and background color (3D). As similarity features we simply use the color of each partition in the RGB space. The similarity features reflect intra-tissue similarities not considered by the application-specific features.

7.2.3 Classification

Given the partition P and the associated feature matrix $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_n]$, with $\mathbf{f}_i \in \mathbb{R}^m$ the m-dimensional application-specific features, we wish to classify each partition element $\boldsymbol{p}_i \in P$ into

a single class. We do so by assigning to it a label $\mathbf{y}_i \in \mathcal{L} = \{1, \ldots, N\}$ representative of its class. This assignment is performed by computing MAP labeling

$$\hat{\mathbf{y}} \in \arg\max_{\mathbf{y} \in \mathcal{L}^n} p(\mathbf{y}|\mathbf{F}). \tag{7.1}$$

We note that under the assumption of conditional independence of features given the labels $p(\mathbf{y}|\mathbf{F}) = \prod_{i \in \mathcal{S}} p(\mathbf{y}_i|\mathbf{f}_i)$ and of equiprobable class probabilities $p(\mathbf{y}_i) = p(\mathbf{y}_j)$, for all $i, j \in \mathcal{V}$, we can reformulate the MAP formulation in (7.1) as

$$\hat{\mathbf{y}} \in \arg\min_{\mathbf{y} \in \mathcal{L}^n} \sum_{i \in \mathcal{V}} -\log p(\mathbf{y}_i|\mathbf{f}_i) - \log p(\mathbf{y}). \tag{7.2}$$

For the posterior $p(\mathbf{y}|\mathbf{F})$ we adopt the DRF model [42],

$$p(\mathbf{y}|\mathbf{F}) \propto \exp\left(-(1-\alpha) \sum_{i \in \mathcal{V}} D(\mathbf{y}_i, \mathbf{f}_i) - \alpha \sum_{\{i,j\} \in \mathcal{E}} V_{\{i,j\}}(\mathbf{y}_i, \mathbf{y}_j) \right), \tag{7.3}$$

where $-D(\mathbf{y}_i, \mathbf{f}_i)$ is the *association potential* (corresponding data-fit energy term E_d)), which links discriminatively the label $\mathbf{y}_i$ with the feature vector $\mathbf{f}_i$, $-V_{\{i,j\}}(\mathbf{y}_i, \mathbf{y}_j)$ is the *interaction potential* (corresponding to the interaction potential E_I), which models the spatial contextual information, and $\alpha \in [0, 1]$ is a regularization parameter that controls the relative weight of the two potentials. The posterior (7.3) is a particular case of the DRF class introduced in [42], because the association potential does not depend on the partition elements. The DRF model used constitutes an excellent trade-off between model complexity and goodness of the inferences, as shown in Section 7.6.

To completely define (7.3), we need to specify the association potential D and the interaction potential $V_{\{i,j\}}$. In this framework, we start from the assumption that $D(\mathbf{y}_i, \mathbf{f}_i) = -\log p(\mathbf{y}_i|\mathbf{f}_i, \mathbf{W})$, resulting from (7.2) and (7.3), where $p(\mathbf{y}_i|\mathbf{f}_i, \mathbf{W})$ is the MLR [102] parameterized with the matrix of regression coefficients $\mathbf{W}$, and $-V_{\{i,j\}}(\mathbf{y}_i, \mathbf{y}_j) = w_{ij}\delta_{\mathbf{y}_i, \mathbf{y}_j}$, where $w_{ij} \geq 0$ is a weight to be defined later, and $\delta_{i,j}$ is the Kronecker delta function. This class of association potentials, which define a MLL-MRF prior [55], promotes neighboring labels of the same class.

Multinomial logistic regression

Let $k(\mathbf{f}) = [k_0(\mathbf{f}), \ldots, k_q(\mathbf{f})]^T$ denote a vector of nonlinear functions $k_i : \mathbb{R}^m \to \mathbb{R}$, for $i = 0, \ldots, q$, with q the number of training samples and with $k_0 = 1$. The MLR models the *a posteriori* probability of $\mathbf{y}_i \in \mathcal{L}$ given $\mathbf{f} \in \mathbb{R}^m$ as

$$p(\mathbf{y}_i = l|\mathbf{f}, \mathbf{W}) = \frac{e^{\mathbf{w}_l^T k(\mathbf{f})}}{\sum_{j=1}^{N} e^{\mathbf{w}_j^T k(\mathbf{f})}}, \tag{7.4}$$

where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N] \in \mathbb{R}^{(q+1) \times N}$ the matrix of regression coefficients. Given that $p(\mathbf{y}_i|\mathbf{f}, \mathbf{W})$ is invariant with respect to a common translation of the columns of $\mathbf{W}$, we arbitrarily set $\mathbf{w}_N = 0$.

Learning the regression coefficients W

Our approach is supervised; we can thus split the dataset into a training set $\{(\mathbf{y}_i, \mathbf{f}_i),\, i \in \mathcal{T}\}$, where $\mathcal{T} \subset \mathcal{V}$ is a set indexing the labeled superpixels, and the testing set $\{\mathbf{f}_i,\, i \in \mathcal{V} - \mathcal{T}\}$.Based on these two sets and on the DRF model (7.3), we can infer matrix $\mathbf{W}$ jointly with the MAP labeling $\hat{\mathbf{y}}$.

Aiming at a lighter procedure to learn the matrix $\mathbf{W}$, we adopt the *sparse multinomial logistic regression* (SMLR) criterion introduced in [103], which, fundamentally, consists in setting $\alpha = 0$ in (7.3), disconnecting the interaction potential, and computing the MAP estimate of $\mathbf{W}$ based on the training set and on a Laplacian independent and identically distributed prior for the components of $\mathbf{W}$. We are then led to the optimization

$$\widehat{\mathbf{W}} \in \arg\max_{W}\; l(\mathbf{W}) + \log p(\mathbf{W}), \tag{7.5}$$

with $l(\mathbf{W}) = \sum_{i \in \mathcal{T}} \log p(\mathbf{y}_i | \mathbf{f}_i, \mathbf{W})$ the log-likelihood, and $p(\mathbf{W}) \propto e^{-\lambda \|\mathbf{W}\|_{1,1}}$ the prior, where λ is the regularization parameter and $\|\mathbf{W}\|_{1,1}$ denotes the sum of the ℓ_1 norm of the columns of the matrix $\mathbf{W}$. The prior $p(\mathbf{W})$ promotes sparsity in the components of $\mathbf{W}$, avoiding overfitting and improving the generalization capability of the classifier, mainly when the training set is small [103]. The sparsity level is controlled by the parameter λ.

LORSAL

We use the LORSAL algorithm (see [96]) to solve the optimization (7.5), by approximating $l(\mathbf{W})$ by a quadratic upper bound [102] and solving the sequence of ℓ_2-ℓ_1 minimization problems with the ADMM [104].

Given the training set, a RBF is a possible choice of function in the vector of nonlinear regression function k used in (7.4), which allows us to obtain a training kernel (computed by a RBF kernel of the training data). This allows us to deal with features that are not linearly separable. With both the regressor matrix $\mathbf{W}$ and the nonlinear regression function k defined, we obtain the class probabilities from the MLR formulation in (7.4).

7.2.4 Computing the MAP labeling

From (7.3), we can write the MAP labeling optimization as

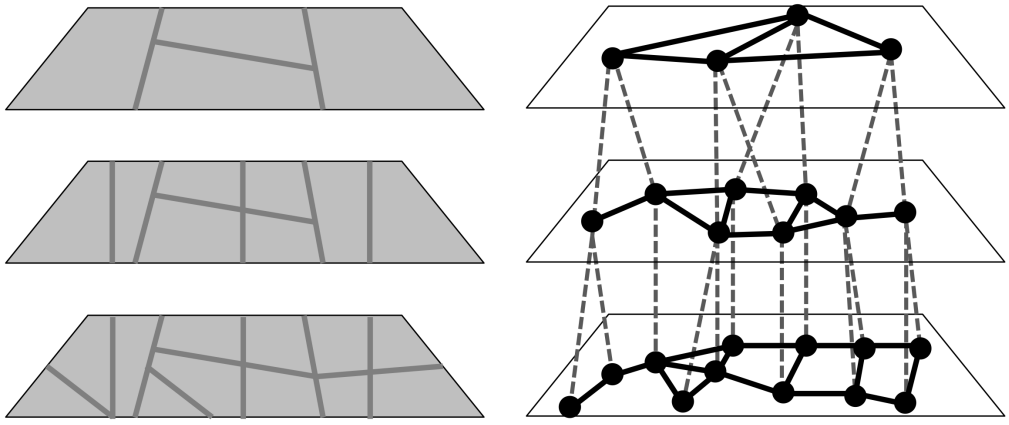
$$\arg\min_{\mathbf{y} \in \mathcal{L}^n}\; (1-\alpha) \sum_{i \in \mathcal{V}} D(\mathbf{y}_i) + \alpha \sum_{\{i,j\} \in \mathcal{E}} V_{\{i,j\}}(\mathbf{y}_i, \mathbf{y}_j). \tag{7.6}$$

This is an integer optimization problem, which, as discussed in Chapter 3 is NP-hard for most interaction potentials promoting piecewise smooth segmentations.

We thus find an approximate solution to this problem by using the graph-cut α-expansion algorithm [47, 51, 69] described in section 3.5.1. With the constraint that $V_{\{i,j\}}$ is metric in the label space, the local minimum found by α-expansion is within a known factor of the global minimum of the labeling.

7.3 Similarity analysis

Similarity analysis is the first step (see Fig. 7.1) of the proposed approach. To represent similarities in the image, we construct a similarity multiscale graph by (a) partitioning the image at different scales and (b) finding both local and multiscale similarities. The partitioning of the image at each scale is computed from the oversegmentation that results from using superpixels [83]. The different scales used for partitioning reflect a compromise between computational cost associated with large multiscale graphs, and the performance gains achieved by having a multiscale graph that correctly represents the problem. The construction of a similarity multiscale graph (as exemplified in Fig. 7.2) allows us to encode local similarities at the same scale, and similarities at different scales. The edges of the similarity multiscale graph define the cliques present in (7.3). This knowledge can be used to improve the performance of the classification, as neighboring and similar partitions are likely to belong to the same class.



(a) Multiscale partition (b) Multiscale graph

Figure 7.2: Example of multiscale partition and resulting multiscale graph. Nodes are denoted by circles, intrascale edges by dashed gray lines, interscale edges by black lines, and scale by the planes.

7.3.1 Multiscale superpixels

We obtain a multiscale partitioning of the image by computing superpixels at different scales, that is, selecting increasing minimum superpixel sizes (MSS) for each superpixelization. This leads to multiple partitions in which the minimum number of pixels in each partition element is changed, corresponding to a scale of the partition. The scale selection must achieve a balance between spatial resolution and representative partition elements (with sufficient size to compute the statistics on the features).

7.3.2 Design of the similarity multiscale graph

The design of the similarity multiscale graph is performed in three steps:

1. compute a graph for each single scale partition;
2. connect the single scale partition graphs;
3. compute similarity-based edge weight assignment and prune edges.

The main idea is that a partition will have an associated graph. By combining partitions with different scales (an inverse relation exists between the number of elements of a partition of an image and the scale associated with that partition), we are able to combine graphs with different scales. This is the fundamental ideal of the multiscale graph.

Single scale graph as a subgraph of the multiscale graph

Let us consider $\mathcal{P}_s(I) = \bigcup_i \{\boldsymbol{p}_i^s\}$, the set of partition elements $\boldsymbol{p}_i^s$ obtained by partitioning of the image I at scale s. We associate a node n_i^s to each partition element $\boldsymbol{p}_i^s \in \mathcal{P}_s(I)$ and defined the set of nodes at scale s as

$$\mathcal{V}_s = \bigcup_i \{n_i^s\}.$$

There is a one-to-one correspondence between partition elements $\boldsymbol{p}_i^s$ and nodes n_i^s. For each pair of adjoint partition elements (partition elements that share at least one pixel at their boundary) at scale s, $(\boldsymbol{p}_i^s, \boldsymbol{p}_j^s)$, we create an undirected edge between the corresponding nodes. We have that the set of intrascale edges at scale s is

$$\mathcal{E}_s = \bigcup_i \bigcup_{j \in \mathcal{N}(n_i^s)} \{(n_i^s, n_j^s)\},$$

where $\mathcal{N}^s(n_i^s)$ is the set of neighbor nodes of n_i^s, that is, the set of nodes that correspond to the partitions adjoint to the partition $\boldsymbol{p}_i^s$. Let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ denote the graph associated to scale s. The union, for all scales, of the single scale graphs, that is,

$$\bigcup_s \mathcal{G}_s = \bigcup_s (\mathcal{V}_s, \mathcal{E}_s)$$

is itself a graph that represents the multiscale partitioning of the image, without edges existing between nodes at different scales.

Multiscale edge creation

The multiscale graph is obtained by extending the union of all single-scale graphs $\bigcup_s \mathcal{G}_s$ to include interscale edges. For $s' > s$, let $\eta(n_i^s, s')$ be a function returning a node at scale s' such that, for $j = \eta(n_i^s, s')$, we have $\boldsymbol{p}_j^{s'} \cap \boldsymbol{p}_{n_i}^s \neq \emptyset$; that is, $j = \eta(n_i^s, s')$ is a node at scale s' whose corresponding partition element $\boldsymbol{p}_j^{s'}$ has non empty intersection with the partition element $\boldsymbol{p}_{n_i}^s$. Based on this construction, a partition element cannot be related to two or more different larger scale partition elements but can be related to multiple lower level partition elements. Let $\mathcal{E}_{(s,s+1)}$ be the set of edges between nodes in $\mathcal{V}_s$ and $\mathcal{V}_{s+1}$; we have that

$$\mathcal{E}_{(s,s+1)} = \bigcup_i \bigcup_{j=\eta(n_i^s, s+1)} \{(n_i^s, n_j^{s+1})\}.$$

The set $\mathcal{E}_{(s,s+1)}$ contains edges between adjacent scales, connecting the finer partition at a lower scale to the coarser partition that a higher scale. A node at scale s has exactly one edge connecting to a node at scale $s+1$ and at least one edge connecting to a node at scale $s-1$.

Considering a set of scales $\mathcal{S}$, we have that the multiscale graph $\mathcal{G}$ resulting from the multiscale partitioning is

$$\mathcal{G} = \left(\underbrace{\bigcup_{s=1}^{|\mathcal{S}|} \mathcal{V}_s}_{\text{nodes}}, \underbrace{\left(\bigcup_{s=1}^{|\mathcal{S}|} \mathcal{E}_s \right)}_{\text{intrascale edges}} \cup \underbrace{\left(\bigcup_{s=1}^{|\mathcal{S}|-1} \mathcal{E}_{(s,s+1)} \right)}_{\text{interscale edges}} \right) = (\mathcal{V}, \mathcal{E}) \,.$$

Edge weight assignment

Given the multiscale graph $\mathcal{G}$, we now compute and assign edge weights based on similarity. Let $f_{\text{si}}(n_i^s)$ be a function that computes similarity features on the node n_i^s, corresponding to the partition element $\boldsymbol{p}_i^s$. The weight of the edge $(n_i^s, n_j^{s'}) \in \mathcal{E}$ is computed as

$$w_{n_i^s, n_j^{s'}} \propto v(s, s') \exp\left(-\| f_{\text{si}}(n_i^s) - f_{\text{si}}(n_j^{s'}) \|^2 / \gamma\right), \tag{7.7}$$

where γ is a scale parameter, $\exp\left(-\| f_{\text{si}}(n_i^s) - f_{\text{si}}(n_j^{s'}) \|^2 / \gamma\right)$ quantifies the similarity between two nodes n_i^s and $v(s, s') = v_{\text{intrascale}}$, if $s = s'$, and $v(s, s') = v_{\text{intercale}}$, if $s \neq s'$. The rationale for different weights for intrascale and interscale edges comes from the different effect of the multiscale structure. For a given value of intrascale weight, lower values of the interscale edge weight downplay the multiscale effect in the graph, and higher values of the interscale edge weight accentuate the multiscale effect.

