The Information Regularization Framework for Semi-supervised Learning

by

Adrian Corduneanu

B.S., University of Toronto (1999)
S.M., Massachusetts Institute of Technology (2002)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author .....
Department of Electrical Engineering and Computer Science
April 7, 2006

Certified by .....
Tommi Jaakkola
Associate Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

The Information Regularization Framework for Semi-supervised Learning

by

Adrian Corduneanu

Submitted to the Department of Electrical Engineering and Computer Science
on April 7, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In recent years, the study of classification shifted to algorithms for training the classifier from data that may be missing the class label. While traditional supervised classifiers already have the ability to cope with some incomplete data, the new type of classifiers do not view unlabeled data as an anomaly, and can learn from data sets in which the large majority of training points are unlabeled. Classification with labeled and unlabeled data, or semi-supervised classification, has important practical significance, as training sets with a mix of labeled an unlabeled data are commonplace. In many domains, such as categorization of web pages, it is easier to collect unlabeled data, than to annotate the training points with labels.

This thesis is a study of the information regularization method for semi-supervised classification, a unified framework that encompasses many of the common approaches to semi-supervised learning, including parametric models of incomplete data, harmonic graph regularization, redundancy of sufficient features (co-training), and combinations of these principles in a single algorithm. We discuss the framework in both parametric and non-parametric settings, as a transductive or inductive classifier, considered as a stand-alone classifier, or applied as post-processing to standard supervised classifiers. We study theoretical properties of the framework, and illustrate it on categorization of web pages, and named-entity recognition.

Thesis Supervisor: Tommi Jaakkola
Title: Associate Professor

Acknowledgments

I would not have been able to complete this journey without the aid and support of countless people. I must first express my gratitude towards my advisor, Professor Tommi Jaakkola, who was the ideal advisor that many graduate students dream of: always available for lengthy debates and clever insights, and accommodating uncommon requests for leaves of absence. I must also thank the other two members of my thesis committee, Professor Tomaso Poggio, and Professor Allan Willsky for their help in finalizing this thesis.

Without the support of Professor Brendan Frey during a one-year exchange program at University of Toronto, this thesis would not have been possible. I would like to extend my acknowledgments to him.

I would like to thank my mentors from my internships at Microsoft Research, Professor Christopher Bishop, Doctor Chris Meek, Doctor Hagai Attias, Professor Eric Brill, Doctor John C. Platt, who contributed to my scholarship in the field of machine learning.

I would like to thank the many graduate students I collaborated with during my PhD. And last but not least, the person who made me want to complete this journey, Simona, to whom I dedicate this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Classification, the topic studied in this thesis, is one of the standard sub-fields of machine learning. In classification the goal is to produce algorithms that can predict the category of an object from a few measurable features of that object. In traditional *supervised learning*, one builds such *classifiers* from a training set of sample objects along with their associated category. Suppose for example that the goal is to determine whether a news article fits the "business" category. One can build a classifier from a number of sample articles that we manually determined whether they belong to the "business" category. Supervised learning algorithms generally require that all training examples are labeled in order to contribute to the classifier[1].

In contrast, in the traditional *unsupervised learning* sub-field of machine learning algorithms do not require any knowledge about the category of the training objects whatsoever. The goal is not classification, but only *clustering*. Given a set of news articles of unknown category, an unsupervised learning algorithm would group the articles in a number of unidentified categories, based on, for example, the similarity of their word distributions.

Semi-supervised learning is a more recent development in machine learning that blurs the line between supervised and unsupervised learning. The goal is to construct a classifier from a training set that consists of a mix of labeled and unlabeled objects. The simplest semi-supervised classifier would be a clustering step followed by a supervised classifier

[1]Some supervised algorithms are robust to a few examples with missing labels; nevertheless, they are not designed to learn from those samples.

that predicts a label for each resulting cluster. If we can separate news articles based on their word distribution alone into "business" and "non-business", without knowing which cluster is "business", then we need only a few articles of known category to label the clusters correctly.

It is typical of semi-supervised algorithms to require only a few labeled objects if they have access to a large number of unlabeled ones. Thus semi-supervised learning is well suited to situations in which it is inexpensive to gather unlabeled data, but labeling it is more involved. This is the case when we can gather unlabeled data automatically, but labeling it requires human labor. We may be able to gather automatically hundreds of news articles every day from online feeds, but most of this data comes untagged, and a person would need to manually read the articles to determine their category.

The topic of this thesis is a framework for semi-supervised learning that encompasses many of the current approaches, the *information regularization* framework. At the center of the framework lies the notion of *semi-supervised bias*. A semi-supervised bias is a subset of the training data that we believe a priori that it consists of objects of similar category. Referring to our news article example, we could formulate a semi-supervised bias for every word in the vocabulary, of all articles that have that word in common. We could also formulate semi-supervised biases based on whether the articles come from a common source, or whether they were written close in time. The semi-supervised biases need not be 100% correct, and weak signals of label similarity are fair. The goal of information regularization is then to assign labels to all unlabeled documents in a way that is most consistent with the observed labeled objects, and with the semi-supervised biases.

The range of semi-supervised biases that can be defined is quite broad, so that the framework is flexible, and subsumes many known semi-supervised algorithms, including parametric models of incomplete data, harmonic graph regularization, redundancy of sufficient features (co-training). Because we can envision an information regularization algorithm based on semi-supervised biases of different kinds, with information regularization it is also possible to combine known semi-supervised algorithms.

Szummer and Jaakkola [53] introduced the original information regularization principle in a study of the influence of the distribution of the objects on the variation in their label,

when the objects are represented by a one-dimensional real number. This thesis reformulates the original idea and makes it into a generic framework applicable to a much wider set of tasks than the original. It introduces a simple and efficient message passing algorithm, that turns information regularization into a practical semi-supervised method. The thesis also touches many theoretical aspects of information regularization.

1.1 Outline

In Chapter 2 we introduce semi-supervised classification, along with a literature review of the notable approaches, and place information regularization in context.

In Chapter 3 we introduce the framework of information regularization in its generic form. We discuss in detail the classification objective, but defer the presentation of specific algorithms for optimizing it to instantiations of the framework in subsequent chapters. We show connections to semi-supervised learning with the EM algorithm, co-training, graph regularization, low-density separation, and discuss the information theoretical interpretation of the framework. We also categorize the various forms in which we can instantiate the information regularization framework.

Chapter 4 is about semi-supervised learning on tasks with continuous features, on which the data density is correlated with the variation of the label. We refine the objective to account for the infinite number of regions that one can define on continuous spaces, obtaining it as a limiting form of the generic information regularization objective.

In Chapter 5 we apply information regularization in a transductive setting, in which we are only interested in computing the labels of a finite number of data points that we know in advance. Information regularization on graphs results in an efficient optimization algorithm that is the main application of the framework. We discuss the various properties of the algorithm.

Chapter 6 illustrates the algorithms developed in previous chapters on both synthetic and real tasks. We demonstrate the performance of the framework on categorization of web pages, and on named-entity recognition.

Appendix A contains a glossary of symbols that appear in the thesis with a consistent

meaning. Please refer to this list often to clarify any ambiguity in regards to notation.

Chapter 2

Background

A central problem in machine learning is *classification*: building algorithms that learn from examples to categorize data with high accuracy. In its abstract formulation, a classifier assigns class labels $y \in \mathcal{Y}$ to data points $\mathbf{x} \in \mathcal{X}$, where we represent each point by a vector of features, a set of measurable quantities that summarize our knowledge about the data point. For example, $\mathbf{x}$ may be a vector of pixel intensities in a 128×128 image, and y the name of the person whose face is depicted by the image; or $\mathbf{x}$ may be the set of words that belong to a document, and y its topic. Assuming that data and associated labels are distributed according to $P_{XY}(\mathbf{x}, y)$, and that $\hat{y}(\mathbf{x})$ denotes the output of the classifier, a measure of its performance is the expected error

$$\sum_{y \in \mathcal{Y}} \int P_{XY}(\mathbf{x}, y) \delta\left(y, \hat{y}(\mathbf{x})\right) d\mathbf{x} \tag{2.1}$$

where $\delta(y_1, y_2) = 0$ if $y_1 = y_2$, and 1 otherwise.[1] Research on classification aims at constructing classifiers of small expected error.

Had the data distribution been known, classification would be trivial, and we could construct an optimal classifier by setting $\hat{y}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} P_{Y|X}(y|\mathbf{x})$.[2] In practice $P_{XY}(\mathbf{x}, y)$ is never known, and classification algorithms minimize approximations to the expected error based on available evidence, often with theoretical guarantees of the error

[1] For utmost generality, we would substitute δ by a loss function $L(y, \hat{y}(\mathbf{x}))$. Such generality is not needed here though.

[2] In the literature this is called the Optimal Bayes Classifier [7].

in the approximation.

In conventional machine learning (a.k.a. *supervised learning*), the evidence on the basis of which the expected error is approximated and minimized consists of a set of training examples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l\}$ sampled from $P_{XY}(\mathbf{x}, y)$ independently. Following up on our previous examples, the training set could be pictures of known persons; or documents of known topic. These labeled examples can be used to teach the classifier to predict class labels for unobserved data.

While it is clear that examples of known class label are informative for training classifiers, recent developments suggest that even unlabeled data is valuable. Ideas of using unlabeled data for classification have been around since as early as 70's [18], but only in the past 5–8 years the field has seen an explosion of articles published on the topic. *Semi-supervised learning*[3] is attractive because training data is usually gathered unlabeled, and there is some cost to labeling it by human experts. Researchers often quote protein shape classification as an extreme example, as the aminoacid sequence of a protein is readily available (the feature part), but determining the 3D structure (the label part) takes months of expensive experimental effort. In the more typical case the cost of labeling samples is not that disproportionate, but unlabeled data is still more available. In what follows, the training data consists of set of points $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, and the labels of the first l points: $y_i, 1 \leq i \leq l$. The other $u = n - l$ points remain unlabeled.

A question asked by many who are exposed to semi-supervised learning for the first time is why unlabeled data carries information useful for classification. How can observing a document of unknown topic can help at all in determining the topic of other documents? Abundant unlabeled data does provide the ability to get an accurate estimate of how data is distributed, and it is often the case that the data distribution affects the likely assignments of labels. An example is illustrated in Figure 2-1, where the fact that data clusters is an indicator that all points within each cluster share the same class label.

[3]In early literature semi-supervised learning was also known as unsupervised classification, learning with labeled and unlabeled data, or with partially labeled data.

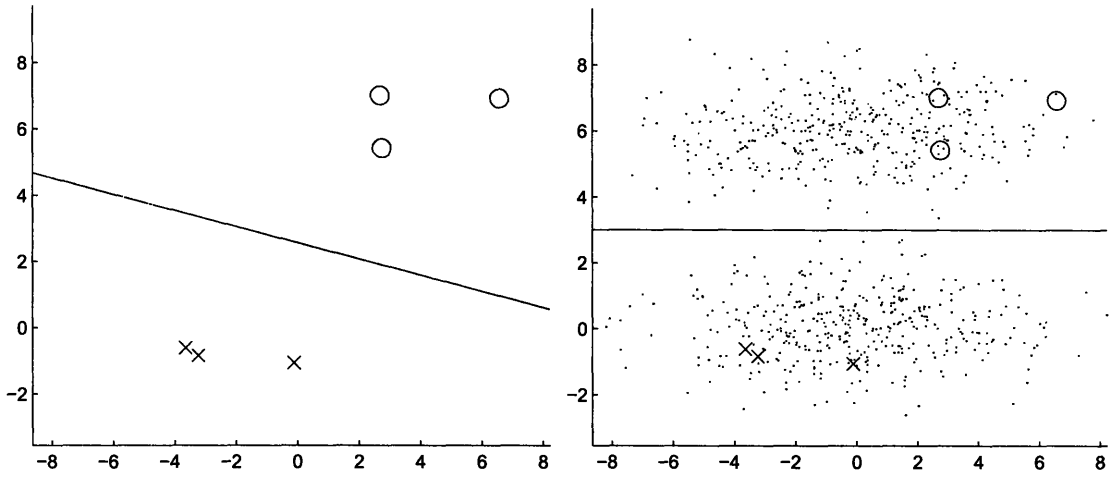

Figure 2-1: Sample semi-supervised learning problem. The goal is to produce the best linear decision boundary between the two classes. Left: decision boundary found by a supervised learning method, trained only on labeled data. Right: decision boundary found by a semi-supervised method, trained also on many unlabeled samples.

2.1 Semi-supervised learning approaches

There have been many recent and diverse approaches to semi-supervised learning, and we introduce the reader to the current work in the field.

To get a better understanding of the role of unlabeled data in classification, let us separate the distribution of data and labels from which the classification task is sampled, P_{XY}, into the marginal distribution $P_X(\mathbf{x}) = \sum_{y \in \mathcal{Y}} P_{XY}(\mathbf{x}, y)$, and the label distribution $P_{Y|X}(y|\mathbf{x}) = P_{XY}(\mathbf{x}, y)/P_X(\mathbf{x})$. We will refer to P_{XY}, P_X, and $P_{Y|X}$ as the *joint*, the *marginal*, and the *conditional*. Unlabeled data provides information about the marginal, while in classification we need to estimate the conditional. If the marginal and conditional distributions are related by a priori domain assumptions, than unlabeled data can contribute to the estimation of the conditional, resulting in semi-supervised learning.

We structure our presentation of existing approaches to semi-supervised learning in terms of the nature of the relation between the marginal and the conditional that each algorithm assumes. The marginal and conditional may be related due to a hard restriction on the joint, such as a parametric from that makes the marginal and the conditional dependent

through the value of the parameter; or through a soft constraint, a domain-specific bias that certain associations of marginals and conditionals are a priori more likely.

All semi-supervised learning algorithms operate on a set of domain-specific assumptions that relate the marginal and conditional, and their performance is highly dependent on the accuracy of these assumptions for the task at hand. Semi-supervised algorithms are even more sensitive to these assumptions that pure supervised ones, and cannot be distribution free, as the data marginal plays an important role in semi-supervised learning. The strength of the restrictions placed on the data distribution trade off robustness for potential in reducing error rates with the addition of unlabeled data.

We identify the following semi-supervised principles treated by the literature, that will be expanded in the rest of the chapter. Some principles may overlap.

hard constraints on the joint Parametric or other type of restrictive definitions of the family of the joint distribution P_{XY} introduce an implicit dependency between the marginal and the joint, that can be exploited for semi-supervised learning. Depending on the type of restriction, we distinguish:

parametric joint The joint distribution of features and labels is restricted to a parametric family such as a mixture of Gaussian classes. $P_X(\mathbf{x})$ and $P_{Y|X}(y|\mathbf{x})$ become functionally dependent through the unknown parameter of the family. This assumption leads to semi-supervised learning by maximum likelihood and the expectation-maximization (EM) algorithm or variations [26, 30, 50, 45, 19, 20, 16, 47].

redundantly sufficient features The data distribution is such that $\mathbf{x}$ contains multiple features each sufficient for supervised classification, yet complementary in the sense the labels obtained from some feature are not a good predictor of the labels derived from other features. For example, $\mathbf{x} = (f_1, f_2)$, and supervised learning is reasonably accurate either based on f_1, or on f_2. Then supervised classifiers based on individual features can be bootstrap each other over the unlabeled data, provided that they are sufficiently different. Notable algorithms that exploit this property include *co-training*, in which supervised

classifiers on individual features can then train off each other on unlabeled examples [9, 25, 17, 46, 31, 44, 12, 4], and some instances of the framework presented in this thesis, *information regularization.*

other types of latent structure Researchers considered various restrictions on the joint distribution that can be exploited for semi-supervised learning, including low-dimensional manifold representations of the joint distribution (see [48] for an overview of such methods), or a tree latent structure that generates both data and labels in [37].

biases on the label distribution This category of methods place a soft bias on the likely label distribution based on the received unlabeled data. We distinguish:

metric-based similarity This principle presumes the existence of a metric on $\mathcal{X}$ such that proximity under this metric is correlated with label similarity. Typically these methods employ a weighted graph that encodes the similarity [10, 52, 62, 15, 11, 63, 61, 64, 13, 59, 60]

relational similarity A characteristic task from this category is the presence of a number of relations that indicate label similarity, relations that are not necessarily linked to the traditional feature representation of the points. Relations can be citation structure, web links, objects that share a property, documents that share a word, etc. The distinction between relational and metric-based similarity is not strict, and many of the semi-supervised methods designed for one type of bias can be adapted to behave like instance of the other type [58, 56, 55, 28, 3]

data density bias We exploit an assumption that the conditional is smooth as a function of $\mathbf{x}$, and that smoothness depends on the data density near $\mathbf{x}$. Typically, the denser the region, the more restricted the variation of the conditional is. The same principle can be stated as the property that the decision boundary between the classes is likely to pass through regions of low-data density. The principle exploits the common knowledge that classes usually span dense clusters. This principle is related to *metric-based similarity*, in the sense that researchers sometimes use the metric to define a density with respect to which the variation

in the label must be smooth. [6] defines one such notion of smoothness from the geometry of a data manifold endowed with a data-dependent notion of distance. Other notable works include [49, 54, 53, 34]

Some semi-supervised algorithms are based on a mixture of semi-supervised principles from different categories. For example [39] combines features of co-training, metric-based similarity, and parametric models into a single algorithm. *information regularization*, the framework developed in this thesis and previous literature ([53, 21, 23, 22]), can account for both hard constraints and soft biases. In particular, it can account for both types of label similarity biases, it can act as a data-dependent smoothness prior, that enforces low-density separation, it can exploit redundancy in the features, and it subsumes the EM algorithm in parametric modeling of the joint.

2.2 Hard constraints on the joint

Semi-supervised methods in this category assume hard constraints on the possible distributions $P_{XY}(\mathbf{x}, y)$. If the true underlying data distribution satisfies indeed the assumed constraints, such methods can be quite powerful, and the contribution of unlabeled data to classification performance can be significant. The disadvantage of these methods is that the true distribution practically never belongs to the assumed family. Instead, the assumed family is only an approximation to the true distribution, and the accuracy of the approximation affects performance. Theory that bounds the penalty on performance given the distance between the true underlying distribution and the assumed family is lacking (even for supervised learning).

2.2.1 Parametric joint

This category of methods comprises parametric restrictions on the joint: P_{XY} that belongs to a parametric family $\{P_{XY}(\mathbf{x}, y; \boldsymbol{\theta})\,;\, \boldsymbol{\theta} \in \Theta\}$. For example, we could assume that $P_{X|Y}(\mathbf{x}|y\,;\,\boldsymbol{\theta})$ is Gaussian for each class. The parameters $\boldsymbol{\theta}$ can be estimated from the

training sample $\mathcal{T}$ by maximizing its log-likelihood:

$$\sum_{i=1}^{l} \log P_{XY}(\mathbf{x}_{\alpha_i}, y_{\alpha_i} ; \boldsymbol{\theta}) + \sum_{j=l+1}^{n} \log P_X(\mathbf{x}_{\alpha_j} ; \boldsymbol{\theta}) \tag{2.2}$$

The objective can be maximized by various algorithms, including the iterative *expectation-maximization* algorithm introduced in [26], that treats the label y as a missing variable when dealing with unlabeled samples. At every iteration, the EM algorithm "labels" the unlabeled data with its most likely expected label is a soft sense (by assigning a distribution over labels for each sample, instead of computing a hard label), and then re-estimates the parameters of the model as if the unlabeled data were labeled. The EM algorithm has been applied successfully to a number of semi-supervised domains [30, 50], including to classification of documents into topics [45]. The EM algorithm has the potential disadvantage that only guarantees a local optima of the likelihood.

Some parametric families posses the property of *identifiability*, that greatly increases the possible gains from semi-supervised learning when present. A family of joint distributions is identifiable with unlabeled samples if enough unlabeled samples restricts the space of possible joints to a finite number of choices [19].

For example, the Gaussian distribution family possesses this property. If each class is Gaussian, then the marginal P_X is a mixture of Gaussians with a known number of mixtures. Given enough unlabeled data sampled from an unknown such mixture, and without any labeled training data, we can estimate the parameters of the mixture exactly. The mixture determines a finite number of possible joints, corresponding to every possible permutation of the labeling of the clusters.

If the joint family is identifiable, unlabeled data can reduce significantly the number of labeled samples required for learning the classifier. On the other hand, if infinite unlabeled data still leaves a large set of possible distributions, semi-supervised learning is weaker, but still useful.

Introducing unlabeled samples into the likelihood function does not always improve classification, even though the same parametric model seems to work well on the labeled data alone. In other words, semi-supervised learning is more sensitive to parametric model assumptions than supervised learning, in the same way in which learning with incomplete

data is sensitive to model assumptions. Cohen [16] attempts to understand this phenomena. Nigam [45] alleviates the problem by stopping EM before convergence, or by artificially reweighting labeled vs. unlabeled samples in the log-likelihood function. In [20], Corduneanu and Jaakkola provide a justification for the reweighting in terms of model mismatch, and suggest a optimal weight. Rosenberg [47] introduces a *self-training* algorithm similar to EM, but with a more conservative E-step that labels only the unlabeled samples that satisfy a measure of confidence.

2.2.2 Redundantly sufficient features

One popular approach to semi-supervised learning applies to the situation in which we have multiple supervised classifiers on the same task, that are compatible, but not identical. The idea is to use each classifier to correct errors made by other classifiers on the unlabeled data, improving the training of all classifiers in the process.

A. Blum [9] introduces one of the earliest examples of this principle, *co-training*. Assume that the feature representation of each data point $\mathbf{x}$ consists of two components (f_1, f_2). Let P_{XY} be the distribution of data and labels, and P_{F_1Y} and P_{F_2Y} distributions obtained by marginalization. Co-training makes the following *compatibility* assumption about the task:

$$P_X(f_1, f_2) > 0 \;\Rightarrow\; \max_{y \in \mathcal{Y}} P_{F_1Y}(f_1, y) = \max_{y \in \mathcal{Y}} P_{F_2Y}(f_2, y) \tag{2.3}$$

In other words, f_1 and f_2 are compatible with each other only if the label obtained by looking only at f_1 is the same as the label obtained by looking only at f_2. Each feature is by itself sufficient for classification: if two samples (f_1, f_2) and (f_1', f_2') have $f_1 = f_1'$ or $f_2 = f_2'$, then they necessarily have the same classification label.

Here is how one may use the compatibility assumption to build a semi-supervised classifier. Given all the available unlabeled data, one may construct a graph with vertexes the data points, and edges between any two points that have the same f_1 or f_2. The edges determine connected components in the graph. The compatibility assumption asserts that all points in a connected component have the same label. Given a training set of observed labeled samples, now it is enough to construct a classifier on the connected components.

Since the numbers of connected components is usually much smaller than the number of points, the classifier requires fewer labeled samples (unless f_1 and f_2 are highly correlated, in which case co-training is unlikely to impact classification).

While the compatibility assumption used in this manner can be quite powerful, it is almost never satisfied in practice. In fact, most of the time the above algorithm will produce a single connected component in the graph, because every f_1 will be compatible with every f_2, even $P_X(f_1, f_2)$ is infinitesimal.

Let us analyze the legitimacy of the compatibility assumption. In supervised learning, especially in the PAC learning framework, we may assume that the feature representation f_1 completely determines the classification label, even though in reality f_1 is never a complete description of the object. Even though the label may be non-deterministic due to noise and other unknown factors, supervised classifiers based on the assumption, such as the SVM, are quite robust and perform well.

In this light it seems natural that if we looked at the other representation of the object, f_2, we should be able to make the same sufficiency assumption. However, the problem emerges when we start propagating these strong assumptions over a large data set for which we otherwise do not have any label information. A single noisy sample can connect two large components in the graph, ruining the algorithm. It is conceivable that if the task does not strictly abide to the compatibility assumption, SVM's constructed over $f1$, or over $f2$ would perform better than the same SVM's boosted over the unlabeled data by enforcing the assumption. Thus the same type of sufficiency assumption that works well for supervised learning, turns out to impose a much stronger prior if propagated over unlabeled data.

Nevertheless, there are many ways to *soften* the compatibility assumption of co-training and turn it into a powerful robust semi-supervised algorithm. Even the original article [9] propagates label information from one feature to the other only on points on which the classifiers are most confident, in an iterative bootstrapping fashion.