7.4 Expert classification

The expert classification block of the system is constructed from two sequential steps: feature extraction and classification. The feature extraction step consists in computing the application-specific features and extracting statistics of the features on each of the lowest level partitions. In the classification step, the classifier is trained, applied to the data, and the classification risk is computed. As the feature extraction procedure was introduced in Section 7.2.2, and the classification procedure was described in Section 7.2.3, we will focus on the computation of the classification risk.

7.4.1 Rejection by risk minimization

By approaching classification as a risk minimization problem, we are able to introduce rejection. To improve accuracy at the expense of not classifying all partitions, we classify while rejecting. Let $\mathcal{L}' = \mathcal{L} \cup \{K+1\}$ be an extended set of partition class labels with an extra label. The rejection class can be considered as an *unknown* class that represents the inability of the classifier to correctly classify all samples. The extra label $K+1$ corresponds to this rejection class.

Classification with rejection by risk minimization

Given a feature vector $\mathbf{f}_i$, associated to a partition element $\boldsymbol{p}_i$, and the respective (unobserved) label $\mathbf{y}_i \in \mathcal{L}$, the objective of the proposed classification with rejection is to estimate $\mathbf{y}_i$, if the estimation is reliable, and do nothing (rejection) otherwise.

To formalize the classification with rejection, we introduce the random variable $\hat{\mathbf{y}}_i \in \mathcal{L}'$, for $i \in \mathcal{V}$, where $\hat{\mathbf{y}}_i = K+1$ denotes rejection. In addition, let us define a $(K+1) \times K$ cost matrix $C = [c_{j_1,j_2}]$ where the element c_{j_1,j_2} denotes the cost of deciding that $\hat{\mathbf{y}}_i = j_1$, when we have $\mathbf{y}_i = j_2$ and does not depend on $i \in \mathcal{V}$.

Let the classification risk of $\hat{\mathbf{y}}_i = k$ conditioned to $\mathbf{f}_i$ be defined as:

$$\begin{aligned} R(\hat{\mathbf{y}}_i = k|\mathbf{f}_i) &= \mathbb{E}_{\mathbf{y}_i}[c(\hat{\mathbf{y}}_i = k, \mathbf{y}_i)|\mathbf{f}_i] \\ &= \sum_{j_2=1}^{K} c_{k,j_2} p(\mathbf{y}_i = j_2|\mathbf{f}_i, \widehat{\mathbf{W}}). \end{aligned}$$

By setting $c_{K+1,j_2} = \rho$, we get

$$\begin{aligned} R(\hat{\mathbf{y}}_i = k, k \neq K+1|\mathbf{f}_i) &= \sum_{j_2=1}^{N} c_{k,j_2} p(\mathbf{y}_i = j_2|\mathbf{f}_i, \widehat{\mathbf{W}}), \\ R(\hat{\mathbf{y}}_i = k, k = K+1|\mathbf{f}_i) &= \rho. \end{aligned} \tag{7.8}$$

By minimizing (7.8) over all possible partition labellings $\mathcal{L}'^{|\mathcal{S}|}$, we obtain

$$\widehat{\mathbf{y}} \in \arg \min_{\mathbf{y} \in \mathcal{L}'^{|\mathcal{V}|}} \sum_{i \in \mathcal{V}} R(\mathbf{y}_i|\mathbf{f}_i). \tag{7.9}$$

Note that if $c_{j_1,j_2} = 1 - \delta_{j_1 - j_2}$, where δ_n is the Kronecker delta function, minimizing (7.9) yields

$$\hat{\mathbf{y}}_i \in \begin{cases} \arg\max\limits_{\mathbf{y}_i \in \mathcal{L}} p(\mathbf{y}_i|\mathbf{f}_i, \widehat{\mathbf{W}}), & \max\limits_{\mathbf{y}_i \in \mathcal{L}} p(\mathbf{y}_i|\mathbf{f}_i, \widehat{\mathbf{W}}) > 1 - \rho; \\ K+1, & \text{otherwise.} \end{cases}$$

In other words, if the maximum element of the estimate of the probability vector is large, we are reasonably sure of our decision and assign the label as the index of the element; otherwise, we are uncertain and thus assign the unknown-class label.

Including expert knowledge

Expert knowledge can be included in the risk minimization. Class labels can be grouped in L superclasses $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_L\}$ (each super class is an element of the partition of the set of classes $\mathcal{L}$) in which misclassification within the same superclass should have a cost different than misclassifications within different superclasses.

Let us now consider the following cost elements with a cost g for misclassification within the same superclass,

$$c'_{j_1,j_2} = \begin{cases} 0 & \text{if } j_1 = j_2; \\ g & \text{if } j_1 \text{ and } j_2 \text{ belong to the same superclass;} \\ 1 & \text{otherwise.} \end{cases}$$

The expected risk considering expert knowledge of selecting the class label $\mathbf{y}_i \in \mathcal{L}'$ in the partition is

$$R'(\hat{\mathbf{y}}_i = k, k \neq K+1|\mathbf{f}_i) = \sum_{j_2=1}^{N} c'_{k,j_2} p(\mathbf{y}_m = j_2|\mathbf{f}_i, \widehat{\mathbf{W}}),$$
$$R'(\hat{\mathbf{y}}_i = k, k = K+1|\mathbf{f}_i) = \rho. \tag{7.10}$$

Minimizing (7.10) over all possible partition labelings yields

$$\hat{y}'_i \in \arg\min_{\mathbf{y}\in\mathcal{L}'^{|\mathcal{V}|}} \sum_{i\in\mathcal{V}} R'(\mathbf{y}_i \mid \mathbf{f}_i, \widehat{\mathbf{W}}).$$

This formulation allows us to include expert knowledge in the assessment of a risk of assigning a label.

7.5 Robust classification with context and rejection

7.5.1 Problem formulation

We formulate the problem of robust classification with context and rejection as a risk minimization problem defined over the similarity multiscale graph $\mathcal{G}$.

As shown in (6.1) and in (7.6), we can pose the robust classification problem as an energy minimization problem of two potentials over the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ representing the multiscale partitioning of the image I. The association potential D is the data term, the interaction potential $V_{\{i,j\}}$, for $(i,j) \in \mathcal{E}$, is the contextual term, and $\alpha \in [0,1]$ is a weight factor that balances the relative weight between the two is denoted as contextual index. Then,

$$\hat{y} \in \arg\min_{\mathbf{y}\in\mathcal{L}'^{|\mathcal{V}|}} (1-\alpha) \sum_{i\in\mathcal{V}} D(\mathbf{y}_i, \mathbf{f}_i) + \alpha \sum_{(i,j)\in\mathcal{E}} V_{\{i,j\}}(\mathbf{y}_i, \mathbf{y}_j). \tag{7.11}$$

7.5.2 Association potential: expert knowledge

The association potential measures the disagreement between the labeling and the data; we formulate it as a strictly increasing function of the classification risk in (7.10):

$$D(\mathbf{y}_i, \mathbf{f}_i) = \log(R'(\mathbf{y}_i \mid \mathbf{f}_i, \hat{\mathbf{W}})), \text{ for } i \in \mathcal{V}.$$

This unary association potential is associated with the nodes $\mathcal{V}$ of the graph (partitions), and includes the rejection that is present in the classification risk R'.

7.5.3 Interaction potential: similarity

The interaction potential is based on the topology of the graph $\mathcal{G}$, combining intra and inter level interactions between the pairs of nodes connected by edges, based on their similarity. We define

an interaction function ψ that enforces piecewise smooth labeling among the pairs of nodes connected by edges.

In the design of the similarity multiscale graph, the difference between intralevel and interlevel edges is encoded in different multiplier constants of the edge weight (7.7). This allows us to work with intralevel and interlevel edges in the same way, without increasing the complexity of the pairwise potential. Accordingly, we set

$$V_{\{i,j\}}(\mathbf{y}_i, \mathbf{y}_j) = w_{i,j}\psi(\mathbf{y}_i, \mathbf{y}_j),$$

where $w_{i,j}$, for $(i,j) \in \mathcal{E}$, corresponds to the edge weight defined in (7.7).

Interaction function

The interaction function ψ enforces piecewise smoothness in neighboring partitions; its general form is $\psi(\mathbf{y}_i, \mathbf{y}_j) = 1 - \delta_{\mathbf{y}_i - \mathbf{y}_j}$, that is 0 if $\mathbf{y}_i = \mathbf{y}_j$ and 1 otherwise.

It is desirable, however, both to ease the transition into and out of the rejection class, and ease the transitions between classes belonging to the same superclass. We achieve this by adding a superclass consistency parameter ψ_C and a rejection consistency parameter ψ_R to the interaction potential as follows:

$$\psi(\mathbf{y}_i, \mathbf{y}_j) = \begin{cases} 0 & \text{if } \mathbf{y}_i = \mathbf{y}_j; \\ \psi_C & \text{if } \mathbf{y}_i \text{ and } \mathbf{y}_j \text{ belong to the same superclass;} \\ \psi_R & \text{if } \mathbf{y}_i = K+1 \text{ or } \mathbf{y}_j = K+1; \\ 1 & \text{otherwise.} \end{cases} \tag{7.12}$$

Defining a rejection consistency parameter ψ_R allows us to have an interaction function that can be metric, meaning that the interaction potential will be metric. Another effect is the ability of controlling the structure of the rejected area. With a rejection consistency parameter close to 0 we obtain a labeling with structure with unstructured rejection; this means that rejection areas can be spread in the image and can consist of one partition element only. With a higher value, we are imposing structure both on the labeling but also on the rejection areas, leading to larger and more compact rejection areas.

7.6 Experimental results

We now illustrate the advantaged of using a robust classification scheme combining rejection and context in the classification of H&E stained teratoma tissue images. We also explore the joint interaction between context and rejection in the classification problem, and the behavior of the framework as the difficulty of the classification problem increases.

As the concept of combining classification with context with classification with rejection in pixelwise image classification is novel, there are no competing methods nor frameworks to compare to. To provide an assessment of the performance of the framework, we compare the performance of the framework with the performance of context only, and with the performance of rejection only, with selection of optimal rejected fractions, determined in an ideal setting.

7.6.1 H&E data set

Our H&E data set consists of 36 1600×1200-size images of H&E stained teratoma tissue slides imaged at $40\times$ magnification containing 20 classes; Fig. 7.5 shows three examples.

Experimental setup

The statistic extracted for the application-specific (HV features) and the similarity features (RGB color), on the lowest level of the partition, consists of the sample mean of the feature values on the partition. It is a balance between good classification performance, low feature dimensionality, and low complexity. This results in 10 dimensional application-specific feature vectors, and 3 dimensional similarity feature vectors. The superclasses are constructed from the germ layer (endoderm, mesoderm, and ectoderm). Classes corresponding to tissues derived from the same germ layer will belong to the same superclass.

The multiscale similarity graph is built with six scales with a MSS of (100, 200, 400, 800, 1600, 3200) for each of the layers of the similarity graph. This provides a compromise between the computational burden associated with large similarity graphs and the performance increase obtained. The results we present with six scales are marginally better than the ones achieved with five or seven scales.

Parameter analysis

In this section we analyze the impact of the contextual index α and the rejection threshold ρ. The contextual index α describes the contextual information; $\alpha = 0$ means no contextual information and $\alpha = 1$ means no classification information is taken in account. The rejection threshold ρ denotes our confidence in the classification result; lower values of ρ denote low confidence in classification and higher values of ρ denote high confidence in classification.

To evaluate the parameters, we define two types of training sets, based on the origin of the training samples: (1) A single image training set composed of k samples S_k, extracted from a test image. This training set is used to train the classifier and is applied to the entire image. (2) A training set $S_{k,k}$ containing k training samples from each image of a given set. This training set is used to evaluate the classifier in situations in which we have no knowledge about the tissues. Note that each of the 36 H&E images not only contains a different set of tissues, but was also potentially stained and acquired using different experimental protocols, with no guarantee of normalization of the staining process.

The remaining parameters are set empirically according to the experts. The interscale ($v_{\text{interscale}}$) and intrascale ($v_{\text{intrascale}}$) weights for the similarity graph construction are set to 4 and 1, respectively, to achieve a "vertical" consistency in the multiscale classification. Larger values of the interscale when compared to the intrascale will enforce a higher multiscale effect on the segmentation: the different layers of the graph will be more similar to each other.

The regularization parameter of the classifier λ is set to 10 as it maximizes the overall classification accuracy (66.4%) in the entire data set with a $S_{75,75}$ training set. The superclass misclassification cost g is set to 0.7; the superclass consistency ψ_c and rejection consistency ψ_r are set to 0.7 and 0.5, respectively, to ease transitions into same superclass tissues and rejection, and to maintain a metric interaction potential. Larger values of the superclass consistency ψ_c lead to

smaller borders (in length) between elements of the same superclass, and smaller values lead to larger borders. The value of the rejection consistency ψ_r affects the length of the border of the rejected areas (their perimeter): smaller values of ψ_r lead to disconnected rejected areas (with a large perimeter), usually thin rejection zones between two different classes, whereas larger values of ψ_r lead to connected rejected areas (with a small perimeter), usually rejection blobs that reject an entire area. To achieve similar levels of rejected fraction, the rejection threshold ρ must accommodate the value of the ψ_r as larger values of ψ_r mean more costly rejection areas.

Effect of contextual index, and rejection threshold in the classification performance The inclusion of rejection in the classification leads to problems in the measurement of the performance of the classifier. As the accuracy is measured only on the nonrejected samples, it is not a good index of performance (the behavior of the classifier can be skewed to a very large reject fraction that will lead to nonrejected accuracies close to 1). To cope with this, we use the quality of classification Q, as defined in 4.4. The intuition being that, with the initial classifier parameters set to maximize the overall classification accuracy, by maximizing Q, we maximize both the number of accurately classified samples not rejected and the number of misclassified samples rejected. By varying the value of the contextual index α in (7.11), we are weighting differently the role of contextual information in the classification. For $\alpha = 0$, no contextual information is used, whereas for $\alpha = 1$, the problem degenerates into assigning a single class to the entire image. By varying the value of the rejection threshold ρ in (7.8), we are assigning different levels of confidence to the classifier, *i.e.*, $\rho = 0$ is equivalent to no confidence in the classifier (reject everything), whereas $\rho = 1$ is equivalent to total confidence in the classifier (reject nothing).

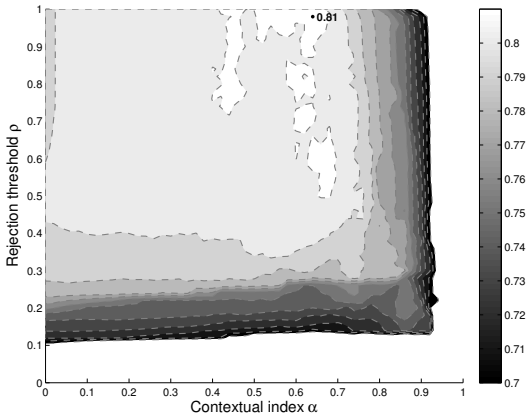
As the contextual index α and the rejection threshold ρ interact jointly, we now analyze the classification quality Q for different situations.

We test with three single image training sets S_{60}, S_{120}, S_{240}, corresponding roughly to using 1.5%, 3% and 6% of the samples of the image. We test with an entire data set training set $S_{60,60}$, in which only 3% of the data set is composed of samples from the test image. For each type of training set, we use as test images each of the 36 images of the data set, presenting the mean value of Q.

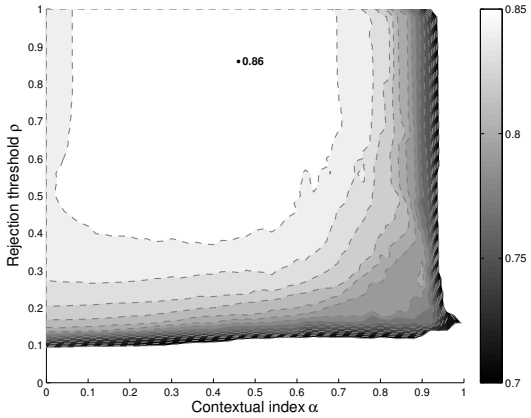
From Figure 7.3, we can observe the variation of the performance of the classifier with α and ρ for different situations. The change from (a) to (c) corresponds to an increase in the dimension of the training set. Both the improvement of the maximum value and the shift to lower values of contextual index and higher values of rejection threshold can be explained by increasing performance of the classification. This means that a more reliable classification is available, decreasing the need to use contextual information and rejection. On the other hand, (d) corresponds to an extreme situation in which the training set is highly noisy, with only 3% of samples belonging to the test image. The high dependency of contextual information in this case is clear. The maximum value of Q is attained at lower values of the rejection threshold and higher values of the contextual index.

Parameter selection

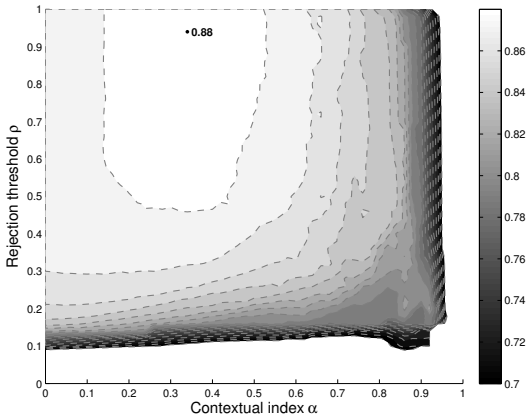
As seen in Figure 7.3, the quality of classification varies with the type of applications; applications for which the training set is easier will lead to lower reliance on contextual information



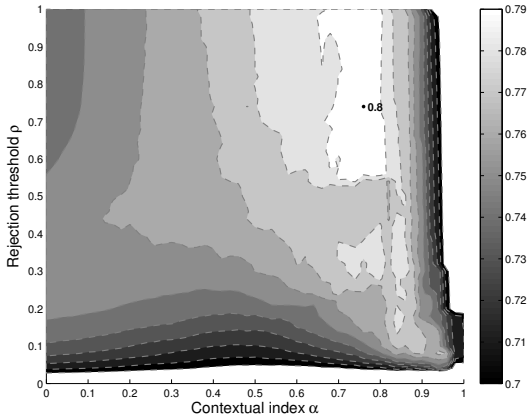
(a) Q for S_{60} (max. Q 0.81).



(b) Q for S_{120} (max. Q 0.86).



(c) Q for S_{240} (max. Q 0.88).



(d) Q for $S_{60,60}$ (max. Q 0.80).

Figure 7.3: Variation of quality of classification Q with the contextual index α and the rejection threshold ρ for four different training sets. Adjacent contour lines correspond to a 0.01 variation of Q. It is clear a shift to lower dependency on rejection and contextual information as the size of the training set, and consequently the classifier performance, increases.

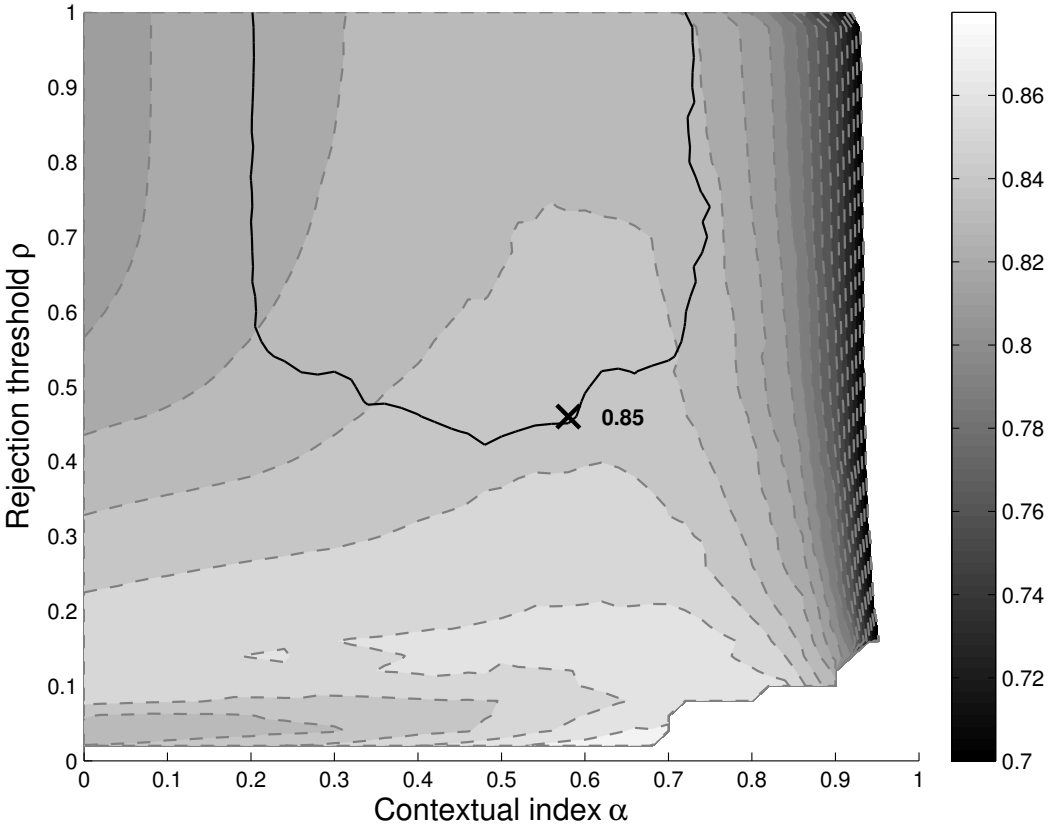
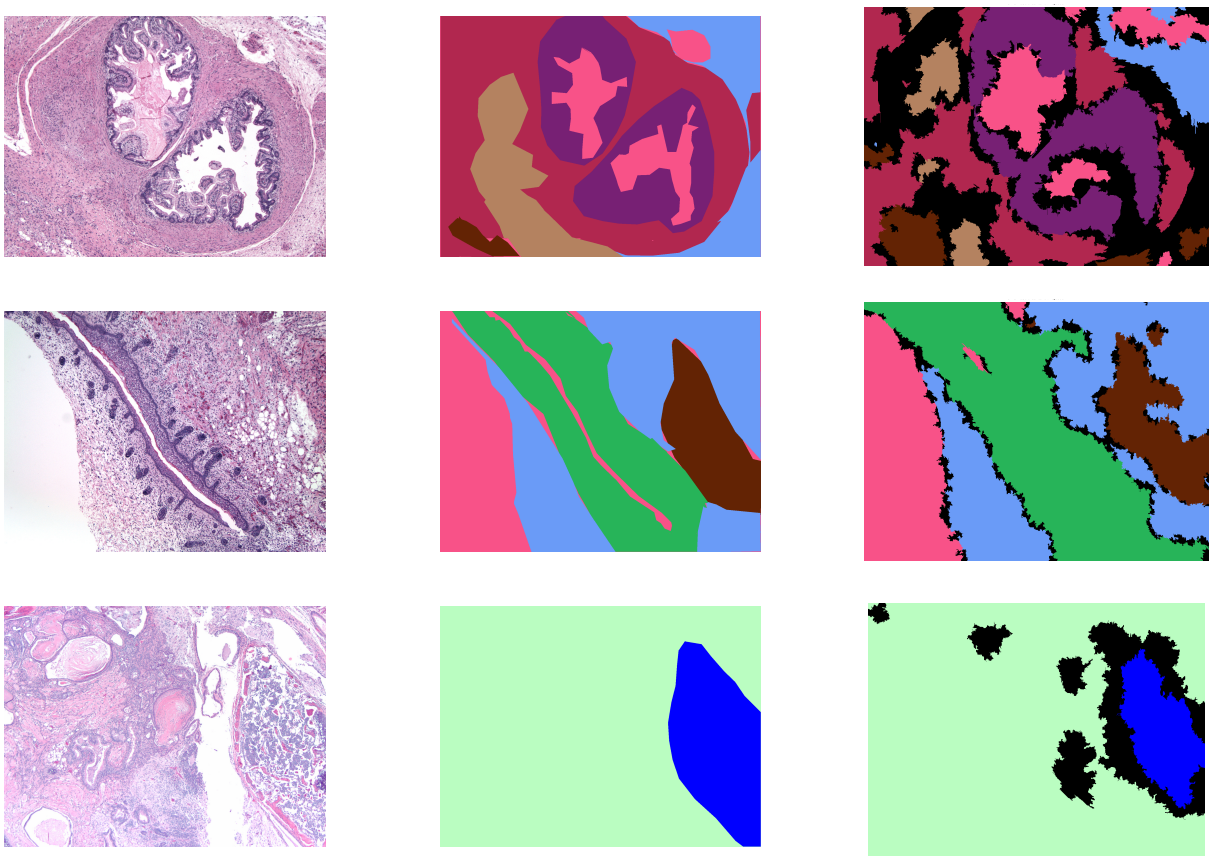


Figure 7.4: Variation of nonrejected accuracy with the contextual index α and rejection threshold ρ. The dark line corresponds to the level set of quality of classification Q equal to 99% of its maximum value. The maximum nonrejected accuracy is 85%, corresponding to $\rho = 0.46$ and $\alpha = 0.58$. The corresponding rejection fraction r is 4.6%.

and rejection, and harder training sets will lead to the opposite. In order to select a single set of parameters, we combine the results of the four different training sets for each of the 36 images, obtaining the average of the classification quality Q and nonrejected accuracy for the resulting 4×36 instances. Our motivation for the selection of the parameters is to maximize the accuracy of the nonrejected fractions within a zone of high classification quality. To do so, we select the region of high values of Q (Q larger than 99% of its maximum value). Then we select the parameters that maximize the nonrejected accuracy, as seen in Figure 7.4.

Results

We present results of our method on a set of 3 images from the data set containing a different number of classes (as seen in Figure 7.5). The classifications are obtained with different training sets to illustrate different challenges. In image 1, to create a small and nonrepresentative training



(a) Original image. (b) Ground truth. (c) Classification result.

Figure 7.5: Example of classification results for H&E stained samples of teratoma imaged at $40X$ containing multiple tissues: Image 1 (first row) background (light pink), smooth muscle (dark pink), gastro intestinal (purple), mature neuroglial (light brown), fat (dark brown); mesenchyme (light blue); Image 2 (second row) background (light pink), fat (dark brown), mesenchyme (light blue), skin (green); Image 3 (third row) mesenchyme (light green); bone (dark blue). Rejected partitions are shown in black. The training set consists of: 5 randomly chosen partitions per class (roughly 0.6% of total) for image 1, 120 randomly chosen partitions (roughly 3% of total) for image 2, 20 randomly chosen partitions (roughly 0.5% of total) for image 3.

Table 7.1: Classification and rejection performance metrics for the example images in Figure 7.5. Classification with rejection and context (white background), classification with context without rejection (green background), classification without context with optimal rejection (red background), and classification without context and without rejection (brown background).

Image	Nonrejected accuracy	Rejected fraction	Rejected quality	Classification quality	Accuracy with no rejection
Classification with rejection and context					
1	0.701	0.347	3.37	0.662	0.600
2	0.891	0.067	10.11	0.868	0.862
3	0.967	0.140	9.97	0.866	0.937
Classification with rejection without context					
1	0.702	0.370	3.90	0.673	0.582
2	0.878	0.031	9.69	0.868	0.863
3	0.936	0.000	3920	0.936	0.935

set, the training set is composed of 5 randomly chosen partition elements per class (roughly 0.6% of total). In image 2, to create a representative training set, the training set is composed of 120 randomly chosen partition elements from the entire image (roughly 3% of total). In image 3, to create a small representative training set with high class overlap, the training set is composed of 20 randomly chosen partition elements from the entire image (roughly 0.5% of total).

We analyze both overall results (in Table 7.1) and class-specific results (in Table 7.2). The computation of the rejection quality is based on the results of classification with contextual information and no rejection (*i.e.* comparing the labeling with rejection to the labeling resulting from setting the reject threshold ρ to 1 in (7.11)).

In Table 7.1, we compare the performance of classification with contextual information and rejection with context only (obtained by setting $\rho = 1$) and with classification with rejection only with optimal rejected fraction (obtained by sorting the partition elements according to maximum *a posterior* probability and selecting the rejected fraction that maximizes the classification quality).

Comparing the performance results of classification with rejection using contextual information (white background in Tab. 7.1) with the results of classification with context only (red background in Tab. 7.1), the improvement in classification accuracy at the expense of introducing rejection is clear. For images 1 and 2, this can be achieved at levels of classification quality higher than accuracy of context only, meaning that we are rejecting misclassified samples at a proportion that increases the number of correct decisions made (the underlying concept of classification quality). For image 3, due to the high accuracy of context only (and of the classification with no context and no rejection, brown background in Tab. 7.1), the increase in accuracy is at the expense of rejecting a comparatively large proportion of accurately classified samples, leading to a smaller value of classification quality.

Comparing the performance results of classification with rejection using contextual infor-

Table 7.2: Class-specific results for the example images in Figure 7.5.

Tissue type	Train samples	Test samples	Rejected samples	Rejected quality	Nonrejected accuracy	Classification quality	Accuracy no rejection
			Image 1				
Other	5 (1.2%)	410	134 (32.7%)	0.77	0.72	0.55	0.75
Fat	5 (8.5%)	54	1 (1.9%)	0.00	0.94	0.93	0.93
Gastro.	5 (0.5%)	1036	170 (16.4%)	3.90	0.91	0.83	0.86
Smt. muscle	5 (0.4%)	1283	529 (41.2%)	1.81	0.69	0.64	0.58
Mesenchyme	5 (1.1%)	454	174 (38.3%)	4.04	0.53	0.66	0.38
Mat. neuro.	5 (1.3%)	369	143 (38.8%)	1.82	0.35	0.53	0.29
			Image 2				
Other	30 (3.3%)	885	24 (2.7%)	5.80	0.91	0.90	0.90
Fat	13 (2.5%)	510	48 (9.4%)	4.54	0.77	0.75	0.74
Skin	36 (3.0%)	1157	37 (3.2%)	20.35	0.98	0.96	0.97
Mesenchyme	41 (3.1%)	1268	127 (10.0%)	6.17	0.86	0.83	0.82
			Image 3				
Bone	2 (0.3%)	725	246 (33.9%)	1.60	0.75	0.64	0.69
Mesenchyme	18 (0.6%)	3195	319 (10.0%)	11.27	1.00	0.91	0.99

mation with the results of classification with rejection only with optimal rejected fraction (red background in Tab. 7.1) the results are comparable for images 1 and 2, meaning we can achieve a performance improvement similar to the achieved by rejection with optimal rejected fraction through the introduction of context. For image 3, due to the high accuracy of classification with no context and no rejection (brown background in Tab. 7.1), the optimal rejected fraction is 0, meaning that the increased accuracy is at the expense of rejecting a comparatively large proportion of accurately classified samples.