[46] uses the same *mutual bootstrapping* technique for information extraction, where an algorithm generates both the semantic lexicon, and the extraction patterns for the domain over unlabeled data. [17] uses similar semi-supervised methods for named entity recogni-

tion, where he shows that a model trained on spelling and context features that bootstrap each other performs better that a model trained on all features together. [31] uses yet another heuristic to ensure one classifier communicates label information to the other only in point that are labeled confidently. [44] compares co-training with semi-supervised EM empirically, and concludes that co-training is most effective when the features $f1$ and $f2$ are truly independent.

[4] provides a nice theoretical analysis of co-training, by formalizing the procedure of bootstrapping from sets of confident labels, and iterating. The authors formalize the property that the features must not be correlated by introducing the notion of *expansion*. The property is weaker than the true independence argued in [44]. Using this notion they are able to produce a bound on the error rate and the number of co-training iterations required to achieve it. On the downside, their analysis requires that the classifiers can learn only from positive samples.

[44] also discusses a probabilistic version of co-training, *co-EM*. A probabilistic classifier based on f_1 labels all points probabilistically (after being trained on labeled data initially). The probabilistic labels are then used to train a probabilistic classifier on f_2. We then label all points with the output of the second classifier, and return to training the first classifier. We iterate until convergence. The co-EM algorithm is still a heuristic, as it does not stem from optimization of a global objective. [12] substitute the EM algorithm in co-EM with a Support Vector Machine modified to output probabilistic outputs.

The information regularization itself framework also leads to one way of softening the co-training principle by turning the compatibility assumption into biases: points that share f_1 are biased to have similar labels, and points that share f_2 are also biased to have similar labels. The biases are resolved in the form of a global regularizer, that can be combined with the supervised classifiers.

2.2.3 Other types of latent structure

A distinct direction in semi-supervised learning consists of a number of algorithms that assume that data is embedded in a manifold that can be learned from unlabeled data alone.

A classical example comes from the area of computer vision. While images are usually represented by a large dimensional vector of pixels, they can be often explained by a small number of transformations, such as rotations, translations, changes in lighting, etc. The knowledge of the manifold can result in a classification domain that requires fewer labeled samples to learn, in the same way a vector space of lower dimensionaltiy has lower VC dimension, and lower learning complexity.

[6] considers a data manifold constructed from a metric correlated with the classification labels. He defines a geodesic consistent with the metric, that induces a smoothness criterion on the distribution of labels, based on its representation in a basis relative to the Appalachian operator. Since the variation in the label will be uniform with respect to the intrinsic structure of the manifold, the net result is that the labels are encouraged to vary in regions of low data density. Therefore this line of work is very similar to the semi-supervised learning on directed graphs that we will discuss on the later on in this chapter.

Other approaches that involve the manifold structure of the data exploit the fact that the manifold may have an intrinsic dimensionality lower than that of the vector space $\mathcal{X}$. A number of methods have been developed for learning the structure of the manifold (see [48] for a survey), and unwind it into a vector space of lower dimensionality. Methods vary in the assumptions, but many require that the manifold does not have holes, or that the intrinsic dimensionality is known in advance. These methods usually rely on computing eigenvalues of manifold operators, which makes them unpractical for a large number of unlabeled samples; yet, manifold learning need abundant unlabeled data to be reliable. Fast approximations do exist, and it is conceivable that manifold learning will come with fast, reliable, and robust algorithms.

[37] introduces yet another assumption about the latent structure of the data distribution that can be exploited for semi-supervised learning. Feature values and labels are generated from a tree model. The tree resembles a hierarchical clustering, with each observed point (labeled or unlabeled) being a leaf. To generate a set of labels and features for all points, we sample uniformly a $(\mathbf{x}, y)$ pair at the root, then propagate the value to the leafs, allowing random mutations to occur with small probability on each edge. According to the generative model, closer siblings in the tree are less likely to have different labels or feature

values than remote siblings. In order to perform semi-supervised learning, one can infer the posterior over the tree structures that explain a particular data set by looking at the similarities among unlabeled feature values. The posterior over tree structures can than be used to propagate the known labels to unlabeled points. The authors name the resulting classifier *Tree Based Bayes*. The efficient version of the algorithm uses an agglomerative clustering procedure to approximate the MAP tree. In the end, Tree Based Bayes introduces just another type of smoothness functional correlated with a metric on $\mathcal{X}$.

2.3 Biases on label distribution

This category consists of algorithms that leverage the unlabeled data by assuming that certain sets of points have similar labels due to their proximity with respect to a metric, or due to other relations that they satisfy. The local similarity biases can be used to regularize and assignment of labels to unlabeled data. Information regularization falls under this category, though it is flexible enough to also model other semi-supervised principles.

2.3.1 Metric-based similarity

Semi-supervised methods on undirected graphs

A popular category of semi-supervised algorithms treat the unlabeled training points points $\mathcal{D}$ as vertexes of an undirected graph, connected by weighted edges that express the strength of the similarity between labels. Given a few labeled vertexes, the algorithms resolve the labels of the remaining vertexes in a way that is consistent with the similarity weights encoded by the graph. Regularization on undirected graphs can be a powerful semi-supervised method, limited only by the type of interactions that it accepts: symmetrical, pairwise relations only. If the graph is sparse, that is if we consider only local interactions between the points, graph regularization is computationally efficient.

In what follows let $G = (\mathcal{D}, E)$ be the undirected graph, where E is the set of edges, and e_{ij} be the edge that connects $\mathbf{x}_i$ to $\mathbf{x}_j$, if it exists. Let w_{ij} be the weight that encodes the strength of the similarity bias between the endpoints, with 0 meaning that there is no a

priori bias. Also, let us assume that the classification is binary. Let y_i be the label of point $\mathbf{x}_i$, $y_i \in \{0, 1\}$ that needs to be determined, and z_i its soft version, $z_i \in [0, 1]$. The first l points in $\mathcal{D}$ are labeled with labels y_i^l.

Markov Random Fields The *Markov Random Field* approach to semi-supervised learning places a full probabilistic model on the graph of interactions, that account for the pairwise interactions:

$$\Pr[y_1, y_2, \ldots, y_n] \propto \exp\left(\sum_{e_{ij} \in E} \delta(y_i, y_j) f(w_{ij}) + \lambda \sum_{i=1}^{l} \delta(y_i, y_i^L)\right) \tag{2.4}$$

While the model has a clean theoretical formulation, it presents computational difficulties. *Markov Chain Monte Carlo* and *Belief Propagation* algorithms exist for estimating the probable label configuration given the weights and the labeled data (as a MAP estimate, or as an average over the posterior) [29], but they suffer from many local minima, if the computational time is limited to polynomial. Other methods, as follows, employ approximate objectives that are nevertheless computationally tractable.

Minimum Cut One of earliest semi-supervised algorithms on undirected graphs assigns labels to unlabeled vertexes by cutting the graph across edges such that there is no path connecting points of different classes [10]. All points sharing the same connected component are assigned the same label. The cut is optimized such that the sum of the weights of the edges it crosses is minimal. Minimize:

$$\sum_{e_{ij} \in E} w_{ij} |y_i - y_j| = \sum_{e_{ij} \in E} w_{ij} (y_i - y_j)^2 \tag{2.5}$$

subject to fixing y_i for the labeled samples. Optimization is efficient, with a max-flow algorithm.

The min-cut algorithm operates only on hard labels, thus there is no indication in the label confidence: we cannot tell that labels assigned near a decision binary may be noisy. Also, there are multiple label configurations that achieve the minimum, and the choice made by min-cut is arbitrary. The randomized version of the algorithm [11] alleviates the problems by injecting random noise into the weights.

One difficulty with the Minimum Cut algorithm is that the partitioning can be highly unbalanced. The cut may leave out a single positive point just because it is not connected to the rest of the points by enough, otherwise strong, edges. While other graph regularization methods suffer from the same inconvenience, the discrete nature of Min Cut (and almost discrete for the randomized version) makes it problematic to "adjust" the distribution of labels by setting the decision threshold.

Graph Kernels A number of semi-supervised algorithms on undirected graphs assign labels by minimizing a regularized loss function, with the objective of the following form:

$$\sum_{i=1}^{l} L(z_i, y_i^L) + \lambda \mathbf{z}^T S \mathbf{z} \tag{2.6}$$

where S is a matrix derived from the weights of the graph, and z_i's are relaxed to be any real numbers. [5] solves the generic optimization by linear algebra, and provides a theoretical analysis of generalization bounds.

If $S = \Delta = D - W$ is the graph Laplacian, where D is a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} w_{ij}$, we obtain the *Gaussian Random Fields and Harmonic Functions* algorithm [62]. The objective stems from a Gaussian Random Field approximation to the discrete Markov Random Field. The objective admits a unique minimum, that is a *harmonic function*: every label of an unlabeled point is the weighted average of its neighbors. To find the optimal labeling, it is enough to iterate averaging updates on every unlabeled point – convergence will occur exponentially fast in the number of updates.

The harmonic function algorithm is related to the semi-supervised manifold learning algorithm presented in [6]. Belkin derives a smoothness regularizer on the graph starting from the Laplacian operator on continuous manifolds. The harmonic function algorithm uses instead the discrete version of the Laplacian.

[59] uses a different regularization matrix, $S = D^{-1/2} \Delta D^{-1/2}$, the normalized graph Laplacian. As [33] argues, the normalized Laplacian has better balancing properties than the regular Laplacian.

Other authors have published more aggressive transformations of the Laplacian, such a transforming the spectrum of the Laplacian [15, 38, 63]. Some of these methods even learn the transformation of the Laplacian from data. Such aggressive transformations are akin to learning the structure of the graph from data.

Spectral Graph Transduction In [35], Joachims modifies the Min Cut with a normalized objective that removes it bias towards unbalanced classes. The exact form of the modified objective is NP hard to optimize:

$$\min \frac{\sum_{e_{ij} \in E} w_{ij} |y_i - y_j|}{\sum y_i \cdot \sum (1 - y_i)} \tag{2.7}$$

subject to fixing y_i for the labeled samples.

The objective can be made tractable by relaxing it and optimizing soft labels z_i instead of hard labels:

$$\min_{\mathbf{z}} \quad \lambda \mathbf{z}^T L \mathbf{z} + (\mathbf{z} - \boldsymbol{\gamma})^T (\mathbf{z} - \boldsymbol{\gamma}) \tag{2.8}$$

$$\text{s.t.} \quad \mathbf{z}^T \mathbf{1} = 0 \text{ and } \mathbf{z}^T \mathbf{z} = n \tag{2.9}$$

Here L is the *normalized graph Laplacian* $D^{-1/2} \Delta D^{-1/2}$, thus the objective is similar to that of learning with local and global consistency [59]. The strength of the regularization is given by λ. The labeled training points are encoded into the vector $\boldsymbol{\gamma}$:

$$\gamma_j = \begin{cases} 0, & \text{for unlabeled samples} \\ \sqrt{\frac{\sum (1 - y_i)}{\sum y_i}} & \text{for positive samples} \\ -\sqrt{\frac{\sum y_i}{\sum (1 - y_i)}} & \text{for negative samples} \end{cases} \tag{2.10}$$

Spectral Graph Transduction optimizes the above objective by spectral methods. SGP has better balancing properties than Min Cut.

Semi-supervised methods on directed graphs

Often the relations on which we base the semi-supervised bias are not symmetric (for example, in k-NN, the relation between the center of the region, and a point belonging to it is asymmetric). This motivates a class of semi-supervised graph regularization algorithms that work on directed graphs.

Markov Random Walks [52] studies the graph regularization problem by defining the following *Markov Random Walks* process on the graph. The weights w_{ij} on the edges of the graph induce a transition probability $p_{i \to j}$ from every node i to its neighbor j, where self transitions $p_{i \to i}$ are allowed, and occur with higher probability than other transitions.

[52] defines the following process for generating labels from nodes in the graph, given that the probabilities that we need to estimate, $P_{Y|X}(\cdot|\mathbf{x})$, are known:

1. Select a node i uniformly at random from the graph.
2. Sample a label y according to $P_{Y|X}(\cdot|\mathbf{x}_i)$.
3. Perform t random transitions in the Markov Chain, following edges according to the transition probabilities. After the transitions, say we reached node k.
4. Emit the sampled y as if it were generated by node k.

Then we can estimate $P_{Y|X}(\cdot|\mathbf{x})$ at every node such that the labels emitted by the labels points according to the process described above match their observed label as close as possible.

The semi-supervised bias imposed by the random walk depends strongly on the parameter t (the number of transitions), as well as on the importance of the self transition in comparison to other transitions (the two parameters are related in their contribution). For instance, if $t = 1$, only immediate neighbors of observed labeled samples will be labeled. If $t \to \infty$ all unlabeled points will get the same label.

The work was originally aimed at undirected graphs, but the construction of the Markov Chain works in the same way even if the graph is directed.

Conditional Harmonic Mixing *Conditional Harmonic Mixing* [13] is a transductive graph semi-supervised framework that differs from the mainstream semi-supervised graph regularization methods in the following respects:

- It it based on a directed graph, thus it can model asymmetric influences between labels

- It is a probabilistic graphical model
- As opposed to Bayesian networks, it allows label distributions that are inconsistent with a global joint. This makes the label propagation algorithm efficient, provably convergent to the unique optimal parameters on any type of graph.

Let $\mathcal{D}$ be the set of all training/testing points for which we need to determine probabilistic labels $P_{Y|X}(\cdot|\mathbf{x})$, $\mathbf{x} \in \mathcal{D}$. The CHM model defines a semi-supervised prior by means of a directed graph on $\mathcal{D}$, and a set of given conditional probability distributions P_Y^{ij} for every directed edge $i \longrightarrow j$ in the graph. The semi-supervised assumption is that if the edge $i \longrightarrow j$ is present, than $P_{Y|X}(y|\mathbf{x}_j)$ is similar to its estimate coming from i: $P_{Y|X}(y|\mathbf{x}_i)P_Y^{ij}(y)$. CHM places an objective that quantifies the degree of similarity, and that must be minimized in order to find the labels. The objective is the average *Kullback-Leibler Divergence*, over all incoming links:

$$\sum_{i \text{ s.t. } i \longrightarrow j} \mathrm{KL}\left(P_{Y|X}(y|\mathbf{x}_i)P_Y^{ij}(y) \,\|\, P_{Y|X}(y|\mathbf{x}_j)\right) \tag{2.11}$$

Minimizing this objective leads to an update rule in which each label is updated by the average of its estimates coming from each edge pointing to it. The update provably converges to a *harmonic function*, in which each node equals the average of incoming nodes (multiplied by the fixed conditional distributions attached to edges).

As in all graph regularization methods (including information regularization), there is no good answer to learning the graph and its parameters. CHM provides a way of learning the conditional distributions P_Y^{ij}, but only in the case in which all points are labeled. In practice, it turns out that identity conditional distributions do as well as learned ones. To moderate the difficulty of learning the graph, CHM advocated model averaging: average the resulting labels over a variety of probable graphs.

As other graph regularization methods, CHM can be used to update the probabilities that result from a supervised classifier. This can be done by providing a duplicate set of nodes, in which each node is connected only to its corresponding node in the main graph. The duplicate nodes have their labels fixed to the output of the supervised classifier.

hub-authority [60] introduces a semi-supervised algorithm on directed graph following the hub-authority paradigm. The authors convert the directed graph to a bipartite graph, then define regularizer that is asymmetric in the roles of hubs and authorities (the two layers of the bipartite graph). The resulting algorithm is an iteration in which the label probability at each hub is obtained as an average of the label probabilities of the authorities it liked to it, and the label probability of each authority is an average of the labels of connected hubs. The two averages are weighted and normalized differently.

2.3.2 Relational similarity

We distinguish from the mainstream graph regularization semi-supervised algorithms the tasks for which we can identify heterogeneous label similarity biases, that cannot be modeled well by standard graphs. Such biases typically come in the form of relations derived from different sources of information, that need to be treated differently. The model is that certain data points should have similar labels because they satisfy a certain relation. Examples of relations include documents having a word in common, co-cited papers, genes whose proteins interact, books grouped by author, or by words in the title, and so on. The field of *relational learning* deals with learning under such type of relational biases. It can be viewed as semi-supervised learning, with the semi-supervised bias derived from external sources of information.

Yarowsky [58] illustrates relational semi-supervised on the problem of word sense disambiguation. The author considers the following types of a priori biases of label similarity:

- instances of an word occurring in the same document are likely to be disambiguated with the same meaning;
- instances of an word that have similar context are also likely to be disambiguated with the same meaning.

His algorithm is a heuristic that propagates labels from a small number of "seed" training points alternatively across these regions, and with the help of a decision-list supervised

classifier. While the algorithm makes sense, there is no theoretical support for it – the iteration does not even optimize an objective. [1] studies the original Yarowsky algorithm from a theoretical perspective, and produces a number of variations, each of which optimizes a formal objective. Nevertheless, our information regularization framework can model the same type of biases and it is cleaner, with theoretical backing.

[3] encounters relational biases on a task of classifying persons in webcam video. The authors identify label similarity biases among webcam frames derived from proximity in time, proximity in the color histogram, and similarity (in terms of Euclidean pixel-wise distance) of faces detected in the frame. While the similarities are qualitatively different, the authors still use an off-the-shelf undirected graph regularization semi-supervised method, with good results. They do not have any results to compare on a semi-supervised method that can model (and weight) the different types of similarity biases differently.

The *Probabilistic Relational Model* framework [27] was specifically designed for learning on data from relational databases, in which the relations between constituents are in fact semi-supervised biases on the attributes of those constituents. The framework is pioneering in distinguishing between *relational templates* and the instances of the relations themselves. The edges in a standard graph for semi-supervised learning are instances of relations, that have been generated by a certain rule (template). For example, one rule can be proximity in the Euclidean metric, and another rule can be co-citation. Given a task, there are many possible templates for generating a graph for semi-supervised learning. The PRM framework provides algorithms for learning the best set of templates for a particular domain, and for inference once the templates have been learned. Therefore, the framework can learn the characteristics of the domain, and use those characteristics to define semi-supervised biases for a particular instance of a problem from the domain.

In the PRM framework, the templates induce a full Bayesian network with the parameters of the template copied in all links between data points that have been generated from the template. Thus semi-supervised inference has the same difficulties of inference as inference in loopy Bayesian networks. The PRM framework does not offer the convexity guarantees of other semi-supervised algorithms on graphs.

[27, 28, 56] provide algorithms for learning the templates of a PRM. Taskar introduces

the *Relational Markov Network* framework [55], that differs from PRM's in that it uses undirected models. The concept is similar: relational templates are rolled into a probabilistic Markov network.

2.3.3 Data density bias

A number of semi-supervised classifiers exploit unlabeled data by assuming that the conditional varies smoothly with $\mathbf{x}$, where the smoothness depends on the data-density, as estimated from unlabeled samples. Methods that exploit metric-based similarity, as discussed in the previous section, make implicit data-dependent smoothness assumptions. Here we will discuss only methods that make this assumption explicit.

Adaptive regularization

Schuurmans [49] considers a regularizer that penalizes conditionals that behave differently on the labeled training data versus on unlabeled data. This regularization principle is based on the observation that overfitted conditionals typically behave erratically on unlabeled data, much differently than on data with observed labels. For example, a polynomial of large degree would fit any labeled training set with limited number of samples, but the value of the polynomial would vary by large amount on unlabeled samples; a linear function would be smoother on the unlabeled data, even if it does not fit labeled data well.

The author measures the *behavior* of a conditional $P_{Y|X}(y|\mathbf{x};\theta)$ by computing the distance to a fixed pre-selected function $\phi(\mathbf{x}, y)$. The regularizer is the difference between this distance measured empirically on observed labeled data, and the distance computed on the data distribution estimated from unlabeled data:

$$R(\theta) = \mid \frac{1}{n}\sum_{i=1}^{n}\text{dist}(P_{Y|X}(y_n|\mathbf{x}_n;\theta), \phi(\mathbf{x}_n, y_n)) - \int \text{dist}(P_{Y|X}(y|\mathbf{x};\theta), \phi(\mathbf{x}, y))d\hat{P}_{Y|X}(y|\mathbf{x})d\hat{P}_X(\mathbf{x}) \mid \tag{2.12}$$

The regularizer is a semi-supervised method in the sense that focuses on regions of high unlabeled data density — *erratic behavior* is penalized less in regions where unlabeled data is scarce. In contrast to classical regularization, it is interesting because it adapts to both

labeled and unlabeled data. Compared to other semi-supervised regularizers that pay close attention to the topological relationship between high density clusters, it is quite crude because it averages out the topological structure of the data (in a similar way to using a single region in information regularization – see Chapter 3).

Transductive Support Vector Machines

A *Support Vector Machine* (SVM) is a non-parametric supervised classifier that seeks a linear decision boundary separating negative examples from positive examples, such that the distance to the closest training example (the *margin*) is maximal. Joachims [34] extends the SVM to the situation in which an unlabeled training set is also available. The *Transductive Support Vector Machine* (TSVM) makes the same assumption as the regular SVM that the decision boundary separates well the two classes, but also uses unlabeled data in evaluating the margin, that must be maximized. Since we do not know in advance on which side of the decision boundary the unlabeled points should be (not knowing their class), a naïve approach would take exponential time to try all label combinations. Joachims avoids this complexity with an approximate algorithm that initializes the labels with the SVM values, then flips them as long as the margin improves. TSVM training is as efficient as regular SVM training, and the TSVM can be also extended to non-linear decision boundaries by using kernels.

Kernel expansions with unlabeled data

Szummer [54] introduces a transductive semi-supervised classification algorithm that uses a kernel density estimator from the unlabeled data to smoothly assign labels. The model is that the joint distribution can be expressed as a kernel density estimator on the training data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$:

$$P_{XY}(\mathbf{x}, y) = \frac{1}{n} \sum_{i=1}^{n} Q_{Y|X}(y|i) K(\mathbf{x}, i) \tag{2.13}$$

where $K(\mathbf{x}, i)$ is a kernel density centered at $\mathbf{x}_i$ and $Q_{Y|X}(y|i)$ is a parameter associated with $\mathbf{x}_i$ that needs to be estimated. Given the above definition of the joint, we can express

the label at any point:

$$P_{Y|X}(y|\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} Q_{Y|X}(y|i)K(i|\mathbf{x}) \tag{2.14}$$

where $\sum_{i=1}^{n} K(i|\mathbf{x}) = 1$.

Given a labeled training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, one can estimate $Q_{Y|X}(y|i)$ at the unlabeled points such that $P_{Y|X}(y_j|\mathbf{x}_j)$ computed from equation (2.14) are maximized, in a maximum likelihood sense. Once the parameters are estimated, one can compute the label at any point according to (2.14). The estimated labels will be smooth by means of the kernel density estimator. They are also less likely to vary in dense regions, because in a dense region they are close to many unlabeled points that impact them.

The algorithm is computationally efficient, robust to unlabeled outliers, but somewhat inflexible in terms of the type of semi-supervised biases it can model.

Standard information regularization

Szummer [53] introduced the original version of *information regularization*, in terms of a smoothness principle that states that variation in the conditional $P_{Y|X}(y|\mathbf{x})$ as a function of $\mathbf{x}$, as measured by an information theoretical objective, should be penalized in proportion to the data density. The framework presented in this thesis expands the information theoretical objective to include a wider range of semi-supervised principles, including data-dependent smoothness, parametric joint families, redundancy of sufficient features, metric and relational-based similarity.

2.4 Combined methods

A number of semi-supervised methods have the capability of modeling semi-supervised principles from more than one of the categories introduced above.

Zhu [64] presents an algorithm that combines features of transductive graph-based regularization, and inductive classification with parametric models of the joint. The objective of the algorithm is a linear convex combination of the objectives of harmonic graph reg-

ularization and log-likelihood of the parametric model. The parameters can be estimated with a variant of the EM algorithm.

Krishnapuram [39] introduces an algorithm that combines parametric models, metric-based label similarity, and relational-based similarity and co-training. The semi-supervised principles are weighted relative to each other, with weights trained from data. Also, the algorithm learns the relative weighting between labeled and unlabeled data. The authors also demonstrate semi-supervised active label selection. The algorithm is inductive, and achieved good experimental results.

Information regularization is one such framework that can model various semi-supervised principles, and combine them.

Chapter 3

The information regularization framework

3.1 Similarity biases in semi-supervised learning

In many classification tasks it is natural to express the prior information about the nature of the joint distribution on data and labels as a series of *similarity biases*. Such biases reflect the a priori belief that it is likely that points in a certain subset $\mathcal{R}$ of $\mathcal{X}$ have similar densities $P_{XY}(\mathbf{x}, y)$. It is best to explain the type of similarities we refer to by illustrating biases that other researchers found appropriate for describing various tasks.

web page categorization [56, 23] The features are the text of the web pages, and the class label is the topic of the web page. Additional information comes in the form of links between pages.