Analyzing the classification in Fig. 7.5, the effects of combining rejection with contextual information are clear. We obtain significant improvements for image 1 by combining classification with context with classification with rejection in terms of classification quality and nonrejected accuracy, thus revealing the potential of combining classification with rejection with classification with context. For image 2, only the class boundaries are rejected, leading to high values of overall rejection quality and class-specific rejection quality. In image 3, it is clear the effect of noisy training sets (due to the image characteristics), where a significant amount of the class boundaries are rejected, and the classification quality is lower than the accuracy of the original classification with no context and no rejection.

Finally, we point to the usefulness of the classification quality Q. By analysis of the classification quality, it is possible to compare the performance of the classifier with rejection in different situations and note how the performance will decrease as the complexity of the problem increases (by increasing the number of classes).

7.7 Concluding remarks

We proposed a robust classification system using context and rejection for the automated classification of histopathology images. Furthermore, we are able to impose spatial constraints on the rejection itself departing from the current standard of image classification with rejection. These encouraging results point towards potential application of this method in large-scale automated tissue identification systems of histological slices.

Chapter 8

SegSALSA-R algorithm

8.1 Introduction

Supervised image classification is pivotal in a large number of hyperspectral image applications [52]. As discussed in Chapter 1, the problem of hyperspectral image classification is generally ill-posed. This leads to the need for robust classification systems in hyperspectral image classification. We design the robust classification systems by combining two key features: classification with context through the use of contextual information, and classification with rejection.

Contextual information is used in image classification as a regularizer to impose desired characteristics in the resulting classification, for example through the use of multilevel logistic priors based on MRF [55], widely used in hyperspectral image classification [105], or graph-based methods [106, 107]. Whereas there are alternatives to supervised hyperspectral image classification, such as curve fitting of absorption bands [108], the need for contextual information based regularization is still present. By itself, however, contextual information does not totally remove the effects of classification errors associated with overlapping classes, small or incomplete training sets, and the existence of unknown classes.

Classification errors can be mitigated if we adapt the behavior of the classifier to avoid classifying samples (pixels in the case of images) with high potential for incorrect classifications. This can be achieved by equipping the classifier with rejection, thus obtaining an increase in classification performance at the expense of not classifying the entire image.

In this framework, we combine classification with rejection with classification with context in two different ways, corresponding to two different instantiations of the general scheme in Fig. 8.1:

- Joint computation of Context and Rejection (JCR) as in [8], where rejection is considered as an extra class and computed alongside with context;
- Sequential computation of Context and Rejection (SCR) as in [9], where rejection is computed after the context by use of a rejection field.

We extend and compare these two different formulations for supervised hyperspectral image classification with rejection.

This chapter is organized as follows: Section 8.2 describes our classification method with rejection and context, with Section 8.2.1 corresponding to JCR and Section 8.2.2 to SCR. Section

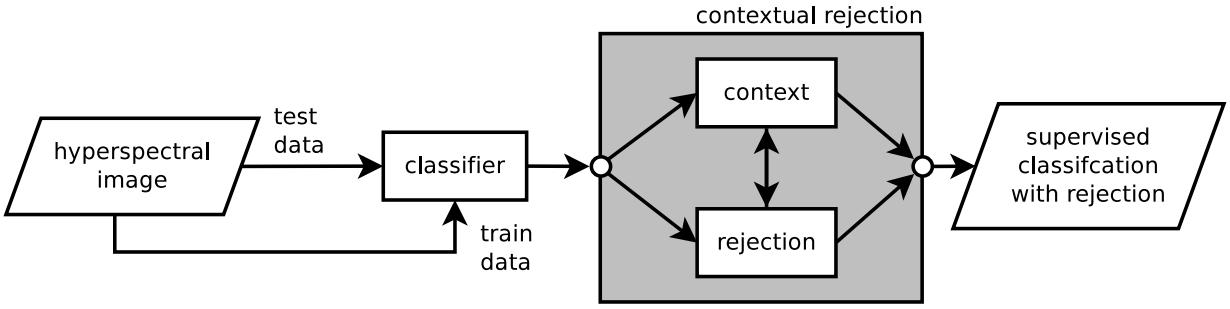


Figure 8.1: General diagram of supervised hyperspectral image classification with rejection. The classification block corresponds to a supervised classifier trained with labeled training pixels and applied to unlabeled test pixels. The contextual rejection block combines the classification with rejection with the classification context. In Section 8.2, two instantiations of contextual rejection are discussed.

8.3 presents experimental results and Section 8.4 concludes the chapter.

8.2 Rejection and context

As stated in chapters 2 and 4, classification with rejection can be achieved based on the existence of simple two mechanisms:

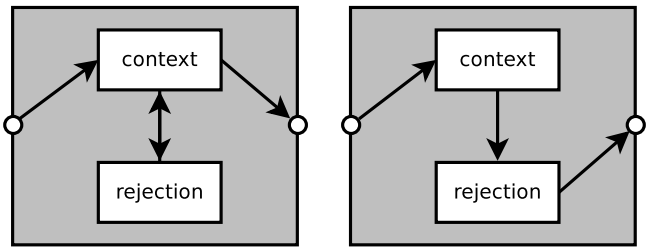
- An implicit ordering of the pixels according to their potential to be rejected;
- A concept of a threshold that controls the amount of pixels that are rejected.

This can be easily achieved by considering an extension of Chow's rule for two class classification with rejection, that is, the derivation of a probability threshold for a binary classification problem that minimizes the empirical risk given a cost matrix and the posterior probabilities [18]. Let us consider an image with K nonrejection classes, and a $K+1$ class that corresponds to rejection. The pixelwise MAP classification of the ith pixel is

$$\widehat{y_i} \in \underset{y_i \in \mathcal{L} \cup \{K+1\}}{\arg\max} \; p(y_i|\mathbf{x}_i) \tag{8.1}$$

where $p(y_i = K+1|\mathbf{x}_i) = \gamma$ represents the probability of rejection. The maximum probability of the K nonrejected classes of each pixel imposes an implicit ordering of the pixels (higher probability leading to lower potential to be rejected), and the amount of rejection is controlled by probability of rejection γ, the threshold.

The simple rejection scheme in (8.1) is limited by its pixel-based behavior. There is no awareness of context. In image classification, the use of context is of paramount importance as neighboring pixels are likely to belong to the same class. The same reasoning applies to the rejection. The potential for a pixel to be rejected should not be independent of whether the pixel is surrounded by other pixels that are rejected, or surrounded by pixels that are not rejected. As discussed in chapters 3 and 5, and in Section 8.1, the use of context in image classification, namely in hyperspectral image classification, is responsible for significant increase in performance.



(a) (b)

Figure 8.2: Architectures for computation of context and rejection. (a) JCR — joint computation of context and rejection, and (b) SCR — sequential computation of context and rejection

To solve the need for contextual awareness of rejection, we combine rejection and context. We consider two different ways to combine classification with rejection with classification with context. We can jointly compute context and rejection — JCR (as seen in Fig. 8.2(a)) by considering rejection to be an extra class, subject to the same contextual cues that the other classes are. This is explored in Subsection 8.2.1, where we instantiate JCR with the the SegSALSA-VTV algorithm applied to an extended set of probabilities, containing rejection as a $K+1$ class. On the other hand, we can harness the potential of the SegSALSA algorithm to provide a hidden field that provides us with an implicit ordering of the pixels according to their potential to be rejected — the maximum value of the hidden field for each pixel — that takes in account the contextual cues. This allows us to compute sequentially the rejection after the context — SCR (as seen in Fig. 8.2(b)). We follow this approach in Subsection 8.2.2, where we instantiate SCR with the rejection computed from the hidden field resulting from the SegSALSA-VTV algorithm with K classes through the computation of a rejection field.

8.2.1 JCR — joint computation of context and rejection

To compute jointly context and rejection, we consider rejection as an extra class. Rejection is conceptualized as an extra class that should be selected when there is evidence of probable misclassification by the classifier. In this formulation, the threshold γ in (8.1) is connected to the probability of misclassification by the classifier. This corresponds to the joint context and rejection architecture described in .

Let p_i^r denote the probability of the classifier misclassifying the ith pixel, we can easily extend the set of labels $\mathcal{L} = \{1, \ldots, K\}$ to include the extra class $K+1$ corresponding to rejection $\mathcal{L}' = \{1, \ldots, K, K+1\}$. With the new rejection class in place, we need to normalize the probabilities. The new class probabilities p' become

$$p'(y_i|\mathbf{x}_i) = \begin{cases} p_i^r, & \text{if } y_i = K+1, \\ (1-p_i^r)p(y_i|\mathbf{x}_i), & \text{otherwise.} \end{cases} \tag{8.2}$$

SegSALSA-JCR The joint computation of context and rejection leads to an extended SegSALSA-VTV formulation of (5.23), where the hidden field is now of dimension $\mathbf{z} \in \mathbb{R}^{(K+1)\times n}$ and the

probability vector $\boldsymbol{p}_i$ becomes $\boldsymbol{p}'_i$,

$$\widehat{\mathbf{z}}_{MMAP} \in \arg\min_{\mathbf{z}\in\mathbb{R}^{(K+1)\times n}} -\sum_{i\in\mathcal{S}} \left(\ln\left(\mathbf{p}'^{T}_{i}\mathbf{z}_i\right)\right) - \ln p(\mathbf{z})$$
$$\text{subject to:} \quad \mathbf{z} \geq 0, \quad \mathbf{1}_K^T\mathbf{z} = \mathbf{1}_n^T.$$

The rejection extra class is subject to the same vectorial total variation prior as the other classes. By considering rejection as an extra class, we are able to seamlessly combine classification with context with classification with rejection in the SegSALSA formulation.

The basic assumption for the JCR is that of rejection as an extra class with a probability associated to classifier failure. A scaling parameter γ controls the relative weight of the probability of classifier misclassification with regard to the probability of the other classes. By varying the value of γ we are able to vary the amount of rejection obtained, with larger values of γ corresponding to larger values of the rejected fraction. We now present two different rejection schemes based on two different models for classifier:

- Uniform probability of classifier failure — classifier failure is equiprobable across all the pixels;
- Entropy-weighted probability of classifier failure — classifier failure is more likely in pixels with higher entropy associated to their classification.

JCR-U — uniform probability of classifier failure

This uniformly weighted model assumes that, regardless of the probability distribution for each of the labels on a pixel, there is a constant probability of failure of the classifier, *i.e.* for all the pixels, the probability of misclassification, and thus rejection, is constant. The rejection depends only on the scaling parameter γ that defines how frequently misclassification is assumed,

$$p_i^r = \gamma.$$

Class probabilities of extended set of labels The class probabilities for the extended set of labels $\mathcal{L}'$ are

$$p'(y_i|\mathbf{x}_i) = \begin{cases} \gamma, & \text{if } y_i = K+1, \\ (1-\gamma)p(y_i|\mathbf{x}_i), & \text{otherwise.} \end{cases} \tag{8.3}$$

In this model, misclassifications are assumed to be equiprobable across the entire image.

JCR-E — entropy weighted probability of classifier failure

This entropy-weighted model assumes that the probability of failure of the classifier scales with the entropy associated with the probability vector from the classification, *i.e.* pixels with higher entropy are more likely to be misclassified, and thus rejected. The rejection depends both from the scaling parameter γ that defines how frequent the misclassification is assumed, and from the uncertainty associated with the classification modeled by the entropy weighting

$$p_i^r = \gamma H(\boldsymbol{p}_i),$$

where $H(\boldsymbol{p}_i)$ denotes the entropy of the probability distribution $\boldsymbol{p}_i = [p(y_i = 1|\mathbf{x}_i) \dots p(y_i = K|\mathbf{x}_i)]$.

Class probabilities of extended set of labels The class probabilities for the extended set of labels $\mathcal{L}'$ are

$$p'(y_i|\mathbf{x}_i) = \begin{cases} \gamma H(\boldsymbol{p}_i), & \text{if } y_i = K+1, \\ (1 - \gamma H(\boldsymbol{p}_i))p(y_i|\mathbf{x}_i), & \text{otherwise.} \end{cases} \tag{8.4}$$

In this model, misclassifications are assumed to be more probable in pixels with higher entropy.

Limitations of joint computation of context and rejection

A major limitation of considering rejection as an extra class modeling classifier failure is the inability to define *a priori* the amount of rejection obtained. Whereas γ in (8.3) and (8.4) corresponds to the scaling factor associated with the probability of classifier failure, the use of context through SegSALSA makes it impossible to predict the rejected fraction before the computation of SegSALSA. This means that, given an ordering of the pixels according to their potential to be rejected before the computation of context, there is no guarantee the ordering of the pixels will be the same after the computation of context.

8.2.2 SCR — sequential computation of context and rejection

To mitigate the aforementioned limitations associated with the joint computation of context and rejection, we consider a second approach where rejection is computed after the context, *i.e.* a sequential approach. This corresponds to the sequential context and rejection architecture described in . We start by noting that by using SegSALSA-VTV to compute the context, in addition to the labeling $\widehat{\mathbf{y}}$, we have the hidden field $\widehat{\mathbf{z}}_{\text{MMAP}}$ resulting from the optimization problem (5.23) from where the labeling with context only is computed.

SegSALSA-SCR This hidden field $\mathbf{z}$ provides an indication of the degree of confidence associated with the label of each pixel. If $[\mathbf{z}_i]_k > [\mathbf{z}_j]_l$, this is if the kth component of the hidden vector associated with the ith pixel $[\mathbf{z}_i]_k$ has a larger value than the lth component of the hidden vector associated with jth pixel $[\mathbf{z}_j]_l$, then we are led to believe that assigning the label l in the jth pixel corresponds to a lower degree of confidence than assigning the label k in the ith pixel.

Let us consider the labeling $\widehat{\mathbf{y}}$

$$\widehat{\mathbf{y}} \in \arg\max_{\mathbf{y} \in \mathcal{L}^n} p(\mathbf{y}|\widehat{\mathbf{z}}_{\text{MMAP}}),$$

and the associated maximum probabilities of the labeling $\mathbf{z}_{\widehat{\mathbf{y}}}$, such that

$$\mathbf{z}_{\widehat{y}_i} = p(\widehat{y}_i|\widehat{\mathbf{z}}_{\text{MMAP}}). \tag{8.5}$$

If $[\mathbf{z}_i]_{\widehat{y}_i} > [\mathbf{z}_j]_{\widehat{y}_j}$, there is strong evidence that a higher degree of confidence exists in the labeling of the ith pixel as $\widehat{y}_i$ than in the labeling of the jth pixel as $\widehat{\mathbf{y}}_j$. We denote the resulting field $\mathbf{z}_{\widehat{\mathbf{y}}}$ as *rejection field*.