- for each word in the vocabulary, pages that have that word in common are biased to have similar topics
- two pages that are linked or link to many common pages are biased to have similar topics
- pages that are pointed to by links that have words in common are biased to have similar topics

word sense disambiguation [58] The task is to determine the sense of words that can have multiple meanings. Features: the neighboring words of an instance, as well as the identity of the document in which the instance appears.

- multiple instances of the same word appearing in the same document are biased to have the same meaning
- multiple instances of the same word that have common words in their context are likely to have the same meaning
- $P_{Y|X}(y|\mathbf{x})$ of each instance is biased to be similar to the distribution of a decision-list classifier

named entity classification [17] Collins approaches the task of classifying proper names from a large corpora of text into *Person*, *Organization*, or *Location*. The careful feature selection amounts to implicit similarity biases. He identifies two types of rules/biases: *contextual* rules refer to the context of the entity as determined by a parser; *spelling* rules refer to features derived from the spelling of the word:

contextual Entities which have the same context, or the same type of context (appositive or prepositional) are biased to be of similar type

spelling Entities that have any of the following properties in common are biased to be of similar type: both are all-capital letters; both are all-capital with full periods (`N.Y.`); both contain non-alphanumeric characters; having the same exact spelling.

person identification in video [3] The features are video frames (color pixel images), and the label is the name of the person present in the frame (or "unknown" person, or no person)

- frames within a short time interval are likely to contain the same person
- frames with similar color histogram that are not to far in time are likely to contain the same person. If the frames are too distant in time the clothing may change, rendering the color cue unreliable.

- proximity in pixel-wise Euclidean distance between detected faces is an indicator of label similarity.

collaborative prediction Given a set of reviews/raters, and a set of reviewed objects, the task is to predict a label associated with each object. For example, the objects can be article submissions at a conference, rated by peer reviewers, and we need to predict the quality of each article. Because each reviewer has a different style, objects reviewed by the same reviewer are naturally biased to share certain similarities in how the reviews relate to the labeling.

The above list is by no means exhaustive, with notable natural similarity biases ubiquitous in bioinformatics, information extraction, tracking objects in video [51].

The notion of similarity bias exhibited by various tasks is quite broad, ranging from similarity in class labels $P_{Y|X}(y|\mathbf{x})$, to similarity in the parameters that fully characterize the joint model in the case of tracking from video. It will be apparent that the information regularization framework is able to encompass a wide range of such biases. For clarity, in the next section we introduce the framework on a specific scenario. Subsequently, we will define the information regularization framework.

3.2 Information regularization for categorization of web pages

We illustrate the concept of information regularization on the problem of determining the topics of a collection of web pages $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ out of a finite number of choices $\mathcal{Y}$, where $\mathbf{x}$ is a feature representation of the body of the page. Besides the actual contents of the page, we also have access to the hyperlinks among pages from $\mathcal{D}$. We observe the topics $y_1, y_2, \ldots, y_l$ of l pages from our collection. Supervised document classification trains a model from labeled data alone $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$. Here we would like to use all available data, as well as the information provided by the link structure.

3.2.1 Non-parametric biases

We identify two types of a priori biases in how the labels should be distributed among data points:

- for each word in the vocabulary (or feature), web pages that have that word in common are likely to have similar topics.

- Consider the collection of words that are in the anchor of the hyperlinks that point to a certain web page. These words, that come from other pages, are typically quite indicative of the topic of a document. We represent this by the bias that web pages having words in common in anchors that link to them are likely to have similar topics.

The rules we identified above are not strict constraints, but only biases. It would be impossible to satisfy them exactly at the same time. Nevertheless, among alternative labelings that are equally likely from the point of view of the observed labeled data, we prefer the labeling that is the most consistent with the identified biases.

Let us formalize the semi-supervised biases we talked about. In general, we can represent one such semi-supervised bias by a subset R of the training points $\mathcal{D}$ (or *region*) that we believe it contains web pages of related topics. In the case of the biases identified above, for every word in the vocabulary we can define a region R of all web pages containing that word; or of all web pages that contain that word in the anchor of some link pointing to it. The assumption is that the labels of points from R, $P_{Y|X}(y|\mathbf{x}_\alpha)$, $\alpha \in \{1, 2, \ldots, n\}$, are likely to be similar.

We choose to represent the semi-supervised bias of region R by a *regularization penalty*, an objective that can be computed from the labels of points in R that quantifies the degree to which the labels are similar. The objective has the property that the smaller it is, the more similar the labels are, and that it is 0 if all labels are equal. In building a classifier for web documents we seek to minimize the regularization penalty on region R, among other constraints that take into consideration the labeled training data.

Had we need to quantify the similarity between the labels of two documents only, indexed in $\mathcal{D}$ by say α_1 and α_2, we could have used the Kullback-Leiber Divergence (in short

KL-divergence) [24], that is widely used for measuring the distance between two distributions:

$$\mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_{\alpha_1}) \,\|\, P_{Y|X}(\cdot|\mathbf{x}_{\alpha_2})\right) = \sum_{y\in\mathcal{Y}} P_{Y|X}(y|\mathbf{x}_{\alpha_1}) \log \frac{P_{Y|X}(y|\mathbf{x}_{\alpha_1})}{P_{Y|X}(y|\mathbf{x}_{\alpha_2})} \tag{3.1}$$

KL-divergence in the form presented above does not suffice for our purpose because it is not symmetric, and it does not generalize to more that two points. Instead, we measure the distance from each point to an average distribution:

$$\frac{1}{2}\mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_{\alpha_1}) \,\|\, Q^\star_Y\right) + \frac{1}{2}\mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_{\alpha_1}) \,\|\, Q^\star_Y\right) \tag{3.2}$$

where $Q^\star_Y(y) = (P_{Y|X}(y|\mathbf{x}_{\alpha_1}) + P_{Y|X}(y|\mathbf{x}_{\alpha_2}))/2$.

It is easy to extend the above objective to quantify the similarity between the labels in a set of documents R:

$$\frac{1}{|R|}\sum_{\alpha\in R} \mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_{\alpha}) \,\|\, Q^\star_{Y|R}(\cdot|R)\right) \tag{3.3}$$

where $Q^\star_{Y|R}(y|R) = \frac{1}{|R|}\sum_{\alpha\in R} P_{Y|X}(y|\mathbf{x}_\alpha)$.

Note that $Q^\star$ could be obtained minimizing the objective, had we allowed it to vary:

$$Q^\star_{Y|R}(\cdot|R) = \arg\min_{Q_{Y|R}(\cdot|R)} \frac{1}{|R|}\sum_{\alpha\in R} \mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_{\alpha}) \,\|\, Q_{Y|R}(\cdot|R)\right) \tag{3.4}$$

This allows us to quantify the similarity bias on region R by the following objective, that we seek to minimize as a function of the labels:

$$\min_{Q_{Y|R}(\cdot|R)} \frac{1}{|R|}\sum_{\alpha\in R} \mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_{\alpha}) \,\|\, Q_{Y|R}(\cdot|R)\right) \tag{3.5}$$

Turning the objective into a minimization may seem unnecessary at a first sight, but it does allow us to easily generalize to other useful definitions of $Q^\star_{Y|R}(\cdot|R)$, that cannot even be represented in analytic form.

Multiple regions

We can define one such region for every word in the document (the first type of bias mentioned), and for every possible word in the anchor of links pointing to web pages (the second

type of bias mentioned). If $\mathcal{R}$ is the collection of regions, therefore we can represent the global semi-supervised bias as the following *information regularizer*:

$$I(P_{Y|X}) = \sum_{R \in \mathcal{R}} P_R(R) \min_{Q_{Y|R}} \frac{1}{|R|} \sum_{\alpha \in R} \mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_\alpha) \,\|\, Q_{Y|R}\right) \tag{3.6}$$

We have weighted the contribution of each region by $P_R(R)$ to acknowledge the fact that some semi-supervised biases are more important than others and should be satisfied first.

In web page categorization we apply the regularizer as a penalty to the standard log likelihood on the labeled samples:

$$\sum_{i=1}^{l} \log P_{Y|X}(y_i|\mathbf{x}_i) - \lambda I(P_{Y|X}) \tag{3.7}$$

where λ is a positive number that represents the strength of the regularizer (or how informative the task prior is).

Maximizing the regularized log likelihood would select among labelings that are otherwise equally likely the choice that is most consistent with the semi-supervised biases imposed by the regions.

3.2.2 Parametric biases

In many tasks domain knowledge dictates that the data distribution takes a particular parametric form. For example, one parametric model that has been successful in document categorization is *naïve Bayes* [45]. In multinomial naïve Bayes, we represent each document by a bag of words, and assume that the words are generated independently of each other if the topic of the document is known. Thus the probability of generating a document can be written as:

$$P_{XY}(\mathbf{x}, y) = P_Y(y) \prod_{w \in \mathbf{x}} P_{W|Y}(w|y) \tag{3.8}$$

for every word w contained in the bag-of-words feature representation $\mathbf{x}$ of a document. Equivalently, P_{XY} belongs to a distribution family $\mathcal{M}$ parametrized by a vector $\boldsymbol{\theta} = \{P_Y(y), y \in \mathcal{Y};\ P_{W|Y}(w|y), y \in \mathcal{Y}, w \in \mathcal{V}\}$.

Typically in supervised learning we enforce the parametric form of P_{XY} dictated by domain knowledge by optimizing the regularized log likelihood under the constraint $P_{XY} \in$

$\mathcal{M}$. This approach can be quite powerful if the joint truly belongs to $\mathcal{M}$, it is well-studied in the supervised learning literature.

However, there are many situations in which the parametric generative assumption on the joint is not really satisfied, but a member of the parametric family is still a good approximation to the joint. For example, experts agree that the naïve Bayes model of documents is far from realistic, because words will not occur independently of each other even if the topic is known. Still, supervised naïve Bayes performs well in practice, because the approximation is good enough.

Recognizing that parametric families are only reasonable approximations to the real joint, when training a classifier we would like to have the ability to express the bias that the joint is similar to a certain parametric family, without imposing the strict constraint that the joint actually belongs to that parametric family. This ability would be quite powerful especially in situations in which different sources of information seem to indicate different parametric models, that would be incompatible if viewed as strict constraints.

In the case of web page categorization, we would like to combine the non-parametric semi-supervised biases that we described in the previous section with fact the the naïve Bayes distribution is a reasonable approximation to the joint.

If $\mathcal{M}$ is the naïve Bayes family of distributions parametrized by $\boldsymbol{\theta}$, we can express the bias that the joint is almost naïve Bayes by controlling the KL-divergence distance between the joint and any member of the family $\mathcal{M}$[1]:

$$\min_{\boldsymbol{\theta}} \mathrm{KL}\left(P_{XY} \,\|\, Q_{XY}(\cdot\,;\boldsymbol{\theta})\right) \tag{3.9}$$

We can incorporate this bias as another additive term in the information regularizer, weighted by a constant τ that expresses the importance of the word constraints relative to the naïve Bayes constraint:

$$\begin{aligned} I(P_{Y|X}) = \tau \min_{\boldsymbol{\theta}} \mathrm{KL}\left(P_{XY} \,\|\, Q_{XY}(\cdot\,;\boldsymbol{\theta})\right) + \\ + (1-\tau) \sum_{R \in \mathcal{R}} P_R(R) \min_{Q_{Y|R}} \sum_{\alpha \in R} \mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}_\alpha) \,\|\, Q_{Y|R}\right) \end{aligned} \tag{3.10}$$

[1]Minimizing the KL-divergence is related to likelihood maximization. If P_{XY} is given, that the optimal θ is the maximum likelihood estimate.

The understand the effect of τ, note that if $\tau = 0$ we classify only according to the parametric bias. Minimizing the KL-divergence by itself is equivalent to maximizing the log-likelihood of the data, where the label is treated as a latent variable. Thus $\tau = 0$ reproduces the EM algorithm with naïve Bayes for semi-supervised learning, as in [45]. As τ increases we are also incorporating the other biases. If $\tau = 1$, we completely ignore the naïve Bayes parametric model. $\tau = 1$ is less than ideal though, because $\mathcal{R}$ typically does not cover the entire data set, and it would be impossible to set the labels of documents not covered by $\mathcal{R}$ without the naïve Bayes model.

We will demonstrate the effectiveness of the above information regularizer for categorization of web pages in Chapter 6.

3.3 The generic information regularization framework

We introduce the information regularization framework in its generic form. To facilitate the presentation let us assume that the objects involved in classification are points in an Euclidean space, where $\mathbf{x} \in \mathcal{X}$ denotes the coordinates of a point. In general $\mathcal{X}$ in the space of the feature vectors $\mathbf{x}$ that represent the data.

The goal of the learning algorithm is to predict a certain quantity associated with each point available to the algorithm. We denote by $\mathcal{A}$ the set of points available to the algorithm, and by z the quantity to be predicted.

Let us discuss in more detail the meaning of $\mathcal{A}$ and z. In standard supervised learning, $\mathcal{A}$ and $\mathcal{X}$ are always the same, because the feature vector $\mathbf{x}_\alpha$ is all we know about the object α. The classifier would not be able to distinguish between α_1 and α_2 if $\mathbf{x}_{\alpha_1} = \mathbf{x}_{\alpha_2}$. In our semi-supervised information regularization framework we may introduce other information about the objects then their feature representation. In particular, the semi-supervised similarity biases will be defined at the level of the objects α, not at the level of their feature representation $\mathbf{x}$. If two different objects have the same $\mathbf{x}$, they may still participate in our similarity biases differently. For example, in the web page classification task, where $\mathbf{x}$ denotes the body of the web page, we would like to group together pages based on information external to $\mathbf{x}$, such as the link structure of the web pages.

In a typical inductive classification setting, the algorithm must have the ability to assign labels to any point in the space $\mathcal{X}$. In this case $\mathcal{A} = \mathcal{X}$, as all the points are available to the algorithm. On the other hand, if the classification problem is transductive, than the classifier can predict labels only for the received training data $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. In this case $\mathcal{A} = \mathcal{D}$. In general, we should think of $\alpha \in \mathcal{A}$ as an unique identifier for the object to be classified, and $\mathbf{x}_\alpha$ as the feature representation of that object. It may be that different objects with different α's have the same feature representation, because $\mathbf{x}$ is not a complete description of the object. $\mathcal{A}$ is the space of objects to be classified, and $\mathcal{X}$ is their representation.

z is the quantity around which we define the notion of similarity bias. Normally $z = y$, the classification label, in that the biases represent similarity between the labels of the objects. In some cases we will want to define z differently. For example, if $z = (\mathbf{x}, y)$, then the semi-supervised bias can measure similarity in the distribution $P_{XY} = P_X(\mathbf{x})P_{Y|X}(y|\mathbf{x})$, that includes information about both $\mathbf{x}$ and y.

The goal of the learning algorithm is to estimate $P_{Z|A}(z|\alpha)$ for every $\alpha \in \mathcal{A}$, where the output is probabilistic to reflect uncertainty in the true value of z for a particular α. The input to the algorithm is a finite training set of points $\mathcal{D} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ with their associated features $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. We also observe z for the first l points: $z_1, z_2, \ldots, z_l$. The rest of $n - l$ training points have unknown z.

Typical supervised learning algorithms are trained by minimizing a loss function $\text{loss}_{\mathcal{D}}(P_{Z|A})$ defined on the samples for which z is observed. One standard loss is the log-likelihood of the labeled data, $\sum_{i=1}^{l} \log P_{Z|A}(z_i|\alpha)$. Since $\text{loss}_{\mathcal{D}}(P_{Z|A})$ does not incorporate unlabeled samples, supervised learning does not take advantage of all available information, and can be suboptimal.

Semi-supervised learning incorporates all available information, labeled or unlabeled. In information regularization we do so by providing a regularization penalty that can be applied to the classical supervised loss:

$$\min_{P_{Z|A}} \text{loss}_{\mathcal{D}}(P_{Z|A}) + \lambda I(P_{Z|A}) \tag{3.11}$$

Here λ is the strength of the regularization. If $\lambda = 0$ the algorithm is purely supervised.

However, we will see interesting semi-supervised algorithms for $\lambda \to 0$, that use the unlabeled data.

In general, we may want to restrict the possible conditionals $P_{Z|A}$ over which we minimize (3.11) to some distribution family $\mathcal{F}$. For example, $\mathcal{F}$ can be a parametric family, such as Gaussian class distributions $P_{X|Z}(\mathbf{x}|z)$. Many supervised classifiers enforce such parametric constraints on the joint. Allowing to restrict the minimization to $\mathcal{F}$ provides an easy way of converting a good parametric supervised classifier to a semi-supervised one:

$$\min_{P_{Z|A} \in \mathcal{F}} \text{loss}_{\mathcal{D}}(P_{Z|A}) + \lambda I(P_{Z|A}) \tag{3.12}$$

The regularizer $I(P_{Z|A})$ encodes the semi-supervised bias (task prior). In information regularization, we define the semi-supervised bias by the help of a set of regions $\mathcal{R}$, whose elements R are subsets of $\mathcal{A}$. Each region encodes a semi-supervised bias at the local level, by biasing $P_{Z|A}(z|\alpha)$ to be similar for all $\alpha \in R$.

As in web page classification, we compute the similarity among conditionals $P_{Z|A}(z|\alpha)$, $\alpha \in R$, by evaluating their average distance to a common distribution $Q_{Z|R}(z|R)$, typically the average of all conditional in R, where the distance is the KL-divergence:

$$\begin{aligned} &\int_{\alpha \in R} \pi_{A|R}(\alpha|R) \text{KL}\left(P_{Z|X}(\cdot|\mathbf{x}_\alpha) \,\|\, Q_{Z|R}(\cdot|R)\right) = \\ &\qquad \int_{\alpha \in R} \pi_{A|R}(\alpha|R) \int_{z \in \mathcal{Z}} P_{Z|A}(z|\alpha) \log \frac{P_{Z|A}(z|\alpha)}{Q_{Z|R}(z|R)} dz \end{aligned} \tag{3.13}$$

Note that we have introduced a distribution $\pi_{A|R}(\alpha|R)$ over R that expresses the relative contribution of each point in R to the similarity measure. This weighting is normally uniform, but we envision scenarios in which we know a priori that some objects should not contribute to the similarity measure as much as others, because their membership to R is weak. $\pi_{A|R}(\alpha|R)$, along with the choice of $\mathcal{R}$, is part of the definition of the similarity biases, and needs to be known a priori. If we extend $\pi_{A|R}$ to $R \setminus A$ by setting it to 0 for $\alpha \notin R$, then we can use $\pi_{A|R}$ as the very definition of R.

As in classification of web pages, we can obtain $Q_{Z|R}$ my minimizing the average KL-divergence. If the minimization is unconstrained, than the minimizing Q is the average over the conditionals of points in R. However, we may want to constrain Q to be a parametric distribution. If we do so, the similarity bias expresses both the fact that z should not vary

across R, and that the distribution of z should be approximated well by the parametric form of Q. The generic similarity bias of region R now becomes:

$$\min_{Q_{Z|R} \in \mathcal{M}_R} \int_{\alpha \in \mathcal{A}} \pi_{A|R}(\alpha|R) \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R)\right) d\alpha \tag{3.14}$$

$\mathcal{M}_R$ is a family of distributions that may restrict $Q_{Z|R}$. It may be unconstrained, or defined in terms of non-parametric marginal, or parametric constraints.

At the global level, we combine regularization penalties associated with local regions R into a global regularizer as a weighted average. The weights $\pi_R(R)$ form a task-specific probability distribution on $\mathcal{R}$ that must be given a priori:

$$I(P_{Z|A}) = \sum_{R \in \mathcal{R}} P_R(R) \min_{Q_{Z|R} \in \mathcal{M}_R} \int_{\alpha \in \mathcal{A}} \pi_{A|R}(\alpha|R) \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R)\right) d\alpha \tag{3.15}$$

It is useful to combine the a priori weights π_R and $\pi_{A|R}$ into a single joint distribution $\pi_{AR}(\alpha, R) = \pi_R(R)\pi_{A|R}(\alpha|R)$ that defines the structure of the information regularizer. We are now ready to provide a formal definition of the information regularizer:

Definition Let $\mathcal{A}$ be a set of points, and Z a random variable associated with each point with values from $\mathcal{Z}$. An *information regularizer* is a function that associates a non-negative number to the conditional density $P_{Z|A} \in \mathcal{F}$, defined in terms of the following items:

- a region set $\mathcal{R}$, where each region is a subset of $\mathcal{A}$
- a joint distribution on points and regions $\pi_{AR}(\alpha, R)$
- families $\mathcal{M}_R$ of distributions on $\mathcal{Z}$ associated with each region $R \in \mathcal{R}$

Than the information regularizer is given by:

$$I(P_{Z|A}) = \sum_{R \in \mathcal{R}} \min_{Q_{Z|R} \in \mathcal{M}_R} \int_{\alpha \in \mathcal{A}} \pi_{AR}(\alpha, R) \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R)\right) d\alpha \tag{3.16}$$

Note that λ, $\mathcal{R}$, π_{AR}, $\mathcal{F}$, and $\mathcal{M}_R$ are task-specific and must be known a priori, i.e. before seeing the training data. The selection of these parameters is beyond the scope of this thesis, though at times we will provide selection algorithms for experimental results.

Learning the parameters of information regularization is difficult because the rules for generating the regions are specific to the domain, not to the particular task instance. It is difficult to learn to characteristics of the domain from a single task, though not impossible.

3.3.1 Information regularization as a communication principle

We provide an information theoretical interpretation of the information regularization framework. We begin by rewriting the information regularizer in terms of mutual information. For this purpose, we view the random variables R, α, z as being sampled from a joint generative distribution with the following Markov dependency:

$$R \rightarrow \alpha \rightarrow z$$

The joint distribution is given by $P_{RAZ}(R, \alpha, z) = \pi_R(R)\pi_{A|R}(\alpha|R)P_{Z|A}(z|\alpha)$.

Theorem 1 *The information regularizer is equal to*

$$I(P_{Z|A}) = \sum_{R \in \mathcal{R}} \pi_R(R)\, \mathrm{I}\left(A|R;\, Z|R\right) + \sum_{R \in \mathcal{R}} \pi_R(R) \min_{Q_{Z|R} \in \mathcal{M}_R} \mathrm{KL}\left(P_{Z|R}(\cdot|R) \,\|\, Q_{Z|R}(\cdot|R)\right) \tag{3.17}$$

where $\mathrm{I}\left(A|R;\, Z|R\right)$ *is the mutual information between* A *and* Z *conditioned on* R*, and*

$$P_{Z|R}(z|R) = \int_{\alpha \in \mathcal{A}} P_{Z|A}(z|\alpha)\pi_{A|R}(\alpha|R)d\alpha \tag{3.18}$$

Corollary 2 *If all* $\mathcal{M}_R$*'s are unconstrained, then:*

$$I(P_{Z|A}) = \sum_{R \in \mathcal{R}} \pi_R(R)\, \mathrm{I}\left(A|R;\, Z|R\right) \tag{3.19}$$

In general, the right-hand side is a lower bound on the information regularizer.

Proof We manipulate the information regularizer as follows:

$$\begin{aligned} I(P_{Z|A}) &= \sum_{R\in\mathcal{R}} \pi_R(R) \min_{Q_{Z|R}\in\mathcal{M}_R} \int_{\alpha\in\mathcal{A}} \pi_{A|R}(\alpha|R) \int_{z\in\mathcal{Z}} P_{Z|A}(z|\alpha) \log \frac{P_{Z|A}(z|\alpha)}{Q_{Z|R}(z|R)} dz d\alpha \\ &= K + \sum_{R\in\mathcal{R}} \pi_R(R) \min_{Q_{Z|R}\in\mathcal{M}_R} \int_{z\in\mathcal{Z}} P_{Z|R}(z|R) \log \frac{P_{Z|R}(z|R)}{Q_{Z|R}(z|R)} dz \\ &= K + \sum_{R\in\mathcal{R}} \pi_R(R) \min_{Q_{Z|R}\in\mathcal{M}_R} \mathrm{KL}\left(P_{Z|R}(\cdot|R) \,\|\, Q_{Z|R}(\cdot|R)\right) \end{aligned} \tag{3.20}$$

where $P_{Z|R}(z|R) = \int_{\alpha\in\mathcal{A}} P_{Z|A}(z|\alpha)\pi_{A|R}(\alpha|R)d\alpha$. Here we denoted by K the term that does not depend on $Q_{Z|R}$, equal to:

$$K = \sum_{R\in\mathcal{R}} \pi_R(R) \int_{z\in\mathcal{Z}} \int_{\alpha\in\mathcal{A}} \pi_{A|R}(\alpha|R) P_{Z|A}(z|\alpha) \log \frac{P_{Z|A}(z|\alpha)}{P_{Z|R}(z|R)} dz d\alpha \tag{3.21}$$

The double integral in K from equation (3.21) is exactly the mutual information between A and Z given R [24].

If M_R's are unconstrained, the term in $I(P_{Z|X})$ that depends on $Q_{Z|R}$ is equal to 0, because the KL-divergences vanish when $Q_{Z|R} = P_{Z|R}$ (provided that $P_{Z|R} \in \mathcal{M}_R$, for all $R \in \mathcal{R}$). Thus $I(P_{Z|A}) = K$ when M_R's are unconstrained. □

We formulate the following rate distortion with side information communication problem. Consider a data source that generates (α, R) according to $\pi_{AR}(\alpha, R)$. We would like to transmit a lossy version of α, which we denote by $z \in \mathcal{Z}$, across a channel by sending a minimal number of bits, such that z is still an accurate representation of α. Hence we limit the *distortion* between the output z and the input α. Our measure of distortion is the supervised learning loss function, on which we place an upper bound M:

$$\mathrm{loss}_{\mathcal{D}}(P_{Z|A}) \leq M \tag{3.22}$$

The goal is thus to produce a noisy channel $P_{Z|A}$ such that the rate of information that needs to be transmitted in order to preserve $\mathrm{loss}_{\mathcal{D}}(P_{Z|A}) \leq M$ is minimal. As opposed to standard rate distortion theory, the receiver will also have access to the side information R when decoding z.

According to rate distortion theory, the distribution $P_{Z|A}$ that minimizes the bit rate that needs to be transmitted can be found by minimizing the mutual information between

A and Z subject to the distortion constraint. In our case we also have access to the side information R, therefore we must minimize the average mutual information restricted to each region R:

$$\arg\min_{P_{Z|A} \text{ s.t } \text{loss}_{\mathcal{D}}(P_{Z|A}) \leq M} \sum_{R \in \mathcal{R}} P_R(R) \mathrm{I}\left(A|R;\, Z|R\right) \tag{3.23}$$

If we replace the constraint with a Lagrange multiplier, the objective reduces to:

$$\arg\min_{P_{Z|A}} \ \text{loss}_{\mathcal{D}}(P_{Z|A}) + \lambda \sum_{R \in \mathcal{R}} P_R(R) \mathrm{I}\left(A|R;\, Z|R\right) \tag{3.24}$$

where M is now defined implicitly through λ. This objective is the same as that of information regularization in the case in which $\mathcal{M}_R$'s are unconstrained.

Relationship to *Information Bottleneck*

Information Bottleneck [57] is a popular clustering method with strong connections to rate distortion theory. We highlight the similarities and differences between the information bottleneck method and information regularization.

In Information Bottleneck the goal is to compress a random variable A into a random variable Z, where the number of symbols in $\mathcal{Z}$ is known in advance, and is smaller than the cardinality of $\mathcal{A}$. We represent the compression as a probabilistic mapping $P_{Z|A}(z|\alpha)$ that needs to be determined. We can measure the degree of compression by the mutual information between A and Z:

$$\mathrm{I}\left(A;\, Z\right) = \sum_{\alpha \in \mathcal{A}} \sum_{z \in \mathcal{Z}} P_{AZ}(\alpha, z) \log \frac{P_{AZ}(\alpha, z)}{P_A(\alpha) P_Z(z)} \tag{3.25}$$

The smaller the mutual information, the better the compression, at the expense of discarding information contained in A; therefore there is a trade-off between the achievable compression factor and the information about A that must be retained. We express this trade-off by the means of an auxiliary random variable R correlated with A that contains the relevant information about A that needs to be retained. We may think of R as a quantity that needs to be predicted given the value of A as input. We would like to compress A, while preserving its ability to predict R.

The information bottleneck method compresses A into Z by minimizing the following objective:

$$\min_{P_{Z|A}} \mathrm{I}(A;\, Z) - \beta \mathrm{I}(R;\, Z) \tag{3.26}$$

Thus Z must contain as little information about A as possible (maximum compression), while retaining as much relevant information as possible.

How does the information bottleneck method relate to information regularization? Let us take a closer look at the information regularizer for unrestrictive $\mathcal{M}_R$:

$$\begin{aligned} &I(P_{Z|A}) = \\ &\quad \sum_{R \in \mathcal{R}} \pi_R(R) \mathrm{I}(A|R;\, Z|R) = H(Z|R) - H(Z|A,R) = \\ &\quad H(Z|R) - H(Z|A) = (H(Z) - H(Z|A)) - (H(Z) - H(Z|R)) = \\ &\quad \mathrm{I}(A;\, Z) - \mathrm{I}(R;\, Z) \end{aligned} \tag{3.27}$$

Thus for unrestricted $\mathcal{M}_R$ the information regularizer is a special instance of the information bottleneck objective with $\beta = 1$. In other words, information regularization uses a special form of the information bottleneck objective as a regularizer applied to a standard loss function. Unlike in generic information bottleneck, the special form with $\beta = 1$ ensures convexity. Nevertheless, the information regularization framework does depart from information bottleneck in the case in which $\mathcal{M}_R$ is a restricted family.

3.3.2 Information regularization and convexity

While the information regularization framework is quite expressive as formulated, care must be taken to produce a tractable objective. In particular, the constrained families $\mathcal{F}$ and $\mathcal{M}_R$ may introduce non-convexity, making the optimization complex. In the subsequent chapters we will introduce tractable algorithms for various instances of information regularization. For now we state a generic result that is valid when the sets $\mathcal{M}_R$ are unconstrained. In this situation the information regularizer takes the form presented in equation (3.17).

Theorem 3 *If $\mathcal{M}_R$ is unconstrained for all $R \in \mathcal{R}$, then the information regularizer is a convex function of $P_{Z|A}$.*

Proof Suppose $P_{Z|A}$ is equal to a convex combination $(1-\epsilon)P^1_{Z|A}+\epsilon P^2_{Z|A}$ of conditionals, such that $\epsilon \in (0,1)$ and P^1 and P^2 differ on a subset of $\mathcal{Z} \times \mathcal{A}$ of non-zero measure (with respect to a measure whose support is $\mathcal{Z} \times \mathcal{A}$). It follows immediately that $P_{Z|R}(z|R) = \int_{\alpha \in \mathcal{A}} P_{Z|A}(z|\alpha)\pi_{A|R}(\alpha|R)d\alpha$ satisfies the same convex combination:

$$P_{Z|R}(\cdot|R) = (1-\epsilon)P^1_{Z|R}(\cdot|R) + \epsilon P^2_{Z|R}(\cdot|R) \tag{3.28}$$

Since KL-divergence is convex [24] it follows that

$$\begin{aligned} \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, P_{Z|R}(\cdot|R)\right) <& (1-\epsilon)\mathrm{KL}\left(P^1_{Z|A}(\cdot|\alpha) \,\|\, P^1_{Z|R}(\cdot|R)\right) + \\ & \epsilon \mathrm{KL}\left(P^2_{Z|A}(\cdot|\alpha) \,\|\, P^2_{Z|R}(\cdot|R)\right) \end{aligned} \tag{3.29}$$

Applying this inequality to equation (3.16) yields the convexity of the information regularizer. □

3.4 Semi-supervised principles subsumed by information regularization

We illustrate that various settings of the information regularizer yield a broad range of existing semi-supervised principles. The power of information regularization is that it can not only reproduce these principles, but also combine them as appropriate. Please refer to Chapter 2 for a more detailed explanation of the principles.

3.4.1 Parametric joint

Scenario

In this setting we assume that the joint over features and labels belongs to a parametric family

$$\{P_{XY}(\mathbf{x}, y; \boldsymbol{\theta}) \,;\, \boldsymbol{\theta} \in \Theta\}$$

The goal is to estimate $\boldsymbol{\theta}$ given a finite training set $\mathcal{D} \subset \mathcal{A}$ that consists of both labeled and unlabeled data. The standard approach is to maximize the log likelihood of $\mathcal{D}$ where the

label of unlabeled samples is treated as a latent variable:

$$\sum_{i=1}^{l} \log P_{XY}(\mathbf{x}_{\alpha_i}, y_{\alpha_i}\,;\,\boldsymbol{\theta}) + \sum_{j=l+1}^{n} \log P_X(\mathbf{x}_{\alpha_j}\,;\,\boldsymbol{\theta}) \tag{3.30}$$

The information regularizer

Let the variable of interest z of information regularization be the feature and label pair $(\mathbf{x}, y)$ associated with a point. We define an information regularizer with the following structure:

- $\mathcal{R} = \{\mathcal{D}\}$ (a single region containing all training points, labeled and unlabeled)
- $\pi_{AR}(\alpha, \mathcal{D}) = 1/n$ if $\alpha \in \mathcal{D}$, 0 otherwise (n is the cardinality of $\mathcal{D}$)
- $\mathcal{M}_{\mathcal{D}} = \{P_{XY}(\mathbf{x}, y; \boldsymbol{\theta})\,;\,\boldsymbol{\theta} \in \Theta\}$ (the distribution of the regions is restricted to our parametric family)
- $\mathcal{F}_\sigma = \{P_{Z|A}(\mathbf{x}, y|\alpha) = P_{Y|X}(y|\mathbf{x})K(\mathbf{x}|\alpha, \sigma)\}$, where $K(\mathbf{x}|\alpha, \sigma)$ is a Gaussian kernel of mean α and covariance $\sigma^2 I$. In other words, only look for $P_{Z|A}(z|\alpha)$ of the form $P_{Y|X}(y|\mathbf{x})$, where $\mathbf{x}$ is the feature representation of α. The parameter σ must be given, but can be moved to 0, as the reader will see.

According to equation (3.16) the information regularizer takes the following form:

$$\begin{aligned}
I(P_{XY|A}) &= \min_{Q_{Z|R} \in \mathcal{M}_{\mathcal{D}}} \sum_{\alpha \in \mathcal{D}} \frac{1}{n} \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|\mathcal{D})\right) \\
&= \min_{\boldsymbol{\theta} \in \Theta} \sum_{\alpha \in \mathcal{D}} \frac{1}{n} \int_{\mathbf{x} \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P_{Y|X}(y|\mathbf{x}) K(\mathbf{x}|\alpha, \sigma) \log \frac{P_{Y|X}(y|\mathbf{x}) K(\mathbf{x}|\alpha, \sigma)}{Q_{Z|R}(\mathbf{x}, y; \boldsymbol{\theta}|\mathcal{D})} d\mathbf{x} dy \\
&= \frac{1}{n} \min_{\boldsymbol{\theta} \in \Theta} \left[-\sum_{\alpha \in \mathcal{D}} \int_{\mathbf{x} \in \mathcal{X}} K(\mathbf{x}|\alpha, \sigma) \log Q_{Z|R}(\mathbf{x}\,;\,\boldsymbol{\theta}|\mathcal{D}) d\mathbf{x} + \right. \\
&\quad + \sum_{\alpha \in \mathcal{D}} \int_{\mathbf{x} \in \mathcal{X}} K(\mathbf{x}|\alpha, \sigma) \mathrm{KL}\left(P_{Y|X}(\cdot|\mathbf{x}) \,\|\, Q_{Z|R}(\cdot|\mathbf{x}\,;\,\boldsymbol{\theta}|\mathcal{D})\right) d\mathbf{x} - \\
&\quad \left. \sum_{\alpha \in \mathcal{D}} H(K(\cdot|\alpha, \sigma)) \right]
\end{aligned} \tag{3.31}$$

Let us examine the three terms that form the information regularizer as illustrated in the above equation in a situation in which all data points are unlabeled. In this situation we apply the regularizer by simply minimizing it, because there is no labeled evidence:

$$\min_{P_{Y|X}} I(P_{XY}|A) \tag{3.32}$$

The last term in the information regularizer, the entropy of the Gaussian kernels, can be ignored, because it depends on neither $P_{Y|X}$ nor $\boldsymbol{\theta}$. The term in the middle vanishes when minimizing over $P_{Y|X}$, because $P_{Y|X}$ is unconstrained and can be made equal to $Q_{Z|R}(\cdot|\mathbf{x}\,;\,\boldsymbol{\theta}|R)$. Thus information regularization in this case is equivalent to the following estimation:

$$\min_{\boldsymbol{\theta}\in\Theta} -\sum_{\alpha\in\mathcal{D}} \int_{\mathbf{x}\in\mathcal{X}} K(\mathbf{x}|\alpha,\sigma) \log Q_{Z|R}(\mathbf{x}\,;\,\boldsymbol{\theta}|R) d\mathbf{x} \tag{3.33}$$

When σ is very large ($\sigma \to \infty$), this objective is exactly maximization of the incomplete data log-likelihood of the parametric family:

$$\max_{\boldsymbol{\theta}\in\Theta} \sum_{\alpha\in\mathcal{D}} \log Q_{Z|R}(\mathbf{x}_\alpha\,;\,\boldsymbol{\theta}|R) \tag{3.34}$$

If the training set $\mathcal{D}$ contains also some labeled points, it is now easy to see that information regularization would be similar to the standard objective in equation (3.30), except that there will be a weighting between the labeled and unlabeled parts as dictated by λ.

3.4.2 Redundantly sufficient features

Scenario

In this setting the feature representation $\mathbf{x}$ of each data point consists of multiple features $\mathbf{x} = (f_1, f_2, \ldots, f_k)$ that are redundant in the sense that each component is sufficient for building a noisy classifier. This redundancy can be the basis of a semi-supervised principle: an unlabeled data point labeled confidently by one classifier can be used to correct other classifiers that are uncertain about their label. One popular instance of this semi-supervised principle is *co-training*, in which samples labeled confidently by one classifier will be used as training points for another classifier in the next iteration of the algorithm.

We consider a scenario in which each of the k classifiers is parametric. Therefore we define classifier i by a restricted family of probability distributions on f_i:

$$\mathcal{M}_{R_i} = \{P_{YF_i}(y, f_i\,;\,\boldsymbol{\theta})\,;\,\boldsymbol{\theta} \in \Theta_{R_i}\} \tag{3.35}$$

We have one such family for each $1 \leq i \leq k$. Without loss of generality we can assume that $\mathcal{M}_{R_i}$ is a family of distributions on $\mathcal{X} \times \mathcal{Y}$ that constrains (f_i, y) to a parametric form, and leaves the joint on the rest of the features unconstrained.

The information regularizer

We show that information regularization can incorporate redundantly sufficient features. We build upon the result from the previous section in which information regularization with a single parametric region can mimic a parametric classifier. Since here we have k classifiers, we cover the data with k regions. Each region contains the entire training set $\mathcal{D}$, thus the regions differ only in terms of their parametric restriction.

As before, the variable of interest z is $(\mathbf{x}, y)$. The information regularizer has the following structure:

- $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$
- $\pi_{AR}(\alpha, R_i) = 1/(nk)$ if $\alpha \in \mathcal{D}$, 0 otherwise (n is the cardinality of $\mathcal{D}$). This must be true for all $1 \leq i \leq k$.
- the distribution family associated with region R_i is M_{R_i} as defined in equation (3.35)
- As in the previous scenario, we relate $P_{Z|A}(z|\alpha)$ to $P_{Y|X}(y|\mathbf{x})$ by the means of a Gaussian kernel: $\mathcal{F}_\sigma = \{P_{Z|A}(\mathbf{x}, y|\alpha) = P_{Y|X}(y|\mathbf{x})K(\mathbf{x}|\alpha, \sigma)\}$

The information regularizer that must be minimized is a sum of the information regularizers representing each classifier:

$$I(P_{XY|A}) = \frac{1}{kn}\sum_{i=1}^{K} \min_{Q_{Z|R}(\cdot|R_i)\in\mathcal{M}_{R_i}} \sum_{\alpha\in\mathcal{D}} \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R_i)\right) \tag{3.36}$$

As we have seen in the previous section, minimizing the information regularizer of one of the regions R_i is equivalent to estimating the parameters of the classifier i by maximum

likelihood in a semi-supervised fashion. When we bring together the information regularizers of each region, the net effect is that samples labeled confidently by one classifier will contribute to the training of the other classifiers. The rule that governs the resolution of the posteriors of each classifier $Q_{Z|R}(y|\mathbf{x}\,;\,\theta_i|R_i)$ into a single posterior $P_{Y|X}(y|\mathbf{x})$ will be apparent in Chapter 5 when we discuss optimization of information regularization in detail.

3.4.3 Label similarity bias

We show that the information regularization framework can model the objective of graph regularization semi-supervised methods [61]. The semi-supervised principle underlying these methods is that points that are similar according to some metric or some rule are biased to have similar labels.

Scenario

Consider a binary classification task ($\mathcal{Y} = \{-1, 1\}$), and suppose that we are only interested in the labels of the unlabeled points that we received as training data. Therefore $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, and $\mathcal{D} = \mathcal{A}$. Suppose that we are also given an undirected graph with $\mathcal{A}$ as vertexes, and a set of edges $(i, j) \in E$ and associated positive weights w_{ij}. The underlying assumption is that the magnitude of w_{ij} reflects the degree to which the labels y_{α_i} and y_{α_j} are constrained to be similar.

Let z_α be a real number that expresses the confidence in point α having label 1. The goal is to estimate z_α for every $\alpha \in \mathcal{A}$. We define a simple graph regularization method for semi-supervised learning by penalizing

$$\sum_{(i,j)\in E} w_{ij}(z_{\alpha_i} - z_{\alpha_j})^2 \tag{3.37}$$

Therefore, given the labels of the first l data points in $\mathcal{D}$, we can assign labels to all points in $\mathcal{A}$ by minimizing the following regularized loss:

$$\frac{1}{l}\sum_{i=1}^{l}(y_{\alpha_i} - z_{\alpha_i})^2 + \lambda \sum_{(i,j)\in E} w_{ij}(z_{\alpha_i} - z_{\alpha_j})^2 \tag{3.38}$$

In what follows we show that information regularization can naturally model this standard graph regularization objective.

The information regularizer

We define an information regularizer that has the same structure as the undirected graph in the sense that we consider a region for every edge in the graph, that contains its end points. The information regularizer will estimate for each point a soft label that is a Gaussian random variable with z_α its mean. We need to show that the information regularization objective and the graph regularization objective are identical.

We define the following information regularizer:

- $\mathcal{R} = \{R_{ij}\,;\,(i,j) \in E\}$, where $R_{ij} = \{\alpha_i, \alpha_j\}$
- $\pi_{AR}(\alpha, R_{ij}) = w_{ij}/(2w_{tot})$ if $\alpha \in \{\alpha_i, \alpha_j\}$, 0 otherwise ($w_{tot}$ is the total weight of all edges)
- $\mathcal{M}_{R_{ij}}$ is the family of one-dimensional Gaussian random variables of unit variance. The family is parametrized by θ_{ij}, the mean of the Gaussian
- $\mathcal{F}$ is left unconstrained

Then according to the information regularization framework we can estimate $P_{Z|A}(z|\alpha)$ by minimizing:

$$\begin{aligned}\min_{P_{Z|A}} \frac{1}{l}\sum_{i=1}^{l} \text{loss}(y_{\alpha_i}, P_{Z|A}(\cdot|\alpha_i)) + \\ \lambda \sum_{(i,j)\in E} \frac{w_{ij}}{2w_{tot}} \min_{\theta_{ij}} \left[\text{KL}\left(P_{Z|A}(\cdot|\alpha_i) \,\|\, Q_{Z|R}(\cdot\,;\,\theta_{ij}|R_{ij})\right) + \right. \\ \left. \text{KL}\left(P_{Z|A}(\cdot|\alpha_j) \,\|\, Q_{Z|R}(\cdot\,;\,\theta_{ij}|R_{ij})\right)\right]\end{aligned} \tag{3.39}$$

Assume that the loss is a KL-divergence between $P_{Z|A}(\cdot|\alpha_i)$ and a Gaussian of unit variance and mean equal to y_{α_i}:

$$\text{loss}(y_{\alpha_i}, P_{Z|A}(\cdot|\alpha_i)) = \text{KL}\left(P_{Z|A}(\cdot|\alpha_i) \,\|\, Q(\cdot\,;\,y_{\alpha_i})\right) \tag{3.40}$$

We show that minimizing the objective defined above is equivalent to minimizing the graph regularization objective in equation (3.38).

We begin by proving that the optimal $P_{Z|A}(\cdot|\alpha)$ will necessarily be a Gaussian distribution of unit variance. We will need the following lemma:

Lemma 4 *Let p, $q_1, q_2, \ldots, q_k$ be probability measures on a common space. If $\sum_{i=1}^{k} \lambda_i = 1$, where $\lambda_i \geq 0$, then we have:*

$$\sum_{i=1}^{k} \lambda_i \mathrm{KL}\left(p \,\|\, q_i\right) = \log \tilde{K} + \mathrm{KL}\left(p \,\|\, \tilde{K} \prod_{i=1}^{k} q_i^{\lambda_i}\right) \tag{3.41}$$

where $\tilde{K}$ is the normalization that ensures that the distribution in the second term of the KL-divergence integrates to 1.

Proof The proof is only a matter of verifying the identity. □

$P_{Z|A}(\cdot|\alpha_i)$ must achieve the minimum value of the following objective, for some value of the parameters θ_{ij}:

$$\tau_i \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha_i) \,\|\, Q(\cdot\,;\, y_{\alpha_i})\right) + \lambda \sum_{j,\ \text{s.t. } (i,j)\in E} w_{ij} \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha_i) \,\|\, Q_{Z|R}(\cdot\,;\, \theta_{ij}|R_{ij})\right) \tag{3.42}$$

where τ_i is 1 for labeled samples, 0 otherwise.

According to Lemma 4 this may be written as a single KL-divergence. Since $P_{Z|A}$ is unconstrained, if the objective is optimal then $P_{Z|A}(\cdot|\alpha_i)$ is equal to the second argument of the KL-divergence, and the KL-divergence is 0. The second argument is a geometric average of Gaussians of unit variance, which is also a Gaussian of unit variance. This completes the proof that the optimal $P_{Z|A}(\cdot|\alpha_i)$ will be Gaussian of unit variance for every α_i.

Let z_{α_i} be the mean of the Gaussian of unit variance $P_{Z|A}(\cdot|\alpha_i)$. We can now rewrite the information regularization objective in equation (3.39) only in terms of z_α's and θ_{ij}'s. For this purpose we need the following known result:

Lemma 5 *If p and q are Gaussians of unit variance of means μ and τ then:*

$$\mathrm{KL}\left(p \,\|\, q\right) = \frac{1}{2}(\mu - \tau)^2 \tag{3.43}$$

Proof This is only a matter of verifying the identity, given that $p(x) = \exp(-(x - \mu)^2/2)/\sqrt{2\pi})$ and $q(x) = \exp(-(x-\tau)^2/2)/\sqrt{2\pi})$. □

Using the above lemma, equation (3.38) can now be written as:

$$\min_{z_\alpha, \alpha \in \mathcal{A}} \frac{1}{l} \sum_{i=1}^{l} (z_{\alpha_i} - y_i)^2 + \lambda \sum_{(i,j) \in E} w_{ij} \min_{\theta_{ij}} \left[(z_{\alpha_i} - \theta_{ij})^2 + (z_{\alpha_j} - \theta_{ij})^2 \right] \tag{3.44}$$

The minimum over θ_{ij} is achieved when $\theta_{ij} = (z_{\alpha_i} + z_{\alpha_j})/2$. Thus the information regularization method minimizes the following objective:

$$\min_{z_\alpha, \alpha \in \mathcal{A}} \frac{1}{l} \sum_{i=1}^{l} (z_{\alpha_i} - y_i)^2 + 2\lambda \sum_{(i,j) \in E} \frac{w_{ij}}{2w_{tot}} (z_{\alpha_i} - z_{\alpha_j})^2 \tag{3.45}$$

This is the same objective as in semi-supervised graph regularization (3.38), with a slight adjustment in λ. We conclude that the information regularization framework subsumes this semi-supervised principle.