By sorting $\mathbf{z}_{\widehat{\mathbf{y}}}$ we obtain an ordering of the pixels according to their relative confidence. Thus, from the hidden field $\mathbf{z}$ and the resulting rejection field $\mathbf{z}_{\widehat{y}}$, we obtain an implicit ordering of the pixels according to their potential to be rejected. The selection of a fraction of the lowest confidence pixels to be rejected yields a simple, yet effective scheme for rejection. This method allows one not only to define arbitrary values of the rejected fraction, but also to change the values on the fly, without the need to re-solve any contextual problem.

By promoting preservation and alignment of the discontinuities across the classes, the vectorial total variation prior (5.4), when applied to the hidden field $\mathbf{z}$, influences the behavior of the rejection field $\mathbf{z}_{\widehat{\mathbf{y}}}$. This results in an emergent prior behavior on the rejection field. The preservation and alignment of the discontinuities is thus imposed on the rejection field.

8.3 Experimental results

To evaluate the proposed methodologies of joint and sequential computation of context and rejection, we apply them to the task of supervised hyperspectral image classification of two well known hyperspectral scenes: AVIRIS Indian Pines and ROSIS Pavia University scene. In both scenes, the labeled ground truth is only available for a portion of the image. We apply the methodologies on the entire image and assess the performance on the subset of pixels that belongs to the labeled ground truth. We aim to show the following characteristics of supervised hyperspectral image classification with rejection:

- Classification with context and rejection can outperform classification with context only;
- Classification with rejection does not affect all the classes equally;
- By using classification with context and rejection with small training sets, we are able to achieve performances comparable to context only with larger training sets.

This is achieved by assessing the performance of the joint (SegSALSA-JCR-U and SegSALSA-JCR-E) and sequential (SegSALSA-SCR) schemes for context and rejection using SegSALSA to compute the context. The multinomial logistic regression (MLR) weights are modeled with LORSAL [96], thus obtaining the LORSAL-SegSALSA-JCR-U, LORSAL-SegSALSA-JCR-E, and LORSAL-SegSALSA-SCR methods for image classification with context and rejection. The SegSALSA algorithm requires the existence of class probabilities, which restrict us to the use of classifiers that output probabilities. The use of a MLR modeled with LORSAL can be easily replaced by the use of a probabilistic extension to support vector machines, such as relevance vector machines [109]. The LORSAL parameters used are $\lambda = 0.01$, $\theta = 0.001$ with radial basis function (RBF) kernels with a width of 1. For the SegSALSA algorithm, the value of λ_{TV} is 2. Computational complexities of both LORSAL-SegSALSA-SCR and LORSAL-SegSALSA-JCR approaches is dominated by the SegSALSA, which is $O(Kn \log n)$, with K the number of classes and n the number of image pixels. This means that computing LORSAL-SegSALSA-SCR has complexity of $O(Kn \log n)$ and computing LORSAL-SegSALSA-JCR has complexity of $O((K+1)n \log n)$. In LORSAL-SegSALSA-JCR-U and LORSAL-SegSALSA-JCR-E, a sweep on the scaling parameter of γ from 0 to 1 is performed to observe the joint variation of nonrejected accuracy, classification quality and fraction of rejected pixels.

8.3.1 Indian Pine

The AVIRIS Indian Pine scene (Fig. 8.3) was acquired by the AVIRIS sensor in NorthWest Indiana, USA. The scene consists of 145×145 pixel section with 200 spectral bands (with water absorption bands already purged) and contains 16 mutually nonexclussive classes, with the classification accuracy and classification quality being measured on those 16 classes.

The classification maps present in Fig. 8.3 show clearly the effects of classification with context and rejection: a significant number of misclassified pixels are rejected, thus increasing classification performance. We start from an accuracy of 51.39% with the MAP classification (with the training set composed of 10 pixels randomly selected per class, roughly 1.6% of the entire labeled data set) in Fig. 8.3 (b), and by computing the context alone with LORSAL-SegSALSA achieve an accuracy of 69.55% in Fig. 8.3 (c).

In Fig. 8.3 (d)-(f), we show the classification maps for the rejected fraction that corresponds to the maximum classification quality. This means that starting from the 69.55% accuracy of LORSAL-SegSALSA, the value of rejected fraction is selected such that the number of correct decisions (rejected the pixel when incorrectly classified, and not reject the pixel when correctly classified) is maximized. For LORSAL-SegSALSA-JCR-U, we achieve a nonrejected accuracy of 80.31% at a rejected fraction of 20.65% leading to a classification quality of 78.56%. This means that by not classifying the entire image, we depart from an accuracy of 69.55% on the entire image to an accuracy of 80.31% on 79.35% of the image, with 78.56% of the pixels either correctly classified and not rejected, or incorrectly classified and rejected. For LORSAL-SegSALSA-JCR-E, we achieve a nonrejected accuracy of 76.01% at a rejected fraction of 15.85% leading to a classification quality of 74.23%. For LORSAL-SegSALSA-SCR, we achieve 79.97% nonrejected accuracy at a rejected fraction of 23.75% and a classification quality of 76.16%.

The introduction of rejection does not affect all the classes equally. Some classes are more positively affected by rejection, whereas the classification performance of other classes suffers. The classwise classification performances are shown in Table 8.1, with classwise performance improvement highlighted in green and classwise performance decrease highlighted in red.

In Fig. 8.4, we illustrate the variation of the performance measures for classification with rejection as a function of the rejected fraction. It is clear the a steady increase of nonrejected accuracy by increasing the amount of the image rejected. On the other hand, by using the classification quality, we can compare the number of correct decisions made as we change the rejected fraction. From not rejecting any portion of the image, leading to a classification quality equal to the accuracy of the LORSAL-SegSALSA, we are able to increase the performance until it peaks, corresponding to a higher accuracy on the nonrejected pixels without rejecting too much of the image. We note the close position of peaks of the classification qualities for the LORSAL-SegSALSA-JCR-U and the LORSAL-SegSALSA-SCR approaches.

To compare the approaches of classification with rejection with the state of the art methods, we need to consider an increase of the training set dimension. In Table 8.2, we compare the performance of our methods with the results available in [110] for multiple classifiers with large training sets (10% of the pixels as training set): classifiers without context, classifiers with

[1]As all the pixels corresponding to the *oats* class are rejected, it is not possible to compute the nonrejected accuracy.

Table 8.1: Performance of classification with rejection for Indian Pine (10 pixels per class as training set). Overall and classwise nonrejected accuracy, rejected fraction and classification quality corresponding to maximum overall classification. Increase in performance (green) and decrease in performance (red). Best classwise classification performance in bold typeset.

		no rejection	LORSAL-SegSALSA-JCR-U			LORSAL-SegSALSA-JCR-E			LORSAL-SegSALSA-SCR		
class	number pixels	initial accuracy	nonrej. accuracy	rejected fraction	class. quality	nonrej. accuracy	rejected fraction	class. quality	nonrej. accuracy	rejected fraction	class. quality
alfalfa	46	95.65%	100.00%	6.52%	97.83%	100.00%	4.35%	**100.00%**	100.00%	4.35%	**100.00%**
corn-notill	1428	54.55%	63.66%	32.35%	**63.94%**	57.45%	29.06%	56.02%	63.15%	29.69%	**63.94%**
corn-mintill	830	25.66%	44.54%	55.90%	**69.52%**	38.62%	42.29%	61.20%	40.59%	51.33%	65.18%
corn	237	**99.16%**	100.00%	2.53%	98.31%	99.16%	0.00%	**99.16%**	100.00%	5.06%	95.78%
grass-pasture	483	82.82%	88.01%	8.49%	**86.75%**	84.60%	4.55%	83.23%	87.33%	8.49%	85.51%
grass-trees	730	**96.85%**	97.41%	4.66%	93.56%	97.34%	7.40%	90.82%	97.54%	5.48%	93.01%
grass-mowed	28	**100.00%**	100.00%	14.29%	85.71%	100.00%	0.00%	**100.00%**	100.00%	14.29%	85.71%
hay-windrowed	478	99.37%	100.00%	0.42%	**100.00%**	99.37%	0.00%	99.37%	99.37%	0.42%	98.95%
oats	20	**95.00%**	NaN[1]	100.00%	5.00%	95.00%	0.00%	**95.00%**	100.00%	60.00%	45.00%
soybean-notill	972	86.42%	98.44%	20.99%	**90.12%**	93.94%	16.87%	86.63%	94.88%	19.55%	85.80%
soybean-mintill	2455	52.75%	62.60%	24.20%	**66.35%**	58.15%	18.74%	60.49%	59.46%	24.44%	61.55%
soybean-clean	593	72.68%	85.62%	21.42%	**83.31%**	78.85%	14.67%	76.56%	83.70%	23.44%	78.92%
wheat	205	**99.51%**	100.00%	1.46%	99.02%	99.51%	0.00%	**99.51%**	100.00%	1.46%	99.02%
woods	1265	89.09%	91.88%	2.69%	**92.41%**	88.82%	0.32%	88.30%	90.79%	3.00%	90.04%
buildings	386	**63.21%**	63.14%	29.02%	55.44%	65.73%	16.84%	62.95%	61.59%	28.50%	53.37%
stone-steel	93	93.55%	100.00%	6.45%	**100.00%**	93.55%	0.00%	93.55%	100.00%	6.45%	**100.00%**
all	10249	69.55%	80.31%	20.65%	78.56%	76.01%	15.85%	74.23%	79.97%	23.75%	76.16%

context, and classifiers with context based on superpixelization (where an unsupervised segmentation produces an oversegmented partitioning of the image and forces pixels belonging to the same partition element to belong to the same class). We compare the performance of our methods with equivalent and smaller training sets (10% and 5% of pixels randomly selected as training set respectively).

For the classifiers without context, we SVM [111] and LORSAL [96]. For the classifiers with context, we consider SVM with composite kernels (SVM-CK) [112], LORSAL with multilevel logistic Markov random field priors (LORSAL-MLL) [96], sparse representation-based classification (SRC) [113], multinomial logistic regression with generalized composite kernel (MLR-GCK) [114], and LORSAL-SegSALSA [1]. For the classifiers with context based on superpixelization, we consider the superpixel-based classification via multiple kernels (SC-MK) [110] and its simplified version (INTRASC-MK) [110].

The use of classification with context and rejection is able to obtain significant performance improvements. We note that, by using classification with context and rejection with smaller training sets, both in sequential (SCR) and joint (JCR) approaches, we are able to achieve performances on the nonrejected data equivalent to those achieved by using classification with context only in larger training sets (highlighted in magenta in Table 8.2) not considering the superpixel-based methods. For example, with 5% of the pixels as training set, and while rejecting close to 15% of the pixels, we are able to achieve performances close to the ones achieved by context only with 10% of the pixels as training set, such as LORSAL-MLL, SRC, MLR-CGK, and SegSALSA.

On the other hand, we can achieve accuracies equivalent to the accuracies of superpixel-based methods (highlighted in cyan in Table 8.2), with equivalent training set size, by using rejection. By using context and rejection, we are able to close the gap between the state of the art methods

using superpixels (98.06% overall accuracy) and SegSALSA (92.26% overall accuracy). The rejection of 15% of the pixels in SCR allows us to attain values of nonrejected accuracy (97.64%) comparable to the state of the art.

As pointed in the introduction, the performance improvements resulting from the combination of rejection and context are more significant for weaker classifiers with lower performance. This is illustrated in Fig.8.5, where the strength of the classifier is a result of the training set size (from 0.5% to 20% of the labeled pixels used as training set). It is interesting to note the shift of the peak of classification quality to lower values of rejected fraction as the classification problem gets easier and the classifier gets more accurate. There is an increased dependency on the rejector as the classifier gets weaker.

8.3.2 Pavia university

The Pavia University scene (Fig. 8.6) was acquired with the ROSIS sensor in Pavia (Italy). The scene consists of a 610×340 pixel hyperspectral image with 103 spectral bands containing 9 not mutually exclusive classes, with the classification accuracy and classification quality being measured on those 9 classes.

The classification maps in Fig. 8.6, show an easier problem for the LORSAL and LORSAL-SegSALSA, with higher classification performances with context only (seen in Table 8.3) when compared to the Indian Pine scene. The rejector will have a harder task to improve the performance, leading to maximum classification qualities with smaller respective rejected fractions, *i.e.* a larger proportion of correct decisions is achieved by rejecting less.

We start from an accuracy of 70.13% with the MAP classification (with the training set composed of 10 pixels randomly selected per class, roughly 0.2% of the entire labeled data set) in Fig. 8.6 (b), and by computing the context alone with SegSALSA achieve an accuracy of 80.67% in Fig. 8.6 (c).

In Fig. 8.6 (d)-(f), we show the classification maps that correspond to the maximum classification quality. This means that starting from the 80.67% accuracy of LORSAL-SegSALSA, we reject such that the number of correct decisions is maximized. For LORSAL-SegSALSA-JCR-U, we achieve a nonrejected accuracy of 82.25% at a rejected fraction of 3.12% leading to a classification quality of 81.81%. For LORSAL-SegSALSA-JCR-E, we achieve a nonrejected accuracy of 86.45% at a rejected fraction of 12.75% leading to a classification quality of 82.93%. This means that by not classifying the entire image, we depart from an accuracy of 80.67% on the entire image to an accuracy of 86.45% on 86.25% of the image, with 82.93% of the pixels either correctly classified and not rejected, or incorrectly classified and rejected. For LORSAL-SegSALSA-SCR, we achieve 84.54% nonrejected accuracy at a rejected fraction of 9.16% and a classification quality of 82.08%.

The classwise classification performances are shown in Table 8.3. Taking the example of the LORSAL-SegSALSA-JCR-E results, only the classification performance of the *meadows* class is increased, with the performance of the other classes decreasing slightly. However, the abundance of the *meadows* class compensates the results, with a resulting increase in overall classifier performance. There is no decrease on nonrejected accuracies, in LORSAL-SegSALSA-JCR-E a large portion of correctly classified samples are being rejected across all the classes with exception of the *meadows* class.

Table 8.2: Comparison of classification performance for Indian Pine. Overall accuracy (with no rejection) for multiple classifiers with 10% of pixels as training set. Comparison with SCR and JCR for 5% and 10% of pixels as training set for different rejected fractions. Comparable nonrejected accuracies for classification with context only (magenta), and for classification with context only based on superpixels (cyan).

classifier	training set size	nonrej. acc. $A(r)$	rej. frac. r	class. qual $Q(r)$
SVM [111]	10%	79.53%	0.00%	79.53%
SVM-CK [112]	10%	91.51%	0.00%	91.51%
LORSAL-MLL [96]	10%	94.73%	0.00%	94.73%
SRC [113]	10%	94.66%	0.00%	94.66%
MLR-CGK [114]	10%	96.29%	0.00%	96.29%
INTRASC-MK [110]	10%	97.53%	0.00%	97.53%
SC-MK [110]	10%	98.06%	0.00%	98.06%
LORSAL [96]	5%	72.72%	0.00%	72.72%
LORSAL-SegSALSA [2]	5%	86.01%	0.00%	86.01%
LORSAL-SegSALSA-SCR	5%	88.97%	5.00%	88.03%
		91.36%	10.00%	88.44%
		93.63%	15.00%	88.15%
		95.16%	20.00%	86.24%
LORSAL-SegSALSA-JCR-U	5%	89.52%	5.03%	88.93%
		92.06%	10.05%	89.53%
		93.74%	15.00%	88.23%
		95.16%	19.92%	86.19%
LORSAL-SegSALSA-JCR-E	5%	89.43%	5.01%	88.78%
		91.35%	10.04%	88, 25%
		92.73%	14.99%	86.53%
		93.87%	20.02%	84.01%
LORSAL [96]	10%	78.57%	0.00%	78.57%
LORSAL-SegSALSA [2]	10%	92.26%	0.00%	92.26%
LORSAL-SegSALSA-SCR	10%	94.72%	5.00%	92.70%
		96.38%	10.00%	91.22%
		97.64%	15.00%	88.71%
		98.66%	20.00%	85.59%
LORSAL-SegSALSA-JCR-U	10%	95.55%	5.03%	94.02%
		96.61%	10.00%	91.39%
		97.52%	15.02%	88.27%
		98.63%	20.05%	85.27%
LORSAL-SegSALSA-JCR-E	10%	94.38%	5.02%	91.81%
		95.60%	10.00%	89.58%
		96.21%	15.00%	86.07%
		97.19%	20.04%	82.97%

In Fig. 8.7, we illustrate the variation of the performance measures for classification with rejection as a function of the rejected fraction. The peak in classification quality is achieved for values of rejected fraction smaller than the ones in the Indian Pine case. This is a result of an easier classification problem: the high performances achieved by the classifier leads to a low impact of the rejector. As most of the data is correctly classified, it is harder for the rejector to correctly reject pixels. This means that the rejected fraction that optimizes the classification quality, the number of correct decisions made, is much smaller than in the Indian Pine case.