3.4.4 Data-dependent smoothness prior and low-density separation

Scenario

In this scenario $\mathcal{X}$ is an Euclidean space, and we know a priori that $P_{Y|X}(y|\mathbf{x})$ must vary smoothly as a function $\mathbf{x}$ in a manner that depends on the data density $P_X(\mathbf{x})$; the higher the data density, the less likely is that we see large variations in $P_{Y|X}(y|\mathbf{x})$. As a consequence, we prefer decision boundaries that do not cross regions of high density. We show that the information regularizer can act as a smoothness prior.

The information regularizer

Assume that $\mathcal{A} = \mathcal{X}$, thus we can use α and $\mathbf{x}_\alpha$ interchangeably. We define an information regularizer on the variable of interest y based on a uniform covering $\mathcal{R}$ of the space $\mathcal{X}$ with small overlapping cubes of equal size. Their centers can be for example lattice points, where the distance between consecutive lattice points is small, as in [53, 21]. Suppose that we cover the space in such a way that every point $\mathbf{x} \in \mathcal{X}$ belongs to exactly T regions from $\mathcal{R}$. Then we can define the following distribution over regions:

$$\pi_{AR}(\alpha, R) = \begin{cases} \frac{1}{T} P_X^\star(\mathbf{x}_\alpha) & \text{if} \quad \mathbf{x}_\alpha \in R \\ 0 & \text{if} \quad \mathbf{x}_\alpha \notin R \end{cases} \tag{3.46}$$

where $P^\star_X(\mathbf{x})$ is the data density, or estimate of it based on available labeled samples.

The definition of $\pi_{AR}(\alpha, R)$ naturally emerges from the following generative process: generate samples $\mathbf{x}$ from $\mathcal{X}$ according to $P^\star_X$, then generate a region R from the T regions containing $\mathbf{x}$ uniformly.

We complete the specification of the information regularizer by mentioning that $\mathcal{F}$ and $\mathcal{M}_R$ for $R \in \mathcal{R}$ are left unconstrained. Then according to equation (3.17) the information regularizer takes the following form:

$$I(P_{Y|X}) = \sum_{R \in \mathcal{R}} P^\star_X(R)\, \mathrm{I}\,(X|R;\, Y|R) \tag{3.47}$$

Note that our definition of $P_{XR}(\mathbf{x}, R)$ ensures that the local constraint expressed by the regularizer of each region is weighted by the data density of the region $P^\star_X(R) = \int_{\mathbf{x} \in R} P^\star_X(\mathbf{x}) d\mathbf{x}$. The net effect is that the variation in $P_{Y|X}$ is penalized more in regions of high data density than in regions of low data density. Thus the information regularizer is equivalent to a data-dependent smoothness prior.

3.5 A taxonomy of information regularization algorithms

In order to apply the information regularization framework to a specific task we still need to provide an algorithm for optimizing the objective. We distinguish between various possible algorithms across a few dimensions, as depicted in Figure 3.5.

metric vs. relational

This distinction is between algorithms that assume that the similarity between labels is based on a metric defined on $\mathcal{A}$, or based on relations between the samples that come from additional sources of information.

full marginal vs. finite sample

Some information regularization algorithms assume that unlabeled data is abundant, and we can estimate $P_A(\alpha)$ precisely. Then only $P_{R|A}(R|\alpha)$ needs to be specified a priori. Other algorithms assume instead that only a finite number of unlabeled samples are available. It is

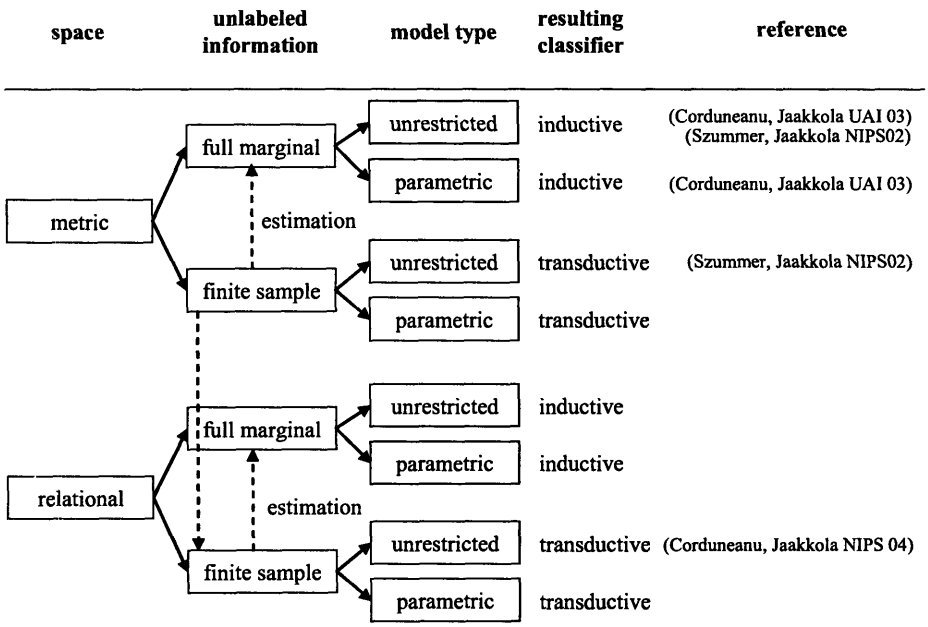

Figure 3-1: Types of information regularization algorithms

always possible to view a "full-marginal" algorithm as a "finite sample" one by providing a kernel estimate of $P_A(\alpha)$ from the finite unlabeled sample.

parametric vs. unrestricted

This category distinguished between applications of information regularization where $\mathcal{F}$ and $\mathcal{M}_R$'s are parametric, or are completely unrestricted.

transductive vs. inductive

It is possible to provide information regularization algorithms that are either *transductive* or *inductive*. Transductive algorithms are aimed at estimating the labels of the received unlabeled points, where inductive algorithms can estimate the label of any point in the space, indifferent of whether it has been received as an unlabeled sample or not.

We will see that information regularization is never strictly transductive. It will always be possible to estimate the labels of other points in the space provided that we can determine the regions from $\mathcal{R}$ to which they belong.

Chapter 4

Information regularization on metric spaces

In this chapter we consider semi-supervised learning on continuous spaces, endowed with a metric correlated with the labeling of the points. In other words, the assumption is that neighboring points with respect to the metric are a priori more likely to have similar labels than distant points. Also, the label is more likely to change in regions of low data density. A common example would be a situation in which the features are vectors in an Euclidean space with the standard Euclidean metric, possibly weighted by the relevance of individual components of the feature vectors. The analysis in this chapter applies to a wide variety of learning tasks in which the feature vectors have continuous components, as long as a suitable metric exists. In what follows we adapt the information regularizer to the continuous setting, discuss specific theoretical results, and derive explicit optimization algorithms.

The distinguishing characteristic of information regularization on continuous spaces is that there is a continuum of regions defining the semi-supervised bias; indeed, we can place a local prior on label similarity centered at every point in the space. Therefore it is not possible to work directly with the standard regularizer defined in the previous chapter, and here we derive the regularizer as the limit of finite discretizations, as in [21]. Note that if we were to limit ourselves to a transductive algorithm, where we are only interested in the labels of a finite set of observed unlabeled points, we could still define an information regularizer that needs only a finite number of regions. The following approach is an induc-

tive algorithm, while the discrete graph regularization that we will introduce in Chapter 5 is transductive.

In this continuous and inductive setting, we can identify each point α with its feature representation $\mathbf{x}_\alpha$. This is because we will derive all regions of information regularization from the metric applied to the vector $\mathbf{x}_\alpha$, and there is no other information that we know about the points besides $\mathbf{x}_\alpha$. Consequently, we identify the $\mathcal{X}$ in the generic formulation of information regularization with $\mathcal{X}$.

4.1 Full knowledge of the marginal

Let us begin by considering the ideal situation in which we have access to unlimited unlabeled data, which is equivalent to knowing the marginal density P_X, an assumption that we will relax later on. The goal is to convert our knowledge of P_X into a priori biases about how the data should be labeled, such that the label is less likely to change in regions of low data density than in high-density regions. The information regularizer that we construct is a penalty whose minimization constrains variations in the label to regions of low data density, without making any parametric assumptions about the underlying data distribution.

4.1.1 The information regularizer

The basic building block of the regularization penalty consists of a region $R \subset \mathcal{X}$ on which we impose the bias that points belonging to the region have similar labels $P_{Y|X}$. In our continuous metric setting, the region is a collection of points that are close to each other with respect to the metric, for instance, a sphere of small diameter. As we have seen in the previous chapter in (equation (3.19)), in a non-parametric setting we can quantify the similarity of the labels of points belonging to a region by the mutual information between A and Y. Specifically, if $\mathcal{R}$ is our collection of regions (for now finite), we can define the following information regularizer:

$$I(P_{Y|X}) = \sum_{R \in \mathcal{R}} \pi_R(R)\, \mathrm{I}\,(X|R;\, Y|R) \tag{4.1}$$

where

$$\mathrm{I}(X|R;\, Y|R) = \int_{\mathbf{x}\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \pi_{X|R}(\mathbf{x}|R) P_{Y|X}(y|\mathbf{x}) \log \frac{P_{Y|X}(y|\mathbf{x})}{P_{Y|R}(y|R)} d\mathbf{x} \tag{4.2}$$

and

$$P_{Y|R}(y|R) = \int_{\mathbf{x}\in\mathcal{X}} \pi_{X|R}(\mathbf{x}|R) P_{Y|X}(y|\mathbf{x}) d\mathbf{x} \tag{4.3}$$

Let us discuss the choice of the weights $\pi_R(R)$ and $\pi_{A|R}(\alpha|R)$ in the information regularizer. In order to capture the low-density separation principle, variations in dense regions must be penalized more. Thus an appropriate choice for $\pi_R(R)$ is to be proportional to the cumulative probability mass in region R:

$$\pi_R(R) \propto P_A(R) = \int_{\alpha\in R} P_A(\alpha) d\alpha \tag{4.4}$$

Next, we choose $\pi_{X|R}(\mathbf{x}|R)$ such that the generative process of choosing R according to π_R, then $\mathbf{x}$ according to $\pi_{X|R}$, results in generating $\mathbf{x}$ according to P_X. It follows that

$$\pi_{X|R}(\mathbf{x}|R) = \begin{cases} 0, & \text{if } \mathbf{x} \notin R \\ P_X(\mathbf{x})/P_X(R), & \text{if } \mathbf{x} \in R \end{cases} \tag{4.5}$$

It remains to choose the regions. Ideally, $\mathcal{R}$ would be a uniform covering of $\mathcal{X}$ with identical regions centered at every point in the space. Since the generic information regularizer would be intractable on an infinite set of regions, our approach is to discretize $\mathcal{R}$, then take the limit as the number of regions converges to infinity. The limiting form has the additional benefit that it no longer requires us to engineer a particular covering of the space.

Sensible regions must have the following properties:

- The regions are small.
- The *overlap* between neighboring regions is significant.
- The shape and distribution of the regions avoids *systematic biases*.

Small regions ensure that the semi-supervised bias of label similarity remains a local property. Overlapping regions ensure that the information regularization functions as a global criterion, with the local semi-supervised bias imposed by each region being propagated to

its neighbors. Needless to say, the information regularizer should not introduce systematic biases that cannot be justified a priori (such as a preference for allowing variations of $P_{Y|X}$ only in a certain direction). For example, to avoid biases we make all regions of the same shape, centered at different points.

Infinitesimal Information Regularizer

While increasing the number of regions to infinity, we also decrease their size to 0, and increase the overlap. The size of the regions is akin to the *resolution* at which we can measure the variation in labels, and decreasing the size of the regions ensures that in the limit we have infinite resolution.

We identify two tendencies in the limit. On the one hand the local mutual information will converge to 0 as the diameter of the region approaches 0; this is normal, as $P_{Y|X}$ will look more constant the smaller the region. On the other hand, as the overlap between regions increases, we get a multiplicative effect from the fact that each point belongs to more and more regions. In the limit, this multiplicative factor is infinity. Thus if the regions do not overlap enough, the regularizer will converge to 0, and if they overlap too much, it will converge to infinity. In order to produce a finite infinitesimal information regularizer we must strike the right balance between the size of the regions and their overlap.

We begin by assessing the asymptotics of the local mutual information as the diameter of the region R converges to 0. We have the following result [21]:

Theorem 6 *Let* $\mathrm{I}(X|R;\, Y|R)$ *be the mutual information restricted to* R *as defined in (4.2). If* $\operatorname{diam}(R)$ *is the diameter of region* R*, then the mutual information takes the following asymptotic form with respect to the diameter:*

$$\mathrm{I}(X|R;\, Y|R) = \frac{1}{2}\mathrm{Tr}\left[\operatorname{cov}_{\pi_{X|R}}(X)\, F(\mathrm{E}_{\pi_{X|R}}[X])\right] + \mathcal{O}\left(\operatorname{diam}(R)^3\right) \tag{4.6}$$

Here $\mathrm{E}_{\pi_{X|R}}[X]$ *is the expected value of the vector* $\mathbf{x}_\alpha$*, and* $\operatorname{cov}_{\pi_{X|R}}(X)$ *its covariance, when* α *is distributed according to* $\pi_{A|R}$*. Also,* $F(\mathbf{x})$ *is the Fisher information matrix of the distribution* $P_{Y|X}$ *evaluated at* $\mathbf{x}$*:*

$$\mathrm{E}_{P_{Y|X}(y|\mathbf{x})}\left[\nabla_{\mathbf{x}} \log P_{Y|X}(y|\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log P_{Y|X}(y|\mathbf{x})^{\top}\right] \tag{4.7}$$

Moreover, $\mathrm{cov}_{\pi_{X|R}}(X)$ *is* $\mathcal{O}(\mathrm{diam}(R)^2)$.

Proof Let $\mathbf{x}_0 = \mathrm{E}_{\pi_{X|R}}[X]$ be the average value of $\mathbf{x}$ in the region. To simplify notation let $G = \nabla_{\mathbf{x}} P_{Y|X}(y|\mathbf{x}_0)$ and $H = \nabla^2_{\mathbf{xx}} P_{Y|X}(y|\mathbf{x}_0)$ be the gradient and the Hessian of the conditional at $\mathbf{x}_0$. The conditional has the following second order Taylor expansion about $\mathbf{x}_0$:

$$P_{Y|X}(y|\mathbf{x}) = P_{Y|X}(y|\mathbf{x}_0) + G^\top(\mathbf{x}-\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0)^\top H(\mathbf{x}-\mathbf{x}_0) + \mathcal{O}\left(\mathrm{diam}(R^3)\right) \quad (4.8)$$

By taking expectation with respect to $\pi_{X|R}(\mathbf{x}|R)$, and using the definition in equation (4.3), we get

$$P_{Y|R}(y|R) = P_{Y|X}(y|\mathbf{x}_0) + \mathrm{Tr}\left[\mathrm{cov}_{\pi_{X|R}}(X)\, H\right] + \mathcal{O}\left(\mathrm{diam}(R^3)\right) \quad (4.9)$$

Next we use $1/(1+x) = 1 - x + x^2 + \mathcal{O}(x^3)$ and $\log(1+x) = x - x^2/2 + \mathcal{O}(x^3)$ to obtain:

$$\begin{aligned}\log \frac{P_{Y|X}(y|\mathbf{x})}{P_{Y|R}(y|R)} =& \frac{1}{P_{Y|X}(y|\mathbf{x}_0)}[G^\top(\mathbf{x}-\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0)^\top H(\mathbf{x}-\mathbf{x}_0) - \\ & \mathrm{Tr}\left[\mathrm{cov}_{\pi_{X|R}}(X)\, H\right] - [G^\top(\mathbf{x}-\mathbf{x}_0)]^2/2P_{Y|X}(y|\mathbf{x}_0)] + \mathcal{O}\left(\mathrm{diam}(R^3)\right)\end{aligned} \quad (4.10)$$

We only need to multiply the above equation by the expansion of $P_{Y|X}(y|\mathbf{x})$ again and take the expectation with respect to $\pi_{X|R}(\mathbf{x}|R)$ to get:

$$\mathrm{I}(X|R;\, Y|R) = \sum_{y\in\mathcal{Y}} \frac{1}{2} P_{Y|X}(y|\mathbf{x}_0)\mathrm{Tr}\left[\mathrm{cov}_{\pi_{X|R}}(X)\, GG^\top\right] + \mathcal{O}\left(\mathrm{diam}(R^3)\right) \quad (4.11)$$

Notice that $\sum_y P_{Y|X}(y|\mathbf{x}_0) GG^\top$ is just the Fisher information at $\mathbf{x}_0$. It follows that:

$$\mathrm{I}(X|R;\, Y|R) = \frac{1}{2}\mathrm{Tr}\left[\mathrm{cov}_{\pi_{X|R}}(X)\, F(\mathbf{x}_0)\right] + \mathcal{O}\left(\mathrm{diam}(R^3)\right) \quad (4.12)$$

Finally, it is easy to verify that the covariance is $\mathcal{O}(\mathrm{diam}(R^2))$ using the same Taylor expansion of $P_{Y|X}(y|\mathbf{x})$ about $\mathbf{x}_0$. □

We conclude that as the size of the regions approaches 0, their overlap must compensate with $\mathcal{O}(\mathrm{diam}(R^2))$ in order to achieve a finite information regularizer. In what follows we construct a specific cover, with a specific type of regions, that achieves this in the limit. It

should be clear though that if the regularizer converges to a finite number, it will converge to the same formula up to a multiplicative factor, no matter the shape of the region, or the overlap factor.

Assuming that $\mathcal{X}$ has vector space structure, we cover it with a homogeneous set $\mathcal{R}$ of overlapping regions of identical shape: regions centered at the axis-parallel lattice points spaced at distance l'. Specifically, the regions are axis-parallel cubes of length l, where l is much larger than l'. Assume also that l/l' is an integer. Let $I_{l,l'}(P_{Y|X})$ be the information regularizer on the set of regions with parameters l and l'.

Each point[1] in $\mathcal{X}$ belongs to $(l/l')^d$ cubic regions from $\mathcal{R}$, where d is the dimensionality of the vector space. Let $\mathcal{R}'$ be the partitioning of $\mathcal{R}$ into atomic lattice cubes of length l'. Each region in $\mathcal{R}$ is partitioned into $(l/l')^d$ disjoint atomic cubes from $\mathcal{R}'$, and each atomic cube is contained in $(l/l')^d$ overlapping regions from $\mathcal{R}$. We may now rewrite the regularizer as a sum over the partition $\mathcal{R}'$:

$$I_{l,l'}(P_{Y|X}) \propto \sum_{R\in\mathcal{R}} P_X(R)\mathrm{I}\left(X|R;\, Y|R\right) = \sum_{R'\in\mathcal{R}'} P_X(R') \sum_{R\supseteq R'} \mathrm{I}\left(X|R;\, Y|R\right) \tag{4.13}$$

Assuming that $P_{Y|X}$ is differentiable, when l and l' are very small $\mathrm{I}\left(X|R;\, Y|R\right)$ for $R \supseteq R'$ will be approximatively equal. Denote by $\mathrm{I}_{R'}\left(X|\mathcal{R};\, Y|\mathcal{R}\right)$ the local mutual information on a region of type $\mathcal{R}$ that contains the atomic region R'. Therefore for small l and l' we have:

$$I_{l,l'}(P_{Y|X}) \propto \sum_{R'\in\mathcal{R}'} P_X(R')(l/l')^d \mathrm{I}_{R'}\left(X|\mathcal{R};\, Y|\mathcal{R}\right) \tag{4.14}$$

When l converges to 0, the above sum becomes integration:

$$\lim_{l\to 0} I_{l,l'}(P_{Y|X}) \propto \int_{\mathbf{x}\in\mathcal{X}} P_A(\mathbf{x}) \left[\lim_{l\to 0}(l/l')^d \mathrm{I}_{R'\ni\mathbf{x}}\left(A|\mathcal{R};\, Y|\mathcal{R}\right)\right] d\mathbf{x} \tag{4.15}$$

The interaction between the overlap and the asymptotics of the local mutual information is now clear. We must choose an overlap factor l/l' such that the following limit is finite:

$$\lim_{l\to 0}(l/l')^d \mathrm{I}_{R'\ni\mathbf{x}}\left(A|\mathcal{R};\, Y|\mathcal{R}\right) \tag{4.16}$$

[1] non-lattice point

Following Theorem 6, it is enough to choose $l' = l^{1+2/d}$ such that $(l/l')^d = l^{-2}$. Then we have:

$$\lim_{l\to 0} I_{l,l'}(P_{Y|X}) \propto \int_{\mathbf{x}\in\mathcal{X}} P_A(\mathbf{x}) \left\{ \mathrm{Tr}\left[F(\mathbf{x}_\mathbf{x}) \lim_{R\ni\mathbf{x},\ \mathrm{diam}(R)\to 0} \frac{\mathrm{cov}_{\pi_{X|R}}(X)}{\mathrm{diam}(R)^2} \right] \right\} d\mathbf{x} \qquad (4.17)$$

Given this form of the regularizer we can argue that regions in the shape of a cube are indeed appropriate. We start from the principle that the regularizer should not introduce any systematic directional bias in penalizing changes in the label. If the diameter of a region R is small enough, $\pi_{A|R}(\mathbf{x}|R)$ is almost uniform on R, and $P_{Y|X}(y|\mathbf{x})$ can be approximated well by $\mathbf{v}\cdot\mathbf{x} + c$, where $\mathbf{v}$ is the direction of highest variation. In this setting we have the following result [21]:

Theorem 7 *Let R be such that* $\mathrm{diam}(R) = 1$ *and* $P_{Y|X}(y|\mathbf{x}) = \mathbf{v}\cdot\mathbf{x} + c$. *The local information regularizer is independent of* $\mathbf{v}/\|\mathbf{v}\|$ *if and only if* $\mathrm{cov}_{\pi_{X|R}}(X)$ *is a multiple of the identity.*

Proof Let $\mathbf{x}_0 = \mathrm{E}_{\pi_{X|R}}[X]$ be the average value of $\mathbf{x}$ in the region. We have $F(\mathbf{x}_0) = \mathbf{v}\mathbf{v}^\top$. The relevant quantity that should be independent of $\mathbf{v}/\|\mathbf{v}\|$ is therefore $\mathbf{v}^\top \mathrm{cov}_{\pi_{X|R}}(X)\,\mathbf{v}$. Let $v = \Phi_i/\|\Phi_i\|$, where Φ_i is an eigenvector of $\mathrm{cov}_{\pi_{X|R}}(X)$ of eigenvalue ϕ_i. Then $\mathbf{v}^\top \mathrm{cov}_{\pi_{X|R}}(X)\,\mathbf{v} = \phi_i$ should not depend on the eigenvector. If follows that $\mathrm{cov}_{\pi_{X|R}}(X)$ has equal eigenvalues, thus $\mathrm{cov}_{\pi_{X|R}}(X) = \phi\mathbf{I}$. The converse is trivial. □

It follows that in order to remove any directional bias, $\mathrm{cov}_{\pi_{X|R}}(X) \approx \mathrm{diam}(R)^2 \cdot \mathbf{I}$, as it is the case if R is a cube or a sphere. Substituting into equation (4.17), we reach our final form of the infinitesimal information regularizer for continuous metric space when the marginal is fully known, and without placing any parametric biases:

$$I(P_{Y|X}) \propto \int_{\mathbf{x}\in\mathcal{X}} P_X(\mathbf{x})\mathrm{Tr}\left[F(\mathbf{x})\right] d\mathbf{x} \qquad (4.18)$$

where the Fisher Information is given by

$$F(\mathbf{x}) = \mathrm{E}_{P_{Y|X}(\cdot|\mathbf{x})}\left[\nabla_\mathbf{x} \log P_{Y|X}(y|\mathbf{x}) \cdot \nabla_\mathbf{x} \log P_{Y|X}(y|\mathbf{x})^\top\right] \qquad (4.19)$$

Note the the dependence of $\mathcal{R}$ is only implicit, and that we removed any multiplicative constants on purpose. The claim is that we reach the same formula up to a multiplicative constant for a variety of (unbiased) region covers.

4.1.2 Classification algorithm

We discuss classification algorithms based on the infinitesimal information regularizer as in equation (4.18). The task is to estimate a label probability distribution $P_{Y|X}(\cdot|\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$ (or, equivalently, for every $\mathbf{x} \in \mathcal{X}$), given the following inputs:

- full knowledge of P_A
- a labeled training sample $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l\}$, where all $\{y_{\mathbf{x}_i}\}_{i=1..l}$ are observed.

Note that we do not require explicit unlabeled training data, because all unlabeled evidence is implicitly represented by the knowledge of P_A. Under this interpretation the task is in fact equivalent to having infinitely many unlabeled samples.

According to the information regularization principle we need to maximize the regularized log-likelihood of the labeled training sample:

$$\max_{\{P_{Y|X}(\cdot|\mathbf{x})\,;\,\mathbf{x}\in\mathcal{X}\}} \sum_{i=1}^{l} \log P_{Y|X}(y_{\mathbf{x}_i}|\mathbf{x}_i) - \lambda \int_{\mathbf{x}\in\mathcal{X}} P_A(\mathbf{x}) \mathrm{Tr}\left[F(\mathbf{x})\right] d\mathbf{x} \tag{4.20}$$

where $F(\mathbf{x}) = \mathrm{E}_{P_{Y|X}(y|\mathbf{x})}\left[\nabla_{\mathbf{x}} \log P_{Y|X}(y|\mathbf{x}) \cdot \nabla_{\mathbf{x}} \log P_{Y|X}(y|\mathbf{x})^\top\right]$, and the maximization is subject to $0 \leq P_{Y|X}(y|\mathbf{x}) \leq 1$ and $\sum_{y\in\mathcal{Y}} P_{Y|X}(y|\mathbf{x}) = 1$.

Let us reflect for a moment on the structure of the optimization criterion. The only component that relates the information received from the labeled samples to the rest of the labels is the information regularizer. Without imposing any parametric constraints, the information regularizer is able to propagate labels from labeled samples to the entire space. In fact, we show that if we fix the label distributions at the observed samples, $P_{Y|X}(\cdot|\mathbf{x}_i) = P^0_{Y|X}(\cdot|\mathbf{x}_i)$, there is a unique set of label distributions that maximizes the objective (or minimizes the regularizer). For clarity we restrict the analysis to binary classification: $\mathcal{Y} = \{-1, 1\}$.

Theorem 8 *[21, 53] The functions $P_{Y|X}(1|\mathbf{x})$ and $P_{Y|X}(-1|\mathbf{x})$ that are differential with respect to $\mathbf{x}$ on $\mathcal{X} \setminus \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$, and continuous on $\mathcal{X}$, and that minimize:*

$$\int_{\mathbf{x}\in\mathcal{X}} P_A(\mathbf{x}) \mathrm{Tr}\left[F(\mathbf{x})\right] d\mathbf{x} \tag{4.21}$$

subject to $0 \leq P_{Y|X}(y|\mathbf{x}) \leq 1$ *and* $P_{Y|X}(1|\mathbf{x}) + P_{Y|X}(-1|\mathbf{x}) = 1$, *are also a solution to the following differential equation:*

$$\nabla_{\mathbf{x}} \log P_A(\mathbf{x}) \cdot \nabla_{\mathbf{x}} P_{Y|X}(1|\mathbf{x})^{\top} + \mathrm{Tr}\left[\nabla^2_{\mathbf{xx}} P_{Y|X}(1|\mathbf{x})\right] + \frac{1}{2} \frac{P_{Y|X}(1|\mathbf{x}) - P_{Y|X}(-1|\mathbf{x})}{P_{Y|X}(1|\mathbf{x}) P_{Y|X}(-1|\mathbf{x})} \left\|\nabla_{\mathbf{x}} P_{Y|X}(1|\mathbf{x})\right\|^2 = 0 \tag{4.22}$$

Moreover, the solution to the differential equation is unique subject to the boundary conditions $P_{Y|X}(\cdot|\mathbf{x}_i) = P^0_{Y|X}(\cdot|\mathbf{x}_i)$, *for all* $1 \leq i \leq l$, *and* $\lim_{\mathbf{x}\to\infty} \nabla_{\mathbf{x}} P_{Y|X}(1|\mathbf{x}) = 0$.

Proof The differential equation is just the Euler-Lagrange condition of the calculus of variations that must be satisfied by any function that minimizes the integral. The solution is unique on any differentiable compact set as long as the boundary of the compact set is fixed. In this case the boundary is $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ and ∞. □

The differential equation thus defines the solution to the optimization problem implicitly. In order to find explicit label distributions that optimize (4.20) one could solve the differential equation numerically for various values $\{P^0_{Y|X}(y_{\mathbf{x}_i}|\mathbf{x}_i)\}_{i=1\ldots l}$, then optimize with respect to $P^0_{Y|X}(y_{\mathbf{x}_i}|\mathbf{x}_i)$. Unfortunately, solving the differential equation numerically involves discretizing $\mathcal{X}$, which is impractical for all but low dimensional spaces. That is why the non-parametric but inductive (find a label for each point in $\mathcal{X}$) information regularization is of more theoretical than practical interest.

Nevertheless, if $\mathcal{X}$ is the one-dimensional real line the differential equation can be solved analytically [21]. We introduce the solution here to illustrate the type of biases imposed by the information regularizer. When $\mathcal{X}$ is one dimensional, the labeled samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l$ split the real line into disjoint intervals; thus if $P^0_{Y|X}(\cdot|\mathbf{x}_i)$ are given, the differential equation can be solved independently on each interval determined by the samples. The solution only depends on the labels of the endpoints, and is given by the following:

$$P_{Y|X}(1|\mathbf{x}) = \frac{1}{1 + \tan^2\left(-c \int \frac{1}{P_A(\mathbf{x})}\right)} \tag{4.23}$$

where c and the additive constant in $\int 1/P_A$ can be determined from the values of the conditional at the endpoints. These two parameters need not be the same on different intervals.

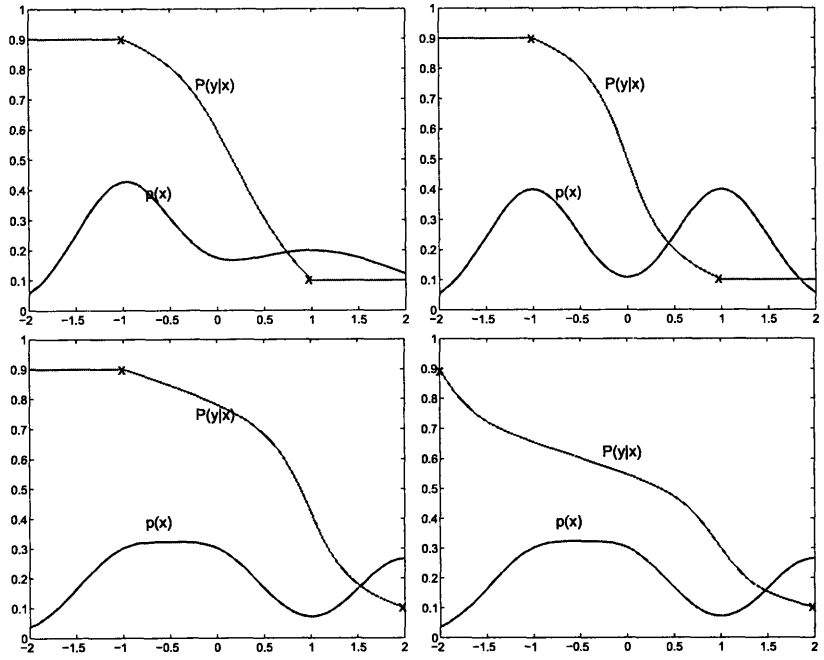

Figure 4-1: Non-parametric conditionals that minimize the information regularizer for various one-dimensional data densities while the label at boundary labeled points is fixed

Figure 4-1 shows the influence of various data distributions $P_A(\mathbf{x})$ on $P_{Y|X}(1|\mathbf{x})$ through information regularization under the boundary conditions $P_{Y|X}(y = 1|\mathbf{x} = 0) = 0.9$ and $P_{Y|X}(y = 1|\mathbf{x} = 1) = 0.1$. The property of preferring changes in the label in regions of low data density is evident. Note that the optimal $P_{Y|X}(1|\mathbf{x})$ will always be between its values at the boundary; otherwise for some $\mathbf{x}_1 \neq \mathbf{x}_2$ we would have $P_{Y|X}(1|\mathbf{x}_1) = P_{Y|X}(1|\mathbf{x}_2)$, and because the cumulative variation is minimized, necessarily $P_{Y|X}(1|\mathbf{x}) = P_{Y|X}(1|\mathbf{x}_1)$ for every $\mathbf{x} \in [\mathbf{x}_1, \mathbf{x}_2]$.

4.2 Finite unlabeled sample

In this section we substitute the requirement of full knowledge of P_A, which is unrealistic in any practical application, with the availability of a set of unlabeled training samples $\{\mathbf{x}_{\mathbf{x}_{l+1}}, \ldots, \mathbf{x}_{\mathbf{x}_n}\}$. we also show how to reconcile the infinitesimal information regularizer with parametric constraints on $P_{Y|X}$ that may be known to describe the task accurately.

Although it is possible to approach this scenario directly by partitioning the space into

regions as in [53], here we reduce the task to the situation in which the full marginal is known by replacing the full marginal with an empirical estimate obtained from the unlabeled sample.

We illustrate this method on *logistic regression*, in which we restrict the conditional to linear decision boundaries with the following parametric form: $P_{Y|X}(y|\mathbf{x};\theta) = \sigma(y\theta^\top \mathbf{x})$, where $y \in \{-1, 1\}$ and $\sigma(x) = 1/(1+\exp(-x))$. The Fisher information is therefore $F(\mathbf{x};\theta) = \sigma(\theta^\top \mathbf{x})\sigma(-\theta^\top \mathbf{x})\theta\theta^\top$ and according to equation (4.18) the information regularizer takes the form

$$\|\theta\|^2 \int \hat{P}_A(\mathbf{x})\sigma(\theta^\top \mathbf{x})\sigma(-\theta^\top \mathbf{x})d\mathbf{x} \tag{4.24}$$

Here $\hat{P}_A$ is the empirical estimate of the true marginal. We compare two ways of estimating P_A: the empirical approximation $\frac{1}{n}\sum_{j=1}^{n}\delta(\mathbf{x}-\mathbf{x}'_j)$, as well as a Gaussian kernel density estimator. The empirical approximation leads to optimizing the following criterion:

$$\max_{\theta} \sum_{i=1}^{l} \log \sigma(y_{\mathbf{x}_i}\theta^\top \mathbf{x}_{\mathbf{x}_i}) - \|\theta\|^2 \frac{\lambda}{n}\sum_{j=1}^{n}\sigma(\theta^\top \mathbf{x}_{\mathbf{x}_j})\sigma(-\theta^\top \mathbf{x}_{\mathbf{x}_j}) \tag{4.25}$$

It is instructive to contrast this information regularization objective with the criterion optimized by *Transductive Support Vector Machines* (TSVM's), as in [34]. Changing the SVM loss function to logistic loss, transductive SVM/logistic regression optimizes:

$$\max_{\theta, y_{\mathbf{x}_{l+1}}, \ldots, y_{\mathbf{x}_n}} \sum_{i=1}^{n} \log \sigma(y_{\mathbf{x}_i}\theta^\top \mathbf{x}_{\mathbf{x}_i}) - \frac{\lambda}{2}\|\theta\|^2 \tag{4.26}$$

over all labelings of unlabeled data. In contrast, our algorithm contains the unlabeled information in the regularizer.

The presented information regularization criterion can be easily optimized by gradient-ascent or Newton type algorithms. Note that the term

$$\sigma(\theta^\top \mathbf{x})\sigma(-\theta^\top \mathbf{x}) = P_{Y|X}(1|\mathbf{x})P_{Y|X}(-1|\mathbf{x})$$

focuses on the decision boundary. Therefore compared to the standard logistic regression regularizer $\|\theta\|^2$, we penalize more decision boundaries crossing regions of high data density. Also, the term makes the regularizer non-convex, making optimization potentially more difficult. This level of complexity is however unavoidable by any semi-supervised

algorithm for logistic regression, because the structure of the problem introduces locally optimal decision boundaries.

If unlabeled data is scarce, we may prefer a kernel estimate $\hat{P}_A(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}_{\mathbf{x}_j})$ to the empirical approximation, where $K(\cdot,\cdot)$ is a kernel density with the restriction that the regularization integral remains tractable. In logistic regression, if the kernels are Gaussian we can make the integral tractable by approximating $\sigma(\theta^\top \mathbf{x})\sigma(-\theta^\top \mathbf{x})$ with a degenerate Gaussian. Either from the Laplace approximation, or the Taylor expansion $\log(1+e^x) \approx \log 2 + x/2 + x^2/8$, we derive the following approximation, as in [21]:

$$\sigma(\theta^\top \mathbf{x})\sigma(-\theta^\top \mathbf{x}) \approx \frac{1}{4}\exp\left(-\frac{1}{4}(\theta^\top \mathbf{x})^2\right) \tag{4.27}$$

With this approximation computing the integral of the regularizer over the kernel centered μ of variance $\tau\mathbf{I}$ becomes integration of a Gaussian:

$$\begin{aligned}\frac{1}{4}\exp\left(-\frac{1}{4}(\theta^\top \mathbf{x})^2\right)\mathcal{N}(\mathbf{x};\,\mu,\tau\mathbf{I}) = \\ \frac{1}{4}\sqrt{\frac{\det\Sigma_\theta}{\det\tau\mathbf{I}}}\exp\left(-\frac{\mu^\top(\tau\mathbf{I}-\Sigma_\theta)\,\mu}{2\tau^2}\right)\mathcal{N}\left(\mathbf{x};\,\frac{\Sigma_\theta\mu}{\tau},\Sigma_\theta\right)\end{aligned} \tag{4.28}$$

where $\Sigma_\theta = \left(\frac{1}{\tau}\mathbf{I} + \frac{1}{2}\theta\theta^\top\right)^{-1} = \tau\left[\mathbf{I} - \frac{1}{2}\theta\theta^\top / \left(\frac{1}{\tau} + \frac{1}{2}\|\theta\|^2\right)\right]$

After integration only the multiplicative factor remains:

$$\frac{1}{4}\left(1+\frac{\tau}{2}\|\theta\|^2\right)^{-\frac{1}{2}}\exp\left(-\frac{1}{4}\frac{(\theta^\top\mu)^2}{1+\frac{\tau}{2}\|\theta\|^2}\right) \tag{4.29}$$

Therefore if we place a Gaussian kernel of variance $\tau\mathbf{I}$ at each sample $\mathbf{x}_{\mathbf{x}_j}$ we obtain the following approximation to the information regularization penalty:

$$\frac{\|\theta\|^2}{\sqrt{1+\frac{\tau}{2}\|\theta\|^2}}\frac{1}{4n}\sum_{j=1}^{n}\exp\left(-\frac{1}{4}\frac{(\theta^\top\mathbf{x}_{\mathbf{x}_j})^2}{1+\frac{\tau}{2}\|\theta\|^2}\right) \tag{4.30}$$

This regularizer can be also optimized by gradient ascent or Newton's method.

4.2.1 Logistic regression experiments

We demonstrate the logistic information regularization algorithm as derived in the previous section on synthetic classification tasks. The data is generated from two bivariate Gaussian

densities of equal covariance, a model in which the linear decision boundary can be Bayes optimal. However, the small number of labeled samples is not enough to accurately estimate the model, and we show that information regularization with unlabeled data can significantly improve error rates.

We compare a few criteria: logistic regression trained only on labeled data and regularized with the standard $\|\theta\|^2$; logistic regression regularized with the information regularizer derived from the empirical estimate to P_A ; and logistic regression with the information regularizer derived from a Gaussian kernel estimate of P_A.

We have optimized the regularized likelihood $L(\theta)$ both with gradient ascent $\theta \leftarrow \theta + \mathbf{x}\nabla_\theta L(\theta)$, and with Newton's method (iterative re-weighted least squares) $\theta \leftarrow \theta - \mathbf{x}\nabla^2_{\theta\theta} L(\theta)^{-1}\nabla_\theta L(\theta)$ with similar results. Newton's method converges with fewer iterations, but computing the Hessian becomes prohibitive if data is high dimensional, and convergence depends on stronger assumptions that those for gradient ascent. Gradient ascent is safer but slower.

We ran 100 experiments with data drawn from the same model and averaged the error rates to obtain statistically significant results. In Figure 4-2 ([21])we have obtained the error rates on 5 labeled and 100 unlabeled samples. On each data set we initialized the iteration randomly multiple times. We set the kernel width τ of the Gaussian kernel approximation to the regularizer by standard cross-validation for density estimation. Nevertheless, on such large number of unlabeled samples the information regularizers derived from kernel and empirical estimates perform indistinguishable. They both outperform the standard supervised regularization significantly.

4.3 Learning theoretical properties

We extend the analysis of information regularization on metric spaces under the assumption of full knowledge of the marginal with a learning theoretical framework. In the non-parametric setting, without the bias imposed by the information regularizer learning would certainly not be possible: no matter how much labeled training data we see, we would still not be able to learn $P_{Y|X}$ because the only constraint on $P_{Y|X}(\cdot|\mathbf{x})$ is that it is a piecewise

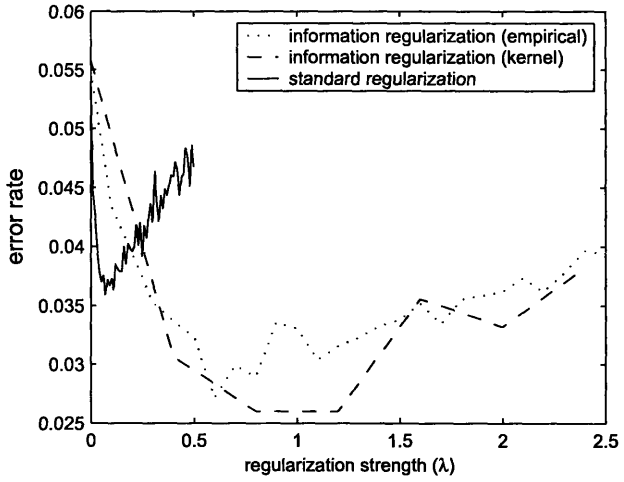

Figure 4-2: Average error rates of logistic regression with and without information regularization on 100 random selections of 5 labeled and 100 unlabeled samples from bivariate Gaussian classes

differentiable function of $\mathbf{x}$. The aim of this section is to show that the introduction of information regularization, without any other parametric constraints, is sufficient to make the conditional learnable. While the learning framework is general, due to technical constraints [2] we derive an explicit sample-size bound only for binary classification when $\mathcal{X}$ is one-dimensional.

We need to formalize the concepts, the concept class (from which to learn them), and a measure of achievement consistent with (4.20). The key is then to show that the task is learnable in terms of the complexity of the concept class.

Standard PAC-learning of indicator functions of class membership will not suffice for our purpose. Indeed, conditionals with very small information regularizer can still have very complex decision boundaries, of infinite VC-dimension. Instead, we rely on the *p-concept* [36] model of learning full conditional densities: concepts are functions $h(y|\mathbf{x}) : \mathcal{X} \rightarrow [0, 1]$. Then the concept class is that of conditionals with bounded informa-

[2] Only in one dimension the labeled points give rise to segments that can be optimized independently.

tion regularizer:

$$\mathcal{I}_\gamma(P_A) = \left\{ h : \sum_{y \in \mathcal{Y}} h(y|\mathbf{x}) = 1 \text{ and } \int_{\mathcal{X}} P_A(\mathbf{x}) \sum_{y \in \mathcal{Y}} h(y|\mathbf{x}) \left\| \nabla_{\mathbf{x}} \log h(y|\mathbf{x}) \right\|^2 d\mathbf{x} \leq \gamma \right\} \tag{4.31}$$

We measure the quality of learning by a loss function $L_h : \mathcal{X} \times \mathcal{Y} \rightarrow [0, \infty)$. This can be the log-loss $-\log h(y|\mathbf{x})$ associated with maximizing likelihood, or the square loss $(h(y|\mathbf{x}) - 1)^2$. The goal is to estimate from a labeled sample a concept h_{opt} from $\mathcal{I}_\gamma(P_A)$ that minimizes the expected loss $\mathrm{E}_{P_A P^\star_{Y|X}}[L_h]$, where $P^\star_{Y|X}$ is the true conditional.

One cannot devise an algorithm that optimizes the expected loss directly, because this quantity depends on the unknown $P^\star_{Y|X}$. We make the standard approximation of estimating h_{opt} by minimizing instead the empirical estimate of the expected loss from the labeled sample:

$$\hat{h} = \arg \min_{h \in \mathcal{I}_\gamma(P_A)} \hat{\mathrm{E}}[L_h] = \arg \min_{h \in \mathcal{I}_\gamma(P_A)} \frac{1}{l} \sum_{i=1}^{l} L_h(\mathbf{x}_i, y_{\mathbf{x}_i}) \tag{4.32}$$

If the loss function is the log-loss, finding $\hat{h}$ is equivalent to maximizing the information regularization objective (4.20) for a specific value of λ. However, we will present the learning bound for the square loss, as it is bounded and easier to work with. A similar result holds for the log-loss by using the equivalence results between the log-loss and square-loss presented in [42].

The question is how different $\hat{h}$ (estimated from the sample) and h_{opt} (estimated from the true conditional) can be due to this approximation. Learning theoretical results provide guarantees that given enough labeled samples the minimization of $\hat{\mathrm{E}}[L_h]$ and $\mathrm{E}_{P_A P^\star_{Y|X}}[L_h]$ are equivalent. We say the task is learnable if with high probability in the sample the empirical loss converges to the true loss uniformly for all concepts as $l \rightarrow \infty$. This guarantees that $\mathrm{E}[L_{\hat{h}}]$ approximates $\mathrm{E}\left[L_{h_{opt}}\right]$ well. Formally,

$$\Pr\{\exists h \in \mathcal{I}_\gamma(P_A) : |\hat{\mathrm{E}}[L_h] - \mathrm{E}[L_h]| > \epsilon\} \leq \delta \tag{4.33}$$

where the probability is with respect to all samples of size l. The inequality should hold for l polynomially large in $1/\epsilon, 1/\delta, 1/\gamma$.

We have the following sample complexity bound on the square loss, derived in [21]:

Theorem 9 *Let* $\epsilon, \delta > 0$. *Then*

$$\Pr\{\exists h \in \mathcal{I}_\gamma(P_A) : |\hat{\mathrm{E}}[L_h] - \mathrm{E}[L_h]| > \epsilon\} < \delta \tag{4.34}$$

where the probability is over samples of size l *greater than*

$$\mathcal{O}\left(\frac{1}{\epsilon^4}\left(\log\frac{1}{\epsilon}\right)\left[\log\frac{1}{\delta} + c_{P_A}(m_{P_A}^{-1}(\epsilon^2)) + \frac{\gamma}{(m_{P_A}^{-1}(\epsilon^2))^2}\right]\right) \tag{4.35}$$

Here $m_{P_A}(\beta) = \Pr\{\mathbf{x} : P_A(\mathbf{x}) \leq \beta\}$, *and* $c_{P_A}(\beta)$ *is the number of disconnected sets in* $\{\mathbf{x} : P_A(\mathbf{x}) > \beta\}$.

Measures of learning complexity

Let us explain the significance of m_{P_A} and c_{P_A} in more detail. The sample size for a desired learning accuracy must be a function of the *complexity* of $\mathcal{I}_\gamma(P_A)$, like VC-dimension in PAC-learning. One such measure is the bound on the information regularizer γ; however, we should also take into account the complexity of P_A.

The quantities $m_{P_A}(\cdot)$ and $c_{P_A}(\cdot)$ characterize how difficult the classification is due to the structure of P_A. Learning is more difficult when significant probability mass lies in regions of small P_A because in such regions the variation of $h(y|\mathbf{x})$ is less constrained. Also, the larger $c_{P_A}(\cdot)$ is, the labels of more "clusters" need to be learned from labeled data. The two measures of complexity are well-behaved for the useful densities. Densities of bounded support, Laplace and Gaussian, as well mixtures of these have $m_{P_A}(\beta) < u\beta$, where u is some constant. Mixtures of single-mode densities have $c_{P_A}(\beta)$ bounded by the number of mixtures.

For future use, let us also define the following related quantities: For each $\beta \in [0, 1)$ let $M_{P_A}(\beta) = \{\mathbf{x} : P_A(\mathbf{x}) \leq \beta\}$ be the points of density below β. Let $C_{P_A}(\beta)$ be the partition of $\mathcal{X} \setminus M_{P_A}(\beta)$ into maximal disjoint intervals. Note that $m_{P_A}(\beta) = \Pr[M_{P_A}(\beta)]$, and $c_{P_A}(\beta)$ is the cardinality of C_{P_A}.