Table 8.3: Performance of classification with rejection for Pavia University. Overall and classwise nonrejected accuracy, rejected fraction and classification quality corresponding to maximum overall classification quality. Increase in performance (green) and decrease in performance (red). Best classwise classification performance in bold typeset.

		no rejection	LORSAL-SegSALSA-JCR-U			LORSAL-SegSALSA-JCR-E			LORSAL-SegSALSA-SCR		
class	number pixels	initial accuracy	nonrej. accuracy	rejected fraction	class. quality	nonrej. accuracy	rejected fraction	class. quality	nonrej. accuracy	rejected fraction	class. quality
asphalt	6631	**96.80**%	97.94%	4.13%	95.11%	97.50%	9.03%	89.61%	98.19%	8.11%	91.77%
meadows	18649	65.88%	67.15%	1.86%	67.79%	74.63%	17.32%	**74.85**%	70.47%	10.11%	70.92%
gravel	2099	71.56%	74.57%	8.00%	**73.65**%	77.15%	20.77%	71.46%	76.35%	15.01%	73.23%
trees	3064	88.41%	90.66%	4.28%	**89.43**%	90.27%	5.42%	87.76%	91.76%	9.73%	86.98%
metal sheets	1345	**100.00**%	100.00%	0.00%	**100.00**%	100.00%	0.00%	**100.00**%	100.00%	0.00%	**100.00**%
bare soil	5029	92.19%	94.60%	2.03%	**95.21**%	96.08%	10.10%	90.67%	95.72%	7.62%	92.28%
bitumen	1330	95.11%	95.78%	0.30%	**96.17**%	95.11%	0.23%	94.89%	96.89%	3.16%	95.71%
bricks	3682	**92.40**%	96.24%	8.31%	**92.40**%	97.20%	13.82%	88.95%	97.71%	12.33%	91.25%
shadows	947	**99.89**%	99.89%	0.21%	99.68%	99.89%	0.11%	99.79%	99.89%	0.21%	99.68%
all	42776	80.67%	82.25%	3.12%	81.81%	86.45%	12.75%	82.93%	84.54%	9.16%	82.08%

8.3.3 Approximation effects

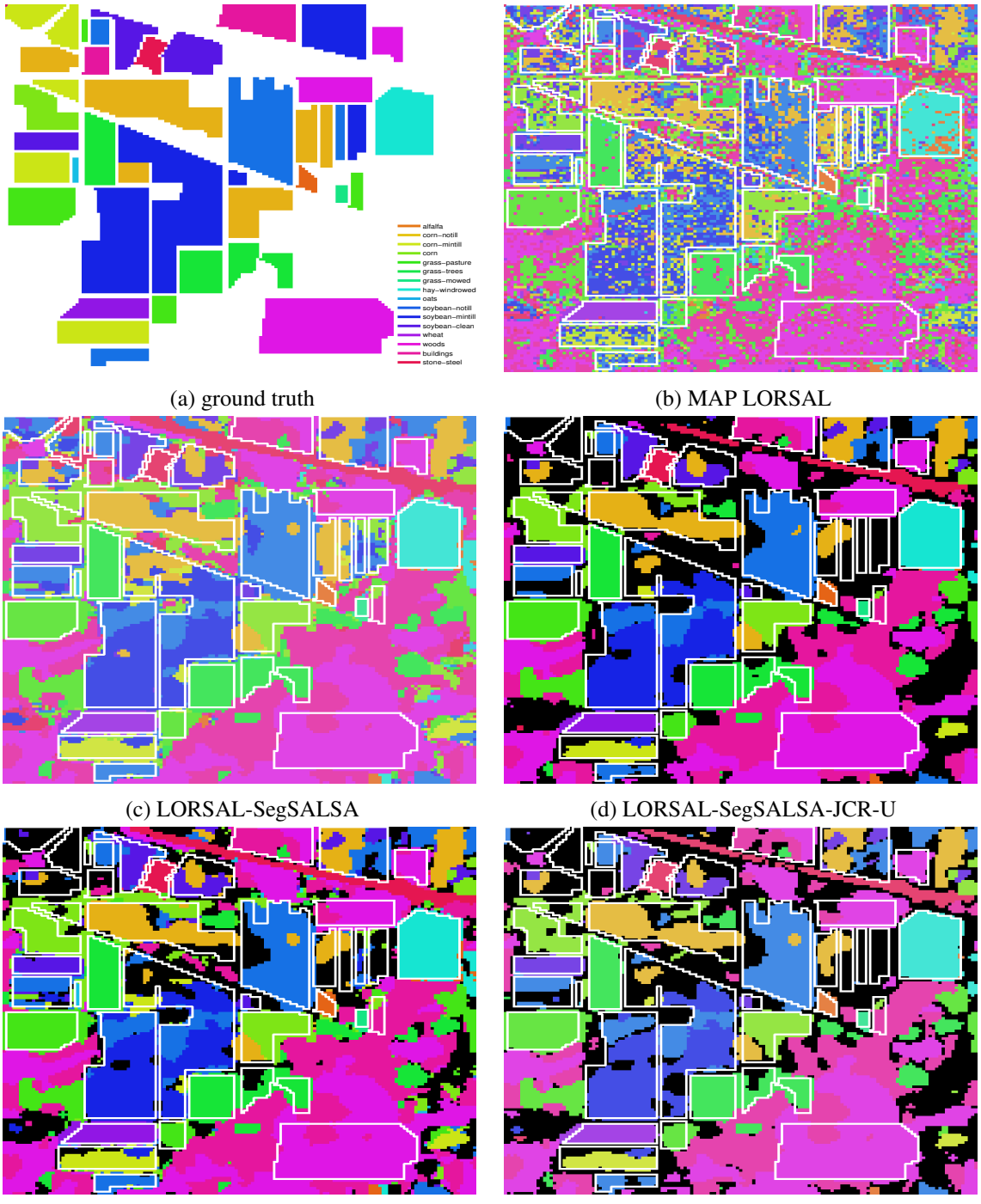
Whereas the JCR approaches, with LORSAL-SegSALSA-JCR-U in Indian Pine and LORSAL-SegSALSA-JCR-E in Pavia University, achieve higher performance than the SCR approach for smaller training sets (10 pixels per class), they are computationally more expensive. Firstly, there is not a clear direct connection between the value of γ and the rejected fraction, this connection is largely affected by the computation of the context. Whereas an increase of the value of γ can lead to larger rejected fractions, it is not possible to predict how much is rejected by the joint context and rejection. This is clear in Table 8.2, where we are able to precisely define *a priori* the rejected fraction for the LORSAL-SegSALSA-SCR approaches, but not able to do so for the LORSAL-SegSALSA-JCR approaches.

Secondly, obtaining the results for joint context and rejection requires a parameter sweep on the value of γ. This implies, *for each* value of γ, to compute the SegSALSA algorithm, or any other context computing algorithm, with $K + 1$ classes. For the SCR approach, the rejection is computed after the context, allowing us to obtain all possible values of the rejected fraction in a single computation of the SegSALSA algorithm,or any other context computing algorithm that provides a rejection field. However the sequential approach is subject to the approximation effect described in Fig. 6.6. This is clear when we observe in detail the accuracy rejection curves, both for Indian Pine and Pavia University, in Fig. 8.8. For the LORSAL-SegSALSA-JCR-U (in red), there is an increase of classification accuracy with no rejection happening. This corresponds to a

change in the labeling simply by the inclusion of the rejection class, as illustrated in Fig.6.6. The effect of the alteration of the labeling by introduction of the rejection class cannot be captured in any SCR approach, as the only change in the labeling allowed is for a pixel to be rejected.

8.4 Concluding remarks

In this chapter, we presented an algorithm for robust classification of hyperspectral images that uses the SegSALSA algorithm for context. We explored two different architectures for robust classification with rejection using context based on *joint* and *sequential* computations of context and rejection. We present experimental results of the methods for supervised hyperspectral image classification with rejection, with context computed using the SegSALSA algorithm. By using robust classifiers equipped with rejection, not only we are able to deal with imperfect knowledge in the training set and with smaller training sets, but also attain performance gains equivalent to increasing the training set size.



(a) ground truth

(b) MAP LORSAL

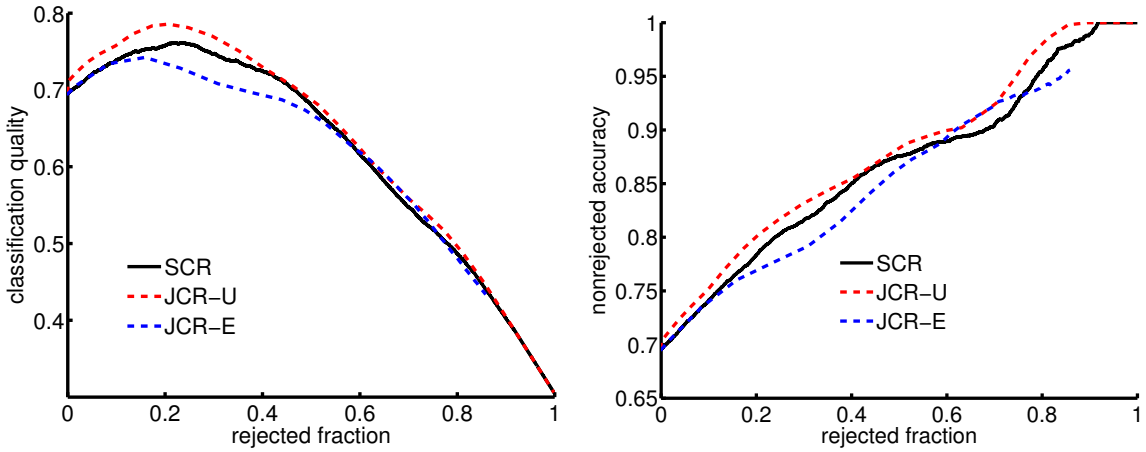
(c) LORSAL-SegSALSA

(d) LORSAL-SegSALSA-JCR-U

(e) LORSAL-SegSALSA-JCR-E

(f) LORSAL-SegSALSA-SCR

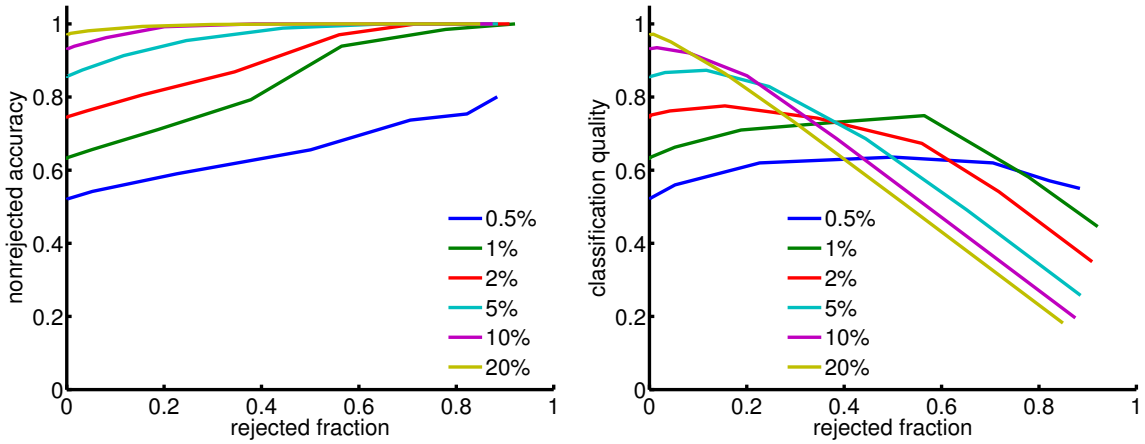
Figure 8.3: Classification results for Indian Pines (10 pixels per class as training set), with rejection in black. Ground truth (a), MAP classification using LORSAL (b), and classification with context - LORSAL-SegSALSA (c). Classification with context and rejection with maximum classification quality for LORSAL-SegSALSA-JCR-U (d), LORSAL-SegSALSA-JCR-E (e), and LORSAL-SegSALSA-SCR (f). Overall and class-wise nonrejected accuracy, rejected fraction and classification quality in Table 8.1.



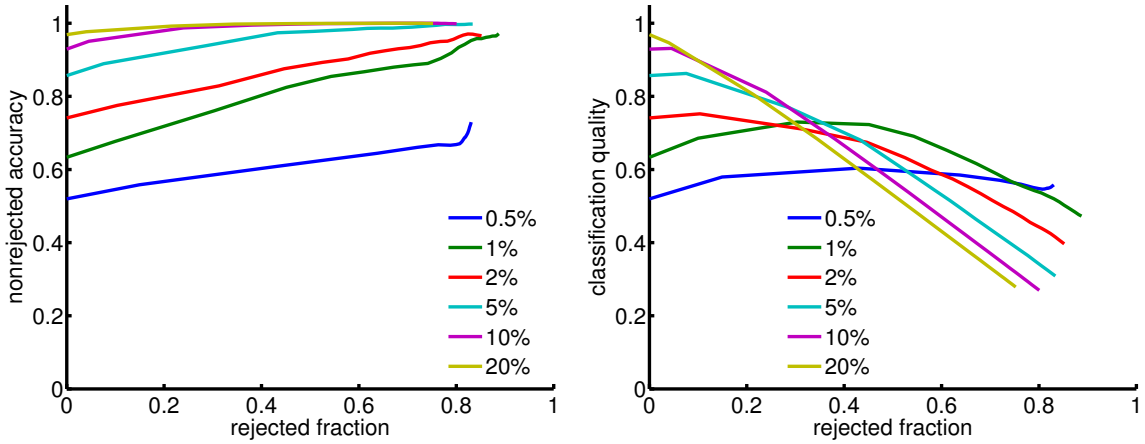
(a) classification quality

(b) nonrejected accuracy

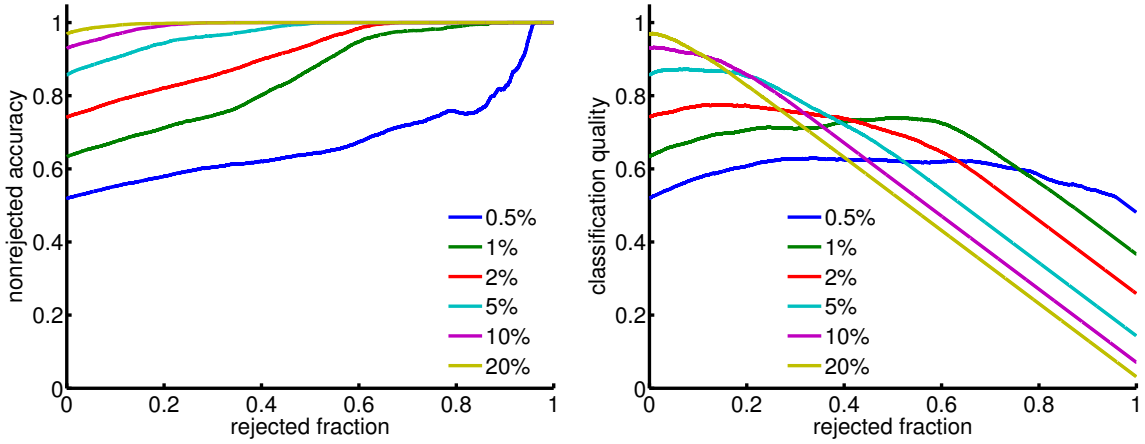
Figure 8.4: Performance for classification with rejection of the Indian Pine scene (10 pixels per class as training set). Classification with LORSAL-SegSALSA-SCR (black), and by LORSAL-SegSALSA-JCR-U (red) and LORSAL-SegSALSA-JCR-E (blue).