Derivation of the learning bound

Lemma 10 *For* $\mathbf{x}_1 < \mathbf{x}_2$, $\mathcal{Y} = \{-1, 1\}$

$$\int_{\mathbf{x}_1}^{\mathbf{x}_2} p(\mathbf{x})\mathrm{E}\left[\left(\frac{d}{d\mathbf{x}}\log h(y|\mathbf{x})\right)^2\right] \geq \frac{4\left(h(1|\mathbf{x}_1) - h(1|\mathbf{x}_2)\right)^2}{\int_{\mathbf{x}_1}^{\mathbf{x}_2}\frac{1}{p(\mathbf{x})}d\mathbf{x}} \tag{4.36}$$

where the expectation is with respect to $h(y|\mathbf{x})$.

Proof After rewriting the expected value we use Cauchy-Schwartz, then $h(1|\mathbf{x})h(-1|\mathbf{x}) \leq \frac{1}{4}$:

$$\int_{\mathbf{x}_1}^{\mathbf{x}_2} \frac{1}{p(\mathbf{x})} d\mathbf{x} \cdot \int_{\mathbf{x}_1}^{\mathbf{x}_2} p(\mathbf{x}) \frac{\left(\frac{d}{d\mathbf{x}} h(1|\mathbf{x})\right)^2}{h(1|\mathbf{x})h(-1|\mathbf{x})} d\mathbf{x} \geq$$
$$\left(\int_{\mathbf{x}_1}^{\mathbf{x}_2} \frac{\frac{d}{d\mathbf{x}} h(1|\mathbf{x})}{\sqrt{h(1|\mathbf{x})h(-1|\mathbf{x})}} d\mathbf{x} \right)^2 \geq 4 \left(\int_{\mathbf{x}_1}^{\mathbf{x}_2} \frac{d}{d\mathbf{x}} h(1|\mathbf{x}) d\mathbf{x} \right)^2 \tag{4.37}$$

□

Lemma 11 *The square loss* $L_h = (h(y|\mathbf{x}) - 1)^2$ *satisfies*

$$|\mathrm{E}\left[L_{h_1}\right] - \mathrm{E}\left[L_{h_2}\right]| \leq 2\mathrm{E}\left[(h_1(1|\mathbf{x}) - h_2(1|\mathbf{x}))^2\right]^{\frac{1}{2}}$$
$$|\hat{\mathbf{E}}\left[L_{h_1}\right] - \hat{\mathbf{E}}\left[L_{h_2}\right]| \leq 2\left[\frac{1}{n}\sum_{i=1}^{n}(h_1(1|\mathbf{x}_i) - h_2(1|\mathbf{x}_i))^2\right]^{\frac{1}{2}}$$

Proof A simple application of Cauchy's inequality. □

Theorem 12 *For every* $\beta \in (0,1)$ *and* M *there exist points* $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ *from* $\mathcal{X}$ *such that any* $h_1, h_2 \in \mathcal{I}_\gamma(P_A)$ *with* $|h_1(1|\mathbf{x}_i) - h_2(1|\mathbf{x}_i)| \leq \tau, i = 1 \ldots M$, $\tau \in (0,1)$ *satisfy*

$$|\mathrm{E}_{P_A(\mathbf{x})}\left[L_{h_1}\right] - \mathrm{E}_{P_A(\mathbf{x})}\left[L_{h_2}\right]| \leq 2\left[m_{P_A}(\beta) + \frac{\gamma}{2N^2\beta^2} + 3\tau\right]^{1/2} \tag{4.38}$$

where $N = M + 1 - 2c_{P_A}(\beta)$. *Also, with probability at least* $1 - (M+1)\exp(-2\epsilon^2 n)$ *over a sample of size* n *from* $\mathcal{X}$, *for any such* h_1 *and* h_2 *we have:*

$$|\hat{\mathbf{E}}\left[L_{h_1}\right] - \hat{\mathbf{E}}\left[L_{h_2}\right]| \leq 2\left[\epsilon + m_{P_A}(\beta) + \frac{\gamma + \gamma N\epsilon}{2N^2\beta^2} + 3\tau\right]^{1/2} \tag{4.39}$$

Proof We construct a partition $\mathcal{P}$ of $\mathcal{X} \setminus M_{P_A}(\beta)$ with intervals by intersecting the intervals that make up $C_{P_A}(\beta)$ with a partitioning of $\mathcal{X}$ into N intervals of equal probability mass. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ be the endpoints of these intervals. There are no more than $N - 1 + 2c_{P_A}(\beta)$ distinct endpoints in $\mathcal{P}$, and we choose N such that $M = N - 1 + 2c_{P_A}(\beta)$.

We bound $(h_1 - h_2)^2$ on each set of the partition $M_{P_A}(\beta) \cup \bigcup_{I \in \mathcal{P}} I$ of $\mathcal{X}$. On $M_{P_A}(\beta)$ $[h_1(1|\mathbf{x}) - h_2(1|\mathbf{x})]^2 \leq 1$ trivially. On each $I \in \mathcal{P}$ we must resort to Lemma 10 to derive an upper bound.

Let $I = (u, v)$. Note that for $\mathbf{x} \in I$, $[h_1(1|\mathbf{x}) - h_2(1|\mathbf{x})]^2 \leq 2[h_1(1|\mathbf{x}) - h_1(1|u)]^2 + 2[h_2(1|u) - h_2(1|\mathbf{x})]^2 + 3\tau$. Thus it suffices to bound the variation of each h on $(u, \mathbf{x})$. This is exactly what Lemma 10 provides:

$$[h(1|\mathbf{x}) - h(1|u)]^2 \leq \frac{R_u^{\mathbf{x}}(h)}{4} \int_u^{\mathbf{x}} \frac{d\mathbf{x}'}{P_A(\mathbf{x}')} \leq \frac{R_u^v(h)}{4} \int_u^{\mathbf{x}} \frac{d\mathbf{x}'}{P_A(\mathbf{x}')} \tag{4.40}$$

where $R_a^b(h)$ is the information regularizer of h on (a, b). Thus $[h_1(1|\mathbf{x}) - h_2(1|\mathbf{x})]^2 \leq 3\tau + \frac{1}{2}(R_u^v(h_1) + R_u^v(h_2)) \int_u^{\mathbf{x}} d\mathbf{x}'/P_A(\mathbf{x}')$. Combining this result with a similar application of Lemma 10 on $(\mathbf{x}, v)$ leads to $[h_1(1|\mathbf{x}) - h_2(1|\mathbf{x})]^2 \leq 3\tau + (R_u^v(h_1) + R_u^v(h_2))/4 \cdot \int_u^v d\mathbf{x}/p(\mathbf{x})$. Since $1/P_A(\mathbf{x}) \leq P_A(\mathbf{x})/\beta^2$ on I, for $\mathbf{x} \in I$ we have

$$[h_1(1|\mathbf{x}) - h_2(1|\mathbf{x})]^2 \leq 3\tau + \frac{R_u^v(h_1) + R_u^v(h_2)}{4N\beta^2} \tag{4.41}$$

To obtain the bound on $|\mathrm{E}[L_{h_1}] - \mathrm{E}[L_{h_2}]|$ take expectation over I of (4.41), use $\sum_I R_I(h) < \gamma$, $\int_I P_A \leq 1/N$, then apply Lemma 11. For the second part of the theorem, we upper bound $\frac{1}{n}\sum(h_1(1|\mathbf{x}_i) - h_2(1|\mathbf{x}_i))^2$ using (4.41) in terms of the fraction f_I of samples that fall in interval I, and the fraction f_0 of samples that fall in $M_\beta(P_A)$. Since $\max_I f_I < 1/N + \epsilon$ and $f_0 < m_\beta(P_A) + \epsilon$ with probability at least $1 - (M+1)\exp(-2\epsilon^2 n)$, the conclusion follows. □

We can now proceed to proving Theorem 9. Had $\mathcal{I}_\gamma(P_A)$ been finite, we could have derived a learning result from McDiarmid's inequality [41] and the union bound as in [32]:

$$\Pr[\exists h \in \mathcal{I}_\gamma(P_A) : |\hat{\mathbf{E}}[L_h] - \mathrm{E}[L_h]| > \epsilon] \leq 2|\mathcal{I}_\gamma(P_A)|e^{-2\epsilon^2 n} \tag{4.42}$$

Hence the idea of replacing $\mathcal{I}_\gamma(P_A)$ with a finite discretization $\mathcal{I}_\gamma^\epsilon(P_A)$ for which the above inequality holds. If for any h in $\mathcal{I}_\gamma(P_A)$ its representative q_h from the discretization is guaranteed to be "close", and if $|\mathcal{I}_\gamma^\epsilon(P_A)|$ is small enough, we can extend the learning result from finite sets with

$$\begin{aligned} |\hat{\mathbf{E}}[L_h] - \mathrm{E}[L_h]| \leq |\hat{\mathbf{E}}[L_h] - \hat{\mathbf{E}}[L_{q_h}]| + \\ + |\mathrm{E}[L_h] - \mathrm{E}[L_{q_h}]| + |\hat{\mathbf{E}}[L_{q_h}] - \mathrm{E}[L_{q_h}]| \end{aligned} \tag{4.43}$$

To discretize $\mathcal{I}_\gamma(P_A)$ we choose some M points from $\mathcal{X}$ and discretize possible values of h at those points into $1/\tau$ intervals of length $\tau > 0$. Any h is then represented by one

of $(1/\tau)^M$ combinations of small intervals. $\mathcal{I}_\gamma^\epsilon(P_A)$ consists of one representative from $\mathcal{I}_\gamma$ corresponding to each such combination (provided it exists). It remains to select the M points and τ to guarantee that h and q_h are "close", and $|\mathcal{I}_\gamma^\epsilon(P_A)| = (1/\tau)^M$ is small.

Our starting point is Lemma 10 that bounds the variation of h on an interval in terms of its information regularizer and $\int 1/P_A$. We can use it to bound $(h(1|\mathbf{x}) - h'(1|\mathbf{x}))^2$ on an interval $(\mathbf{x}_1, \mathbf{x}_2)$ independently of $h, h' \in \mathcal{I}_\gamma(P_A)$, provided h, h' are within τ of each other at the endpoints, and $\int_{\mathbf{x}_1}^{\mathbf{x}_2} d\mathbf{x}/P_A(\mathbf{x})$ is small. If we select the M points of $\mathcal{I}_\gamma^\epsilon$ to make $\int 1/P_A$ small on each interval of the partition (except on the tail $m_{P_A}(\beta)$), we can quantify the "closeness" of h and q_h as in Theorem 12:

$$|\mathrm{E}\left[L_h\right] - \mathrm{E}\left[L_{q_h}\right]| \leq 2\left[m_{P_A}(\beta) + \frac{\gamma}{2N^2\mathbf{x}^2} + 3\tau\right]^{1/2} \tag{4.44}$$

$$|\hat{\mathbf{E}}\left[L_h\right] - \hat{\mathbf{E}}\left[L_{q_h}\right]| \leq 2\left[\bar{\epsilon} + m_{P_A}(\beta) + \frac{\gamma + \gamma N\bar{\epsilon}}{2N^2\mathbf{x}^2} + 3\tau\right]^{1/2} \tag{4.45}$$

with probability at least $1 - (M+1)\exp(-2\bar{\epsilon}^2 n)$, where $\beta \in (0,1)$ is a free parameter to be optimized later, and $N = M + 1 - 2c_{P_A}(\beta)$. We can combine the last two inequalities and (4.42) in (4.43) and optimize over $M, \tau, \beta, \bar{\epsilon}$ to obtain a learning result.

To derive a general result (without knowing $m_{P_A}(\beta)$, $c_{P_A}(\beta)$) we must choose possibly non-optimal values of the free parameters. If $N = \frac{\gamma}{2\beta^2}$, $\bar{\epsilon} = \epsilon^2$, $\tau = \epsilon^2$, $m_{P_A}(\beta) = \epsilon^2$, we obtain the asymptotic sample size stated in the theorem.

4.4 Discussion

We derived the information regularization objective for inductive classification tasks in which each object is represented by a feature vector in a continuous metric space. Initially we also assume that the marginal distribution of data is fully known, and assumption that we later on relax by producing an estimate of the marginal from observed unlabeled data. The objective, obtained as a limiting case of the generic information regularizer, involves the information regularization regions only implicitly. In the special case in which the vector is one dimensional and the joint is non-parametric, the objective can be optimized to an analytic form of the labels. Otherwise, the objective can only be optimized approximately

under additional parametric assumptions on the joint. We illustrate one such example when the main supervised classifier is logistic regression.

Chapter 5

Information regularization on graphs

5.1 Introduction

Previously (Chapter 4) we have analyzed the information regularization framework on tasks in which the feature space is continuous, endowed with a metric that naturally induces a semi-supervised bias on label similarity. The goal has been decidedly inductive: to estimate a label for every possible feature vector, whether we have seen it during training or not. We showed that we can turn the implicit bias represented by the metric into a clean infinitesimal information regularizer that does not require further engineering of the semi-supervised bias (region selection). While the resulting model is clean from a theoretical perspective, the fact that we request the full conditional distribution on a continuous space makes the task computationally infeasible, unless we constrain the problem further with parametric assumptions.

In contrast, in this chapter we take a discrete approach to information regularization, that will enable us to produce efficient algorithms. We restrict the problem to tasks in which the space of objects is either finite, or we are only interested in computing the labels of a finite subset of objects. In other words, $\mathcal{A}$, the collection of objects whose labels we must determine, it finite.

Another advantage of the discrete setting relates to the type of similarity biases that we can incorporate in the regularizer. In the continuous setting, in order to cover the entire space we had to define infinitely many regions, which restricted their specification to re-

gions defined implicitly from a metric relevant to the labeling. In the discrete setting the number of regions and their cardinality is necessarily finite, which allows more flexibility in the way the regions are defined. The biases can originate as in the continuous setting from a metric on $\mathcal{X}$, but can also come from relations related to the labeling, as in *relational learning* [56, 55, 28]. For example, the relations can be co-citation, documents that share a word, or common anchor text for web pages.

From the outset we must clarify that the transductive formulation of information regularization that we are about to introduce is not limited to transduction, and can be extended to an inductive algorithm. While the training step will assign labels only to the finite number of objects in $\mathcal{A}$, the model will have the ability to produce without retraining an estimate of the label of every object not in $\mathcal{A}$ for which we can determine its region membership. The label will however be more accurate if we incorporate the object during the training phase as an unlabeled point.

5.2 Graph representation

The discrete version of information regularization admits a graph representation similar to other graph semi-supervised learning methods [10, 52, 62, 15, 11, 63, 61, 64, 13]. As in Figure 5-1, we can represent the semi-supervised task by a directed bipartite graph, with edges from the set of regions $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ to the set of objects $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. We connect region R_i to object α_j iff $\alpha_j \in R_i$. We associate with every region its weighting $\pi_R(R_i)$, and with every edge the weighting $P_{A|R}(\alpha_j|R_i)$ that defines the relative importance of the points that belong to R_i. Note the key difference between our representation of the task, and mainstream semi-supervised learning on graphs: standard approaches consider only pairwise relationships between points, while our definition of regions allows to specify constraints that depend on a larger number of points.

The goal is to estimate a variable interest $z \in \mathcal{Z}$ for every point $\alpha \in \mathcal{A}$. The inputs to the problem are the semi-supervised biases encoded by the graph, and the true value of the variable of interest for a limited subset of the training data: $\{z_{\alpha_1}, z_{\alpha_2}, \ldots, z_{\alpha_l}\}$. We associate with every point a distribution $P_{Z|A}(z|\alpha)$ that represents our confidence in the

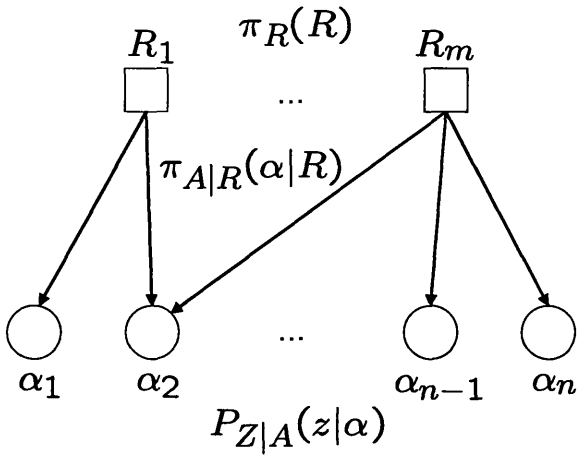

Figure 5-1: Graph representation of the semi-supervised biases of the discrete version of information regularization. The lower nodes are the data points, and the upper nodes are regions that encode label similarity biases. $\pi_R(R)$ and $\pi_{A|R}(\alpha|R)$ are task-specific weights that define the regions and must be given in advance. For some of the points (but not all) the variable of interest z is observed. The goal is to produce a probability distribution $P_{Z|A}$ for every point, that describes the likely values of the variable of interest in light of the semi-supervised biases.

value of z at α that needs to be determined.

5.3 Optimization

We develop an optimization algorithm for minimizing the objective of information regularization that can be carried out efficiently when the task is discrete (i.e. $\mathcal{R}$ and $\mathcal{A}$ are finite). As presented in equation (3.16), the information regularizer in its most generic form is given by

$$I(P_{Z|A}) = \sum_{R \in \mathcal{R}} \min_{Q_{Z|R}(\cdot|R) \in \mathcal{M}_R} \sum_{\alpha \in \mathcal{A}} \pi_{AR}(\alpha, R) \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R)\right) \tag{5.1}$$

where we have adjusted the formula to reflect that $\mathcal{A}$ is finite. Remember that the distribution families $\mathcal{M}_R$ associated with each region define the type of semi-supervised bias imposed by the region.

The information regularization approach to semi-supervised learning is to assign soft labels $P_{Z|A}$ to each data point by minimizing the following objective that combines the regularizer with the labeled evidence:

$$\min_{P_{Z|A} \in \mathcal{F}} - \sum_{i=1}^{l} \log P_{Z|A}(z_{\alpha_i}|\alpha_i) + \lambda I(P_{Z|A}) \tag{5.2}$$

Let us take a closer look at information regularization in the trivial case in which $\mathcal{R}$ consists of only one region R_0. Then finding $P_{Z|A}(\cdot|\alpha)$ for some unlabeled α amounts to finding distributions $P_{Z|A}(\cdot|\alpha) \in \mathcal{F}$ and $Q_{Z|R}(\cdot|R_0) \in \mathcal{M}_{R_0}$ that achieve the KL-divergence distance between the distribution families $\mathcal{F}$ and $\mathcal{M}_{R_0}$:

$$\min_{P_{Z|A}(\cdot|\alpha)} \min_{Q_{Z|R}(\cdot|R_0) \in \mathcal{M}_{R_0}} \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R_0)\right) \tag{5.3}$$

One popular approach to this type of problems is alternative minimization, such as the *em* algorithm that performs alternative information-geometrical projections [2]. The idea is to minimize the objective over $P_{Z|A}(\cdot|\alpha)$ and $Q_{Z|R}(\cdot|R_0)$ alternatively, by holding one set of parameters constant while optimizing the other. If $\mathcal{F}$ and $\mathcal{M}_{R_0}$ are convex, the iteration is guaranteed to converge to the unique minimum.

We follow the same alternative minimization blueprint in the general situation. Consider distributions $P^*_{Z|A}$ and $Q^*_{Z|R}$ that minimize the information regularization objective. Then necessarily:

$$\begin{aligned} P^*_{Z|A} \in \arg\min_{P_{Z|A} \in \mathcal{F}} & - \sum_{i=1}^{l} \log P_{Z|A}(z_{\alpha_i}|\alpha_i) + \\ & \lambda \sum_{R \in \mathcal{R}} \sum_{\alpha \in \mathcal{A}} \pi_{AR}(\alpha, R) \mathrm{KL}\left(P_{Z|A}(\cdot|\alpha) \,\|\, Q^*_{Z|R}(\cdot|R)\right) \end{aligned} \tag{5.4}$$

and

$$Q^*_{Z|R} \in \arg\min_{Q_{Z|R}(\cdot|R) \in \mathcal{M}_R} \sum_{R \in \mathcal{R}} \sum_{\alpha \in \mathcal{A}} \pi_{AR}(\alpha, R) \mathrm{KL}\left(P^*_{Z|A}(\cdot|\alpha) \,\|\, Q_{Z|R}(\cdot|R)\right) \tag{5.5}$$

If we can guarantee that each of the two updates above produces a single answer that can be computed efficiently, then the updates constitute the basis of an alternative minimization algorithm for performing information regularization. We show that this is the case under the following restrictions:

- $\mathcal{A}$ and $\mathcal{R}$ are finite (and not too large)
- $\mathcal{F}$ is unrestricted
- $\mathcal{M}_R$'s are also unrestricted

We require the constraining distributions to be unrestricted to guarantee that the minimum in equations (5.4) and (5.5) is achieved by unique distributions. Technically, to ensure uniqueness it is enough for these distribution families to be convex. Nevertheless, not all convex families yield updates that can be computed efficiently, and that is why for now we limit the analysis to unrestricted families. Later we will relax the *unrestricted* requirement to other examples of convex families that result in tractable updates.

According to Theorem 3, when $\mathcal{M}_R$'s are unrestricted the information regularization objective is strictly convex[1], thus the minimizing pair (P^*, Q^*) is unique and it will necessarily be reached by the alternative minimization iteration irrespective of the initialization. The following theorem provides the explicit form of the updates:

Proposition 1 *Let $P^*_{Z|A}$ and $Q^*_{Z|R}$ be distributions that minimize the objective (5.2) under the assumption that $\mathcal{F}$ and $\mathcal{M}_R$ are unrestricted. For every unlabeled data point $\alpha \in \mathcal{A} \setminus \{\alpha_1, \alpha_2, \ldots, \alpha_l\}$ and for every R the following hold:*

$$\log P^*_{Z|A}(z|\alpha) = \sum_{R \in \mathcal{R}} \pi_{R|A}(R|\alpha) \log Q^*_{Z|R}(z|R) + \text{const.} \tag{5.6}$$

$$Q^*_{Z|R}(z|R) = \sum_{\alpha \in \mathcal{A}} \pi_{A|R}(\alpha|R) P^*_{Z|A}(z|\alpha) \tag{5.7}$$

where the constant is such that the resulting conditional distribution sums to 1.

Proof If we fix $Q^*_{Z|R}(z|R)$, then for every α unlabeled $P^*_{Z|A}(\cdot|\alpha)$ is a distribution that minimizes:

$$\sum_{R \in \mathcal{R}} \pi_{AR}(\alpha, R) \sum_{z \in \mathcal{Z}} P_{Z|A}(z|\alpha) \log \frac{P_{Z|A}(z|\alpha)}{Q^*_{Z|R}(z|R)} \tag{5.8}$$

The first identity follows immediately by taking the variational derivative with respect to $P^*_{Z|A}(\cdot|\alpha)$ and equating the result to 0.

[1]Conditioned on all of $\mathcal{A}$ being covered by π_{AR}

On the other hand, if we fix P^*, then Q^* minimizes

$$-\sum_{\alpha\in\mathcal{A}}\pi_{AR}(\alpha,R)\sum_{z\in\mathcal{Z}}P^*_{Z|A}(z|\alpha)\log Q_{Z|R}(z|R) \tag{5.9}$$

The second identity also follows immediately after equating the derivative to 0. □

Theorem 1 suggests an optimization algorithm in which we treat each identity as one of two iterative updates, of P^* given Q^*, and of Q^* given P^* until convergence. Each update decreases the information regularization objective, and at convergence (P^*, Q^*) must be the unique set of distributions that minimizes the objective. The starting value does not matter[2] because the objective has a unique minimum. The algorithm can be seen as a variant of the Blahut-Arimoto algorithm in rate-distortion theory [8], where the region distributions $Q_{Z|R}(\cdot|R)$ are variational parameters.

The complexity of each update is of the order of the in-degree of the point or region operations, respectively. Thus a full iteration in the worst case scenario takes $\mathcal{O}\left(|\mathcal{A}|\cdot|\mathcal{Z}|\cdot|\mathcal{R}|\right)$ time if the bipartite graph is complete, but can take significantly less time if the bipartite graph is sparse.

A potential difficulty arises in updating $P_{Z|A}$ for points that are labeled, because the fixed-point equation is more involved, and the solution cannot be expressed analytically. Nevertheless, we can compute the solution efficiently by Newton's method, as follows. $P^*_{Z|A}(z|\alpha_i)$, $1\le i\le l$ must minimize the following formula subject to $\sum_{z\in\mathcal{Z}}P_{Z|A}(z|\alpha_i)=1$:

$$-\log P_{Z|A}(z_{\alpha_i}|\alpha_i)+\lambda\sum_{R\in\mathcal{R}}\pi_{AR}(\alpha_i,R)\sum_{z\in\mathcal{Z}}P_{Z|A}(z|\alpha_i)\log\frac{P_{Z|A}(z|\alpha_i)}{Q^*_{Z|R}(z|R)} \tag{5.10}$$

The quantity is strictly convex, thus the minimum exists and it is unique.

Placing a Lagrange multiplier on condition $\sum_{z\in\mathcal{Z}}P_{Z|A}(z|\alpha_i)=1$ we obtain that $P^*_{Z|A}(\cdot|\alpha_i)$ must be the solution to the following system of equations:

$$\sum_{z\in\mathcal{Z}}P_{Z|A}(z|\alpha_i)=1 \tag{5.11}$$

$$-\frac{\delta(z_{\alpha_i},z)}{P_{Z|A}(z|\alpha_i)}+\lambda\pi_A(\alpha_i)\log P_{Z|A}(z|\alpha_i)-\lambda\sum_{R\in\mathcal{R}}\pi_{AR}(\alpha_i,R)Q^*_{Z|R}(z|R)+\gamma=0 \tag{5.12}$$

[2]The starting value may still affect the speed of convergence, even if the final result does not depend on it.

where γ is also an unknown in the system, and the second equation must hold for all $z \in \mathcal{Z}$.

The following Newton update on $P_{Z|A}(z|\alpha_i)$ starting from the initial values $P_{Z|A}(z|\alpha_i) = 1$ converges to the solution of the system for a given γ:

$$P_{Z|A}(z|\alpha_i) \leftarrow \frac{\delta(z_{\alpha_i}, z) + \lambda \pi_A(\alpha_i) P_{Z|A}(z|\alpha_i)}{\lambda \pi_A(\alpha_i)[1 + \log P_{Z|A}(z|\alpha_i)] + \gamma - \lambda \sum_{R \in \mathcal{R}} \pi_{AR}(\alpha_i, R) Q^*_{Z|R}(z|R)} \tag{5.13}$$

It remains to find the right γ such that $\sum_{z \in \mathcal{Z}} P_{Z|A}(z|\alpha) = 1$ at convergence. We can do so by binary searching γ because at convergence $\sum_{z \in \mathcal{Z}} P_{Z|A}(z|\alpha)$ is a decreasing function of γ.

5.3.1 Message passing algorithm

As shown in the previous section, we can optimize the information regularization objective (5.2) in the case in which we do not impose any constraints on the point and region distributions with the algorithm presented in Figure 5-2.

Note that the regularization parameter λ is only used in updating the label distribution of the objects for which we have labels in the training data. λ is therefore a measure of confidence in the observed z. Typically we give full confidence to the labeled evidence, that is we set λ to 0. This amounts to fixing the label distributions of the labeled data to their observed label, and removes the special update for labeled points. At the other extreme, if λ is very large the labeled evidence is ignored, and at convergence all points, labeled or unlabeled, will have the same label.

The algorithm admits a message passing interpretation as in Figure 5-3. The information regularization iteration propagates evidence from the few observed labeled points to the unlabeled points by passing messages between regions and points. Note that there is no convergence issue if the graph has loops, as it would be in belief propagation.

Example

To clarify the information regularization algorithm we illustrate its performance on a simulated task as in Figure 5-4. The goal is two identify the two classes in the 2D plane, given one negative, one positive, and many unlabeled points. We cover the observed samples into

1. Assign distributions $Q_{Z|R}(\cdot|R)$ to every region $R \in \mathcal{R}$.

2. Initialize $P_{Z|A}(z|\alpha),\ \alpha \in \mathcal{A}$ as desired.

3. **Region update:** Recompute all region distributions by averaging point distributions:

$$Q_{Z|R}(z|R) \leftarrow \sum_{\alpha \in \mathcal{A}} \pi_{A|R}(\alpha|R) P_{Z|A}(z|\alpha)$$

4. **Point update:** Recompute all point distributions of unlabeled points by geometrically averaging region distributions:

$$P_{Z|A}(z|\alpha) \leftarrow \frac{1}{\tilde{Z}} \exp\left(\sum_{R \in \mathcal{R}} \pi_{R|A}(R|\alpha) \log Q_{Z|R}(z|R)\right)$$

   For labeled points, recompute $P_{Z|A}(\cdot|\alpha)$ by performing the Newton iteration in equation (5.13) and binary searching γ.

5. Return to Step 3 until convergence.

Figure 5-2: Description of the information regularization algorithm

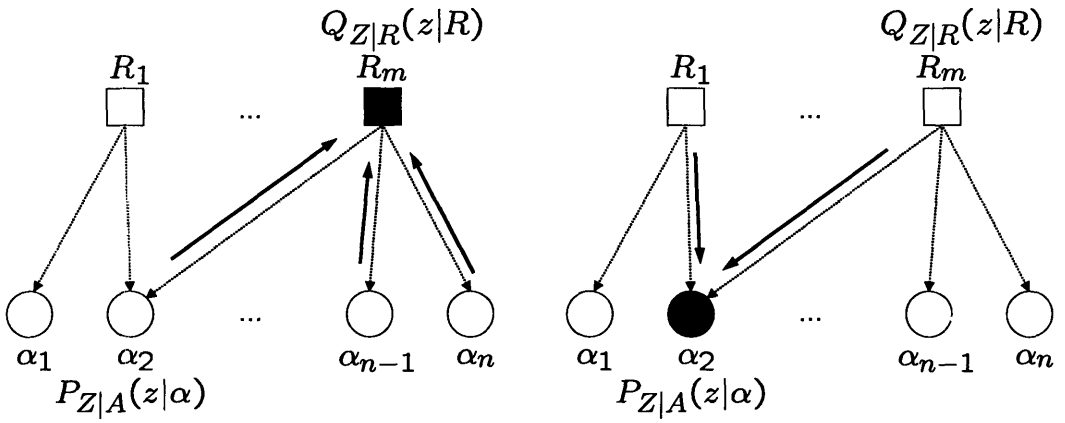

Figure 5-3: Semi-supervised learning by information regularization is a message passing algorithm. In the first part of the iteration each region aggregates messages about the variable of interest from its points. In the second part of the iteration, each point aggregates messages from the regions that contain it. We iterate until convergence.

regions based on the Euclidean distance. We associate a region with every training point, of all the training points that are within radius R of it. The number of regions is thus equal to the number of points. We weight the regions equally, and set the distribution of points with a region to uniform.

The information regularization iteration converges to a state in which the negative and positive classes are well separated, as in the figure. The performance does depend on the a priori parameter R. If R is too small, clusters of unlabeled points become disconnected from any observed labeled sample, and no information about labels can propagate to them. Thus we observe a sharp decrease in performance when the points become disconnected. On the other hand, the algorithm is robust to variations in R as long as all points remain connected. For extremely large R though the regularizer looses it resolution, in the sense that the absolute distance between the unlabeled point and labeled training samples becomes the dominant factor in assigning a label to that point.

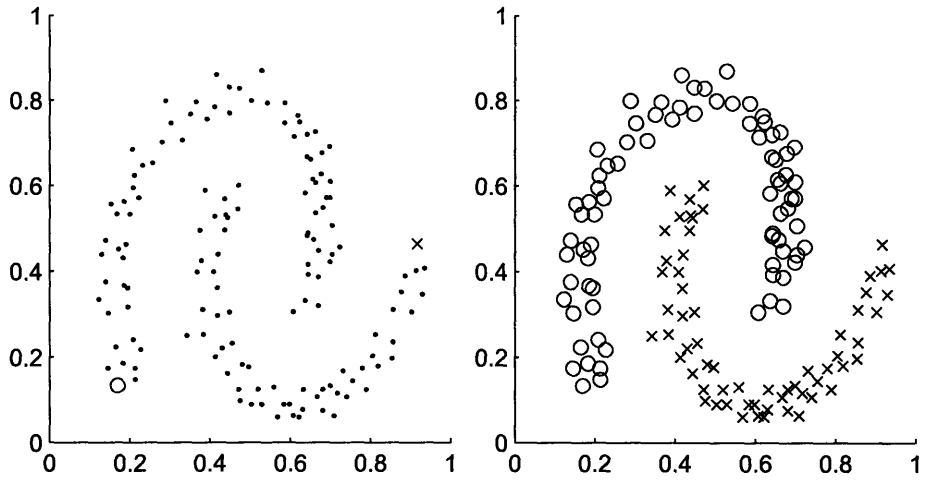

Figure 5-4: Sample run of information regularization on a 2D task where regions are defined in terms of the Euclidean distance. The left figure is the training set, while the right figure is the output of information regularization. The training set contains one positive and one negative sample, as well as many unlabeled samples.

5.3.2 Analysis of the algorithm

Computational complexity

The information regularization iteration as described requires two passes through every edge in the graph, one to propagate messages from regions to points, and one to propagate messages from points to regions. In the worst case scenario (complete bipartite graph), the number of edges in the graph is $\mathcal{O}(|\mathcal{A}| \cdot |\mathcal{R}|)$. For every edge in the graph we need to propagate information about each class, thus the worst-case complexity of the information regularization iteration is $\mathcal{O}(|\mathcal{A}| \cdot |\mathcal{R}| \cdot |\mathcal{Z}|)$.

In practice we choose regions such that the information regularization graph is sparse. However, we must be cautious not to leave the graph disconnected by making it sparse.

The information regularization iteration can be optimized for overlapping regions (or points), by reusing computation performed for the smaller region in determining the label of the larger region. For example, recomputing the labels of a cascading set of regions $R_1 \subset R_2 \subset \cdots \subset R_k$ is $\mathcal{O}(|\mathcal{Z}| \cdot |R_k|)$ rather than $\mathcal{O}(|\mathcal{Z}| \cdot (|R_1| + |R_2| + \cdots + |R_k|))$. Similarly, if two regions overlap, we can reuse the sum of the labels of the overlap in com-

puting the labels of each of the two regions. This optimization can amount to significant savings in computational complexity.

Rate of convergence

We have empirical evidence and theoretical justification (though not a complete formal proof) that the information regularization iteration converges to the optimal value exponentially fast if every node in the graph is connected to at least on labeled object (through a path of non-zero weight), in the following sense:

$$\limsup_{t\to\infty} \max_{\alpha\in\mathcal{A},z\in\mathcal{Z}} \frac{P^{t+1}_{Z|A}(z|\alpha) - P^{\star}_{Z|A}(z|\alpha)}{P^{t}_{Z|A}(z|\alpha) - P^{\star}_{Z|A}(z|\alpha)} < 1 \tag{5.14}$$

where $P^{t}_{Z|A}(z|\alpha)$ is the t'th iterate of the conditional, and $P^{\star}_{Z|A}(z|\alpha)$ is its value at convergence.

To see why this may be the case when $\lambda = 0$, consider a region that contains at least one labeled object. Let τ be the total weight of the labeled objects in the region. Then the label of region must converge to its final value at a exponential rate of $1 - \tau$, because the labeled objects, responsible for a weight of τ in the weighted update of the region, are fixed and are already at their convergence value. Then this exponential rate of convergence propagates from this region to its objects, then to the regions that contain those objects, and so on. The difficulty in making this line of thought a formal proof is that the geometric averaging of labels of the regions when recomputing a object is re-normalized to sum to 1. The normalization breaks the properties of an "average".

Nevertheless, we can say for sure that if a object is connected to a labeled object by a shortest path of M objects, information from the labeled object cannot contribute to this object in less that M full information regularization iterations. Thus in order to achieve sensible labels, the number of information regularization iterations must exceed the maximum number of objects on a shortest path between a labeled and an unlabeled object. Depending on the structure of the graph, this number may be $\mathcal{O}(n)$, for a long graph with a single label at one end, or smaller.

Unbalanced classes

Consider a task in which the variable of interest is the class label $y \in \mathcal{Y} = \{1, 2, \ldots, K\}$ that can take K values. Unfortunately, the label distribution of the labeled training data has a significant impact on the label distribution of the labels assigned by the information regularization iteration. For example, if we received 2 positive examples in Figure 5-4 but still one negative example, we expect most objects to be assigned a positive label as a result of the information regularization algorithm (if not twice as many positives than negatives). This imbalance may have a negative impact on the performance of the algorithm because such a small sample of labeled objects is very noisy and is at best a very coarse indicator of the true distribution of labels.

Note that supervised learning methods have the same difficulty learning the label distribution from such a small labeled sample. Some of them, such as SVM's, are robust to unbalanced classes, while others need more labeled training data to get a better estimate of the label distribution.

We make a simple change to the information regularization algorithm to allow to incorporate a prior on the label distribution, that may be different from the label frequencies of the labeled training data.

The solution is to change the mapping between the label distributions $P_{Y|A}(\cdot|\alpha)$ to hard classification labels from $\mathcal{Y}$. Normally , we would assign to α the label that maximizes $P_{Y|A}$. Instead, we make the decision based on a threshold vector $(t_1, t_2, \ldots, t_K)$:

$$\hat{y}_\alpha = \arg\max_{y \in \mathcal{Y}} \log P_{Y|A}(y|\alpha) + t_y \tag{5.15}$$

We select the thresholds t_y after the information regularization algorithm converges, in a way that makes the resulting label distribution equal to our prior label distribution (or a posterior label distribution if the labeled training data is sufficient to affect the prior).

Let us discuss what would happen if we do not receive any labeled sample from class $y = 1$. The regular information regularization would assign a 0 probability of belonging to that class to every object. The modified version of information regularization would correct this with a threshold that ensures that whenever a object is not explained well enough by the other classes, it will be assigned to $y = 1$.

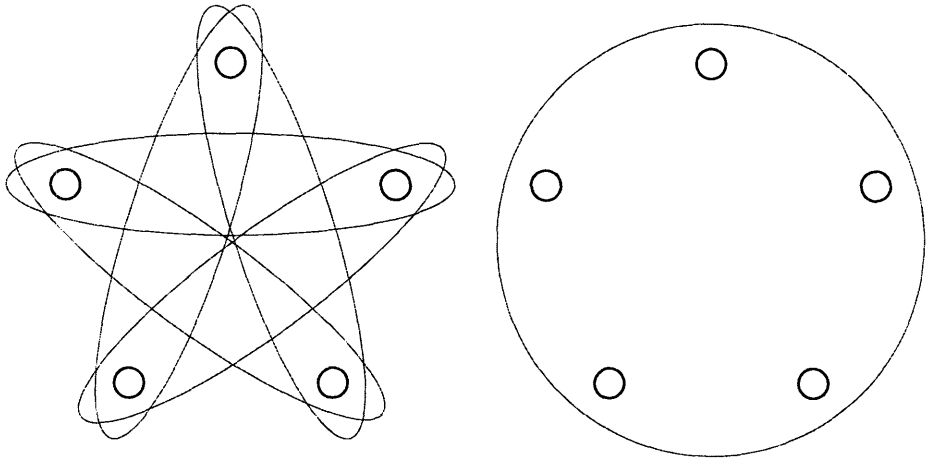

Figure 5-5: Alternative representations of the similarity bias of the labels of a number of objects: all possible pairwise regions (left) or a single region covering all objects (right).

Region size trade-off

This section illustrates the trade-off between grouping a set of objects into a single large region, or into many smaller regions. Suppose that we have the a priori bias that a certain set of objects should have similar labels. The question is whether to represent this bias as a single region containing all objects, or as many pairwise regions of pairs of objects, as in Figure 5-5. From the object of view of computational complexity it is clear that a single large region is more efficient because it introduces fewer edges in the bipartite graph. Let us evaluate the effect on the resulting probabilities.

Assume that the variable of interest is the class label, and that the classification is binary ($\mathcal{Y} = \{0, 1\}$). Assume also that $\lambda \to 0^+$, and that the weights π are uniform. There are n training objects, and only l of them are labeled. Assume that f is the fraction of objects of the labeled objects that are of class 1.

We consider two scenarios (Figure 5-5). In the first scenario we have a single region containing all objects. In the second scenario we have $n(n-1)/2$ regions of 2 objects joining any pair of objects. Because of the symmetric of the problem, in both scenarios after convergence information regularization will assign a common label to all unlabeled objects. Let us find out which is the label in each scenario.

In the single region case it is easy to see that the probability of class $y = 1$ assigned to

all the unlabeled object by information regularization converges f, the fraction of labeled objects of class $y = 1$.

In the multiple region case, let $p = P_{Y|A}(1|\alpha)$ be the probability to which the algorithm converges (for unlabeled points). Let us write the probabilities $Q_{Y|A}(1|R)$ associated with each region. We distinguish between three types of regions:

1. Both endpoints are unlabeled. Then $Q_{Y|A}(1|R)$ must be equal to p at convergence.

2. One endpoint is unlabeled, and the other one is labeled with $y = 1$. Then $Q_{Y|A}(1|R)$ must be equal to $(1 + p)/2$ at convergence.

3. One endpoint is unlabeled, and the other one is labeled with $y = 0$. Then $Q_{Y|A}(1|R)$ must be equal to $p/2$ at convergence.

Each unlabeled point belongs to exactly $n-l+1$ regions of type 1, fl regions of type 2, and $(1-f)l$ regions of type 3. According to the point update of information regularization, the distribution of the unlabeled point must be obtained as a normalized geometric average of the mentioned region distributions. The geometric averages of the regions that contain of an unlabeled point for the positive label and the negative label are, respectively[3]:

$$\frac{1}{n-1}\left[fl\log\frac{1+p}{2} + (1-f)\log\frac{p}{2} + (n-l-1)\log p\right] \tag{5.16}$$

$$\frac{1}{n-1}\left[fl\log\frac{1-p}{2} + (1-f)\log\left(1-\frac{p}{2}\right) + (n-l-1)\log(1-p)\right] \tag{5.17}$$

$$\tag{5.18}$$

These are the logarithms of two numbers whose ratio must be equal to $p/(1-p)$ at convergence. If follows that p must satisfy:

$$\frac{1}{n-1}\left[fl\log\frac{1+p}{1-p} + (1-f)l\log\frac{p}{2-p} + (n-l-1)\log\frac{p}{1-p}\right] = \log\frac{p}{1-p} \tag{5.19}$$

We can solve for f in the above equation. We get:

$$f = \frac{\log\frac{2-p}{1-p}}{\log\frac{(1+p)(2-p)}{(1-p)p}} \tag{5.20}$$

[3]expressed as logarithms

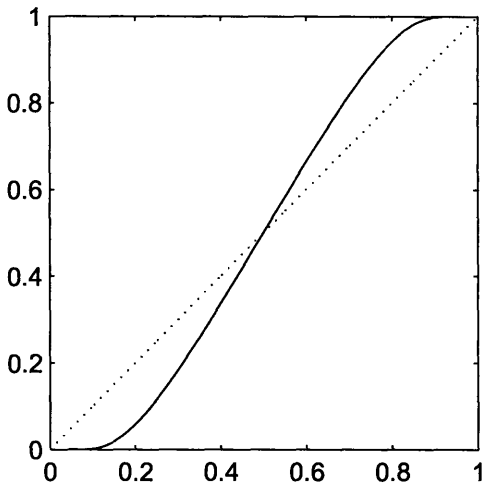

Figure 5-6: The graph of the probability $P_{Y|A}(1|\alpha)$ assigned by information regularization to unlabeled points in a binary classification task in a scenario in which all points belong to a single region (x-axis) versus a scenario in which every pair of points belongs to a region (y-axis). Refer to Figure 5-5 for a depiction of the scenarios.

In Equation 5.20 p is the probability of class $y = 1$ for all unlabeled points at convergence in the scenario with many pairwise regions, while f is the same probability in the scenario with a single region that contains all points. The relationship between the two probabilities is depicted in Figure 5-6.

We observe that in the scenario with pairwise regions, if less then 10% labeled training points are negative (or positive), information regularization ignores them, and the resulting p is approximatively 1 (or 0). Intuitively, the smaller the regions, the more geometric averaging and normalization operations. This type of operations have the tendency of driving small probabilities to 0, and large probabilities to 1, hence the shape of the graph. It is interesting that the shape of the curve does not depend on the number of points considered.

To conclude, it is not easy to decide whether to use smaller or larger regions based on the implied assumptions – it really depends on the task. Nevertheless, one large region instead of many small ones (a quadratic number of small ones) will be less demanding from the computational point of view, but offers no way of capturing spatial structure.

5.4 Learning theoretical considerations

As in the metric case, we seek to show that the information regularizer is an adequate measure of complexity, in the sense that learning a labeling consistent with a cap on the regularizer requires fewer labeled samples. We consider only the simpler setting where the labels are hard and binary, $P_{Y|A}(y|\alpha) \in \{0, 1\}$, and show that bounding the information regularizer significantly reduces the number of possible labelings. Assuming that the points in a region have uniform weights, let $N(\gamma)$ be the number of labelings of $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ consistent with

$$I(P_{Z|A}) < \gamma \tag{5.21}$$

According to [23] we have the following result:

Theorem 13 $\log_2 N(\gamma) \leq C(\gamma) + \gamma \cdot n \cdot t(\mathcal{R}) / \min_R \gamma(R)$, *where* $C(\gamma) \to 1$ *as* $\gamma \to 0$, *and* $t(\mathcal{R})$ *is a property of* $\mathcal{R}$ *that does not depend on the cardinality of* $\mathcal{R}$.

Proof Let $f(R)$ be the fraction of positive samples in region R. Because the labels are binary the mutual information I $(A|R;\, Y|R)$ is given by $H(f(R))$, where H is the entropy. If $\sum_R \pi_R(R) H(f(R)) \leq \gamma$ then certainly $H(f(R)) \leq \gamma / \pi_R(R)$. Since the binary entropy is concave and symmetric w.r.t. 0.5, this is equivalent to $f(R) \leq g_R(\gamma)$ or $f(R) >= 1 - g_R(\gamma)$, where $g_R(\gamma)$ is the inverse of H at $\gamma / \pi_R(R)$. We say that a region is mainly negative if the former condition holds, or mainly positive if the latter.

If two regions R_1 and R_2 overlap by a large amount, they must be mainly positive or mainly negative together. Specifically this is the case if $|R_1 \cap R_2| > g_{R_1}(\gamma)|R_1| + g_{R_2}(\gamma)|R_2|$ Consider a graph with vertexes the regions, and edges whenever the above condition holds. Then regions in a connected component must be all mainly positive or mainly negative together. Let $C(\gamma)$ be the number of connected components in this graph, and note that $C(\gamma) \to 1$ as $\gamma \to 0$.

We upper bound the number of labelings of the points spanned by a given connected component $\mathcal{C}$, and subsequently combine the bounds. Consider the case in which all regions in $\mathcal{C}$ are mainly negative. For any subset $\mathcal{C}'$ of $\mathcal{C}$ that still covers all the points spanned by

C,

$$f(C) \leq \frac{1}{|C|} \sum_{R \in C'} g_R(\gamma)|R| \leq \max_R g_R(\gamma) \cdot \frac{\sum_{R \in C'} |R|}{|C'|}$$

Thus $f(C) \leq t(C) \max_R g_R(\gamma)$ where $t(C) = \min_{C' \in C,\, C' \text{ cover}} \frac{\sum_{\mathcal{R} \in C'} |\mathcal{R}|}{|C'|}$ is the minimum average number of times a point in C is necessarily covered.

There at most $2^{nf(R)\log_2(2/f(R))}$ labelings of a set of points of which at most $nf(R)$ are positive. [4]. Thus the number of feasible labelings of the connected component C is upper bounded by

$$2^{1+nt(C)\max_R g_R(\gamma)\log_2(2/(t(C)\max_R g_R(\gamma)))} \tag{5.22}$$

where 1 is because C can be either mainly positive or mainly negative. By cumulating the bounds over all connected components and upper bounding the entropy-like term with $\gamma/\pi_R(R)$ we achieve the stated result. □

Therefore when γ is small, $N(\gamma)$ is exponentially smaller than 2^n, and

$$\lim_{\gamma \to 0} N(\gamma) = 2$$

5.5 Relation to other graph regularization methods

The information regularization algorithm has similar structure with other semi-supervised algorithms that operate on graphs, including harmonic