(a) SegSALSA-JCR-U

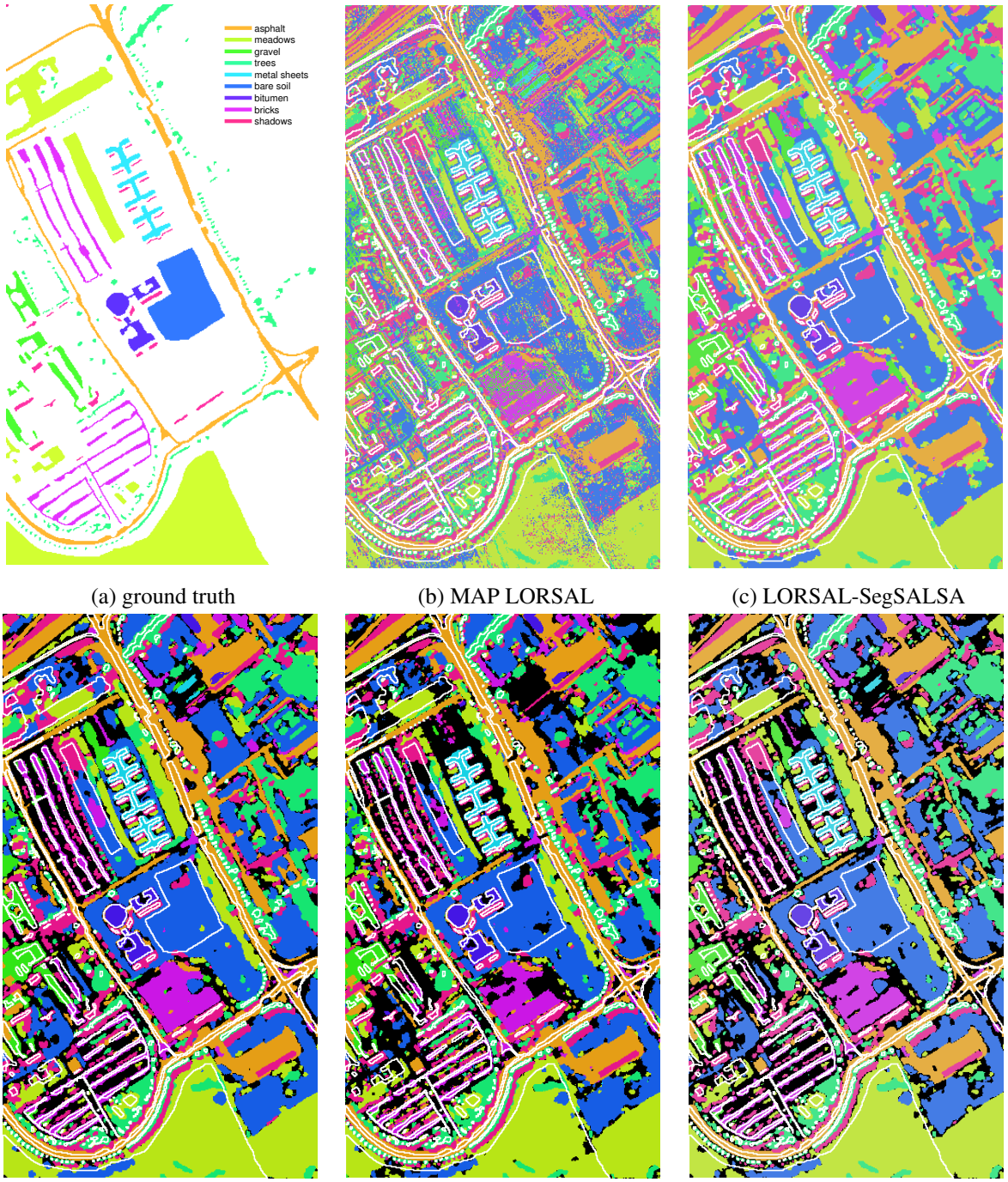


(b) SegSALSA-JCR-E



(c) SegSALSA-SCR

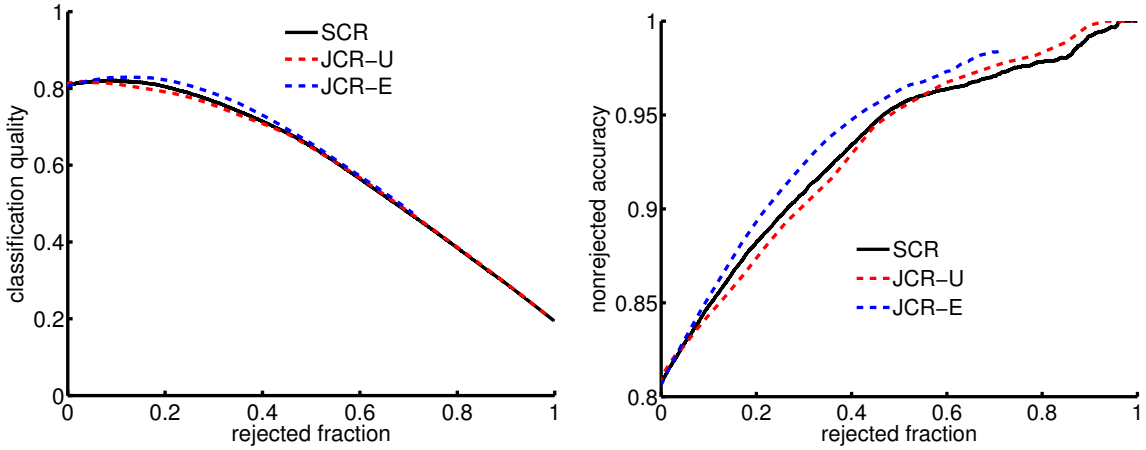
Figure 8.5: Effect of weak *vs.* strong classifiers in classification with rejection. Nonrejected accuracy (left) and classification quality (right). SegSALSA-JCR and SegSALSA-SCR approaches with increasing training size. Stronger classifiers (larger training sets) achieve peak classification quality with smaller values of rejected fraction than weak classifiers (smaller training sets).



(a) ground truth (b) MAP LORSAL (c) LORSAL-SegSALSA

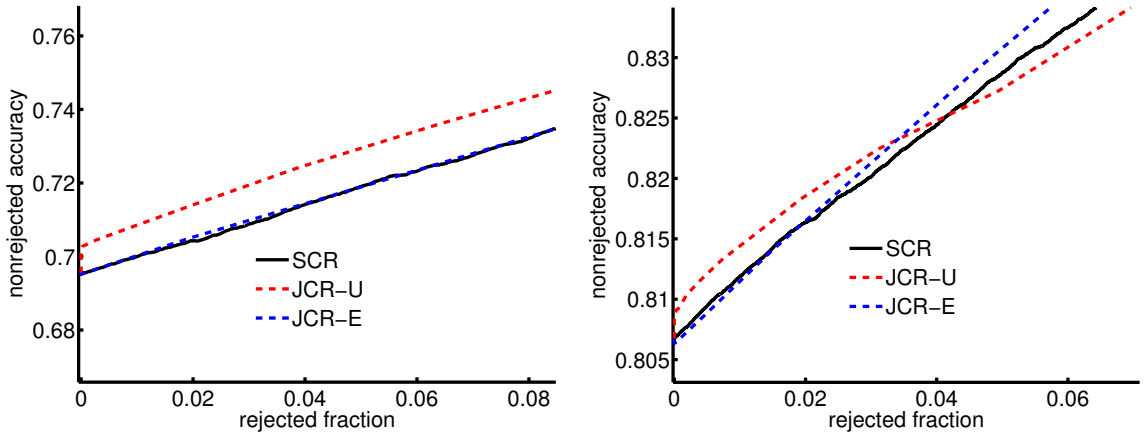
(d) LORSAL-SegSALSA-JCR-U (e) LORSAL-SegSALSA-JCR-E (f) LORSAL-SegSALSA-SCR

Figure 8.6: Classification results for Pavia University, rejection in black. Ground truth (a), MAP classification using LORSAL (b), and classification with context - LORSAL-SegSALSA (c). Classification with context and rejection with maximum classification quality for LORSAL-SegSALSA-JCR-U (d), LORSAL-SegSALSA-JCR-E (e), and LORSAL-SegSALSA-SCR. Overall and class-wise nonrejected accuracy, rejected fraction and classification quality in Table 8.3.



(a) classification quality (b) nonrejected accuracy

Figure 8.7: Performance for classification with rejection of the Pavia University scene. Classification with rejection by LORSAL-SegSALSA-SCR (black), and LORSAL-SegSALSA-JCR-U (red) and LORSAL-SegSALSA-JCR-E (blue) cost.



(a) Indian Pine (b) Pavia University

Figure 8.8: Approximation effects of SCR *vs.* JCR. Detail of nonrejected accuracy-rejection curve. Classification with LORSAL-SegSALSA-SCR (black), and with LORSAL-SegSALSA-JCR-U (red) and LORSAL-SegSALSA-JCR-E (blue). Increase of accuracy in the joint approaches due to the introduction of the rejected option.

Part IV

Concluding remarks and further work

Chapter 9

Concluding remarks

9.1 Concluding remarks

The main goal of this thesis was to create a framework for robust image classification using context and rejection. To this end, existing gaps in both classification with rejection and in classification with context had to be addressed before the creation of the framework for robust classification: the lack of measures associated with the evaluation of the performance of classification systems with rejection, and the design of systems for classification with context that sidestep from the inherent combinatorial nature of classification with context.

We will now summarize the contributions of this thesis in more detail.

Performance measures for classification with rejection, the performance measures presented, nonrejected accuracy, classification quality, and rejection quality, allow us to compare classification systems with rejection when rejecting a different amount of samples. The performance measures can be easily connected to families of loss functions that take in account rejection. This means that we are now able, with two different classifiers with rejection, to state the family of loss functions for which each classifier with rejection outperforms the other. Furthermore, the connection between the loss functions and the performance measures allows us to decide whether the classification should be performed with or without rejection.

A family of algorithms for classification with context — SegSALSA, which reformulates the problem of classification with context as a marginal maximum a posteriori estimation of a continuous hidden field that drives the discrete labeling. Thus, classification with context becomes a continuous convex optimization problem instead of an integer optimization problem. We explore the flexibility associated with the introduction of the hidden field to obtain context through the use of multiple priors: vectorial total variation regularization, structure tensor regularization based on the minimization of the Schatten norm of the patch-based Jacobian, and graph-based total variation. The use of graph-based total variation extends the use of context from local to nonlocal, as the similarity graph can be constructed from nonlocal similarity concepts.

A general framework for combining classification with context and classification with rejection into a framework for robust classification, by posing robust classification as an energy minimization problem defined on a graph. We can obtain different architectures for robust classification systems as we instantiate the interaction between the rejection and the context differently. We present two different schemes for robust classification systems with context and rejection through the definition of joint computation of context and rejection and sequential computation of context and rejection. The sequential computation of context and rejection results in an fast approximation to the energy minimization problem. We derive approximation bounds of the solutions and general considerations on the structure of the difference of solutions.

From the general framework, we apply robust classification with context and rejection to two different tasks: classification of histopathology data and classification of hyperspectral data.

An algorithm for robust image classification of histopathology images — ICRCI, where we obtain a robust classifier with a joint architecture for combining rejection and context by embedding the rejection into the context. Furthermore, based on expert knowledge, we model interclass transition according to the embryonic origin of the tissues. An analysis of the joint effect of context and rejection is performed through the analysis of the proportion of correct decisions (classification quality) and context-related parameters and rejection-related parameters vary.

An algorithm for robust image classification of hyperspectral images — SegSALSA-JCR and SegSALSA-SCR, where we obtain robust classifiers using a sequential and a joint context and rejection architecture. Based on the hidden fields obtained from approaching the problem of context using a SegSALSA-derived algorithm we implement the interaction between context and rejection differently for the two architectures. For the joint approach, rejection is considered as an extra class that models the probability of classifier failure, whereas for the sequential approach, rejection is derived from a rejection field extracted from the hidden field. By having the joint and sequential approaches modeled using the same methods, we are able to point to the approximation effects associated with the sequential approach solving the robust classification problem approximately.

9.2 Future work

We envision future work in the themes presented in this thesis to be focused on three main avenues.

Extensions of the performance measures A significant avenue for further work is the extension of the performance measures to account for the fact that, in certain classification problems, some classes are more important than others. One way this can be achieved is through the use of performance measures for classification with rejection as class-specific performance measures, as performed in Chapter 7. This means that class-specific misclassification costs should exist. On the other hand, class-specific rejection costs impose an extra degree of complexity on the

problem, as one should take in account a confusion matrix for rejection, the cost of rejecting a sample given that it belongs to one class and would have bee classified as a second class.

Parallelization of SegSALSA family of algorithms The use of algorithms for classification with context is inherently dependent on the speed of the algorithm. To this extent, the parallelization of the SegSALSA family of algorithms allows for significant speed improvements. As shown in Chapter 5, the parallelization capability of SegSALSA is dependent on the parallelization potential of the quadratic problem, and the parallelization potential of the split variables (dependent of the prior). On the other hand, the SegSALSA family of algorithms can be extended to account for the possibility of solving the problem of classification with context not only in parallel but also asynchronously.

Extension of robust classification to multiple rejection classes Finally, one interesting avenue to explore is the extension of the robust classification schemes to account for multiple rejection classes. This is akin to consider an outlier rejection class, accounting for new and unknown classes, and multiple inlier rejection classes, where there is no sufficient evidence to classify a sample as a member of an unknown class. This flexibility comes at the cost of a significant increase in the complexity of algorithms and also the design of priors that take in account transitions between multiple rejection classes.

Bibliography

[1] J. Bioucas-Dias, F. Condessa, and J. Kovačević, "Alternating direction optimization for image segmentation using hidden Markov measure field models," in *Proc. SPIE Conf. Image Process.*, San Francisco, Feb. 2014.

[2] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "Supervised hyperspectral image segmentation: a convex formulation using hidden fields," in *IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS'14)*, Lausanne, Switzerland, June 2014.

[3] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "SegSALSA-STR: a convex formulation to supervised hyperspectral image segmentation using hidden fields and structure tensor regularization," in *IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS'15)*, Tokyo, Japan, June 2015.

[4] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "SegSALSA-GTV: Supervised image segmentation using graph-based priors," *IEEE Trans. Image Process.*, 2016, to submit.

[5] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "Performance measures for classification systems with rejection," *Pattern Recognition*, 2016, submitted, preprint available at `http://arxiv.org/abs/1504.02763` [cs.CV].

[6] F. Condessa, J. Bioucas-Dias, C. Castro, J. Ozolek, and J. Kovačević, "Classification with rejection option using contextual information," in *Proc. IEEE Int. Symp. Biomed. Imag.*, San Francisco, Apr. 2013.

[7] F. Condessa, C. Castro, J. Ozolek, J. Bioucas-Dias, and J. Kovačević, "Image classification with rejection using contextual information," *IEEE Trans. Med. Imag.*, 2016, to submit, preprint available at `http://arxiv.org/abs/1509.01287` [cs.CV].

[8] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "Robust hyperspectral image classification with rejection fields," in *IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS'15)*, Tokyo, Japan, June 2015.

[9] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "Supervised hyperspectral image classification with rejection," in *IEEE Geoscience and Remote Sensing Symposium (IGARSS'15)*, Milan, Italy, July 2015.

[10] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "Supervised hyperspectral image classification with rejection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–14, 2016.

[11] J. Quevedo, A. Bahhamonde, M. Pérez-Enciso, and O. Luaces, "Disease liability prediction from large scale genotyping data using classifiers with a reject option," *IEEE/ACM Transactions on Computation Biology and Bioinformatics (TCBB)*, vol. 9, no. 1, pp. 88 – 97, 2012.

[12] G. Giacinto, F. Roli, and L. Bruzzone, "Combination of neural and statistical algorithms for supervised classification of remote-sensing images," *Pattern Recognition Letters*, vol. 21, no. 5, pp. 385–397, 2000.

[13] K. Karu and A. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389–404, 1996.

[14] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang, "Image classification for context-based indexing," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 117 – 130, 2001.

[15] R. Huber, H. Ramoser, K. Mayer, H. Penz, and M. Rubik, "Classification of coins using an eigenspace approach," *Pattern Recognition Letters*, vol. 26, no. 1, pp. 61–75, 2005.

[16] A. Payne and S. Singh, "Indoor vs. outdoor scene classification in digital photographs," *Pattern Recognition*, vol. 38, no. 10, pp. 1533–1545, 2005.

[17] C. Chow, "An optimum character recognition system using decision functions," *Electronic Computers, IRE Transactions on*, , no. 4, pp. 247–254, 1957.

[18] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Inf. Theory*, vol. 16, no. 1, pp. 41–46, Jan. 1970.

[19] G. Fumera, F. Roli, and G. Giacinto, "Reject option with multiple thresholds," *Pattern Recogn.*, vol. 33, no. 12, pp. 2099–2101, Dec. 2000.

[20] F. Tortorella, "An optimal reject rule for binary classifiers," in *Advances in Pattern Recognition*, pp. 611–620. Springer, 2000.

[21] D. Tax and R. Duin, "Growing a multi-class classifier with a reject option," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1565–1570, 2008.

[22] F. Tortorella, "A roc-based reject rule for dichotomizers," *Pattern Recognition Letters*, vol. 26, no. 2, pp. 167–180, 2005.

[23] C. Marrocco, M. Molinara, and F. Tortorella, "An empirical comparison of ideal and empirical roc-based reject rules," in *Machine Learning and Data Mining in Pattern Recognition*, pp. 47–60. Springer, 2007.

[24] R. Herbei and M. Wegkamp, "Classification with reject option," *Canadian Journal of Statistics*, vol. 34, no. 4, pp. 709–721, 2006.

[25] P. Foggia, C. Sansone, F. Tortorella, and M. Vento, "Multiclassification; reject criteria for the Bayesian combiner," *Pattern Recogn.*, vol. 32, no. 8, pp. 1435–1447, Aug. 1999.

[26] I. Pillai, G. Fumera, and F. Roli, "Multi-label classification with a reject option," *Pattern Recogn.*, vol. 46, no. 8, pp. 2256 – 2266, 2013.

[27] D. Tax, *One-class Classification*, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 2001.

[28] T. Landgrebe, D. Tax, P. Paclik, R. Duin, and C. Andrew, "A combining strategy for ill-

defined problems," in *Fifteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2004, pp. 57–62.

[29] P. Foggia, G. Percannella, C. Sansone, and M. Vento, "On rejecting unreliably classified patterns," in *Multiple Classifier Systems*, M .Haindl, J. Kittler, and F. Roli, Eds., vol. 4472 of *Lecture Notes in Computer Science*, pp. 282–291. Springer, 2007.

[30] M. Wegkamp, "Lasso type classifiers with a reject option," *Electronic Journal of Statistics*, pp. 155–168, 2007.

[31] F. Tortorella, "Reducing the classification cost of support vector classifiers through an roc-based reject rule," *Pattern Analysis and Applications*, vol. 7, no. 2, pp. 128–143, 2004.

[32] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu, "Support vector machines with a reject option," *in Advances in Neural Information Processing Systems*, pp. 537 – 544, 2009.

[33] G. Fumera and F. Roli, "Support vector machines with embedded reject option," in *Proc. Int. Workshop on Pattern Recognition with Support Vector Machines (SVM2002), Niagara Falls*, Niagara Falls, Canada, Aug. 2002, pp. 68–82, Springer-Verlag.

[34] M. Wegkamp and M. Yuan, "Support vector machines with a reject option," *Bernoulli*, vol. 17, no. 4, pp. 1368 – 1385, 2011.

[35] P. Bartlett and M. Wegkamp, "Classification methods with reject option using a hinge loss," *Journal Machine Learning Research*, vol. 9, pp. 1823–1840, Aug. 2008.

[36] M. Yuan and M. Wegkamp, "Classification methods with reject option based on convex risk minimization," *Journal Machine Learning Research*, vol. 11, pp. 111–130, Mar. 2010.

[37] G. Fumera and F. Roli, "Analysis of error-reject trade-off in linearly combined multiple classifiers," *Pattern Recogn.*, vol. 37, no. 6, pp. 1245 – 1265, 2004.

[38] R. Sousa and J. Cardoso, "The data replication method for the classification with reject option," *AI Communications*, vol. 26, no. 3, pp. 281 – 302, 2013.

[39] G. Fumera, I. Pillai, and F. Roli, "Classification with reject option in text categorisation systems," in *Proc. 12th International Conference on Image Analysis and Processing*, Washington, DC, USA, 2003, ICIAP '03, pp. 582–587, IEEE Computer Society.

[40] T. Landgrebe, D. Tax, P. Paclík, and R. Duin, "The interaction between classification and reject performance for distance-based reject-option classifiers," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 908 – 917, 2006.

[41] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[42] S. Kumar and M. Hebert, "Discriminative random fields," *Int. J. Comput. Vis.*, vol. 68, no. 2, pp. 179–201, 2006.

[43] J. Besag, "On the statistical analysis of dirty images," *Journal of Royal Statistics Society*, vol. 48, pp. 259–302, 1986.

[44] S. Geman and C. Graffigne, "Markov random field image models and their applications to

computer vision," in *Proceedings of the International Congress of Mathematicians*, 1986, vol. 1, p. 2.

[45] W. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *International journal of computer vision*, vol. 40, no. 1, pp. 25–47, 2000.

[46] S. Barnard, "Stochastic stereo matching over scale," *International Journal of Computer Vision*, vol. 3, no. 1, pp. 17–32, 1989.

[47] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[48] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 1, pp. I–74.

[49] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. IEEE, 2001, vol. 1, pp. 105–112.

[50] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*. ACM, 2004, vol. 23, pp. 309–314.

[51] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient ND image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[52] J. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *Geoscience and Remote Sensing Magazine, IEEE*, vol. 1, no. 2, pp. 6–36, 2013.

[53] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes," *Advances in neural information processing systems*, vol. 14, pp. 841, 2002.

[54] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 6, pp. 721–741, 1984.

[55] S. Li, *Markov random field modeling in computer vision*, Springer-Verlag New York, Inc., 1995.

[56] R. Potts, "Some generalized order-disorder transformations," in *Mathematical proceedings of the Cambridge philosophical society*. Cambridge Univ Press, 1952, vol. 48, pp. 106–109.

[57] J. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," 1971.

[58] S. Kumar and M. Hebert, "Discriminative random fields: A discriminative framework for contextual interaction in classification," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1150–1157.

[59] J. Pearl, *Probabilistic inference in intelligent systems*, Morgan Kaufmann San Mateo, CA,

1988.

[60] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *International journal of computer vision*, vol. 70, no. 1, pp. 41–54, 2006.

[61] Tommi Wainwright, M. and A. Willsky, "Tree consistency and bounds on the performance of the max-product algorithm and its generalizations," *Statistics and computing*, vol. 14, no. 2, pp. 143–166, 2004.

[62] M. Wainwright, T. Jaakkola, and A. Willsky, "Map estimation via agreement on trees: message-passing and linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 11, pp. 3697–3717, 2005.

[63] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1568–1583, 2006.

[64] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields," in *Computer Vision–ECCV 2006*, pp. 16–29. Springer, 2006.

[65] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 1068–1080, 2008.

[66] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 9, pp. 1124–1137, 2004.

[67] M. Tappen and W. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 900–906.

[68] V. Kolmogorov and C. Rother, "Comparison of energy minimization algorithms for highly connected graphs," in *Computer Vision–ECCV 2006*, pp. 1–15. Springer, 2006.

[69] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 147–159, 2004.

[70] C. Zach, D. Gallup, J. Frahm, and M. Niethammer, "Fast global labeling for real-time stereo using multiple plane sweeps.," in *VMV*, 2008, pp. 243–252.

[71] T. Pock, A. Chambolle, D. Cremers, and H. Bischof, "A convex relaxation approach for computing minimal partitions," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 810–817.

[72] C. Nieuwenhuis, E. Töppe, and D. Cremers, "A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model," *International journal of computer vision*, vol. 104, no. 3, pp. 223–240, 2013.

[73] A. Chambolle, D. Cremers, and T. Pock, "A convex approach for computing minimal partitions," 2008.

[74] J. Lellmann, F. Lenzen, and C. Schnörr, "Optimality bounds for a variational relaxation of the image partitioning problem," *Journal of mathematical imaging and vision*, vol. 47, no. 3, pp. 239–257, 2013.

[75] E. Arce Marroquin, J. and S. Botello, "Hidden Markov measure field models for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 11, pp. 1380–1387, 2003.

[76] M. Figueiredo, "Bayesian image segmentation using wavelet-based priors," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 437–443.

[77] X. Bresson and T. Chan, "Fast dual minimization of the vectorial total variation norm and applications to color image processing," *Inverse Problems and Imaging*, vol. 2, no. 4, pp. 455–484, 2008.

[78] B. Goldluecke, E. Strekalovskiy, and D. Cremers, "The natural vectorial total variation which arises from geometric measure theory," *SIAM Journal on Imaging Sciences*, vol. 5, no. 2, pp. 537–563, 2012.

[79] S. Lefkimmiatis, A. Roussos, M. Unser, and P. Maragos, *Convex Generalizations of Total Variation Based on the Structure Tensor with Applications to Inverse Problems*, Springer, 2013.

[80] S. Lefkimmiatis, J. Ward, and M. Unser, "Hessian schatten-norm regularization for linear inverse problems," *arXiv preprint arXiv:1209.3318*, 2012.

[81] Pu. Kohli and P. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[82] S. Chen, A. Sandryhaila, G. Lederman, Z. Wang, J. Moura, P. Rizzo, J. Bielak, J. Garrett, and J. Kovačević, "Signal inpainting on graphs via total variation minimization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 8267–8271.

[83] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[84] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, 2012.

[85] J. Yagnik, D. Strelow, D. Ross, and R. Lin, "The power of comparative reasoning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2431–2438.

[86] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, 2011.

[87] D. Bertsekas J. Eckstein, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.

[88] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[89] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires," *Revue Française dAutomatique, Informatique et Recherche Operationnelle*, vol. 9, pp. 41–76, 1975.

[90] J. Kovačević, V. Goyal, and M. Vetterli, *Fourier and Wavelet Signal Processing*, Cambridge University Press, 2015.

[91] P. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212. Springer, 2011.

[92] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[93] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, second edition, 1989.

[94] J. Santner, *Interactive multi-label segmentation*, Ph.D. thesis, University of Graz, Graz, Austria, 2010.

[95] C. Nieuwenhuis and D. Cremers, "Spatially varying color distributions for interactive multilabel segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 5, pp. 1234–1247, 2013.

[96] J. Li, J. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new bayesian approach with active learning," *Geoscience and Remote Sensing Magazine, IEEE*, vol. 49, no. 10, pp. 3947–3960, Oct. 2011.

[97] R. Bhagavatula, M. Fickus, J. Kelly, C. Guo, J. Ozolek, C. Castro, and J. Kovačević, "Automatic identification and delineation of germ layer components in H&E stained images of teratomas derived from human and nonhuman primate embryonic stem cells," in *Proc. IEEE Int. Symp. Biomed. Imag.*, Rotterdam, Apr. 2010, pp. 1041–1044.

[98] M. McCann, R. Bhagavatula, M. Fickus, J. Ozolek, and J. Kovačević, "Automated colitis detection from endoscopic biopsies as a tissue screening tool in diagnostic pathology," in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, Sept. 2012, pp. 2809–2812.

[99] R. Bhagavatula, M. McCann, M. Fickus, C. Castro, J. Ozolek, and J. Kovačević, "A vocabulary for the identification of teratoma tissue in H&E-stained samples," *J. Pathol. Inform.*, 2014.

[100] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1814–1821.

[101] A. Kulkarni, F. Condessa, and J. Kovačević, "Unsupervised image segmentation using comparative reasoning and random walks," in *Proc. IEEE Glob. Conf. Signal Information*

Process., Orlando, FL, Dec. 2015, pp. 338–342.

[102] D. Böhning, "Multinomial logistic regression algorithm," *Ann. Inst. Stat. Math.*, vol. 44, no. 1, pp. 197–200, Mar. 1992.

[103] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957 – 967, June 2005.

[104] D. Bertsekas J. Eckstein, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.

[105] J. Li, J. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subpsace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[106] X. Bai, H. Zhang, and J. Zhuo, "VHR object detection based on structural feature extraction and query expansion," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6508–6520, Oct. 2014.

[107] X. Bai, Z. Guo, Y. Wang, Z. Zhang, and J. Zhuo, "Semi-supervised hyperspectral band selection via spectral-spatial hypergraph model," *IEEE J. Sel. Topics App. Earth Obs. and Remote Sens.*, vol. 8, no. 6, pp. 2774–2783, June 2015.

[108] A. Brown, "Spectral curve fitting for automatic hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1601–1608, June 2006.

[109] B. Demir and S. Ertürk, "Hyperspectral image classification using relevance vector machines," *IEEE Geosci. Remote Sens. Let.*, vol. 4, no. 4, pp. 586–590, Oct. 2007.

[110] L. Fang, S. Li, W. Duan, J. Ren, and J. Benediktsson, "Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.

[111] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[112] G. Camps-Valls, L. Gomez-Chova, J. Mu noz Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[113] Y. Chen, N. Nasrabadi, and T. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[114] J. Li, P. Marpu, A. Plaza, J. Bioucas-Dias, and J. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sept. 2013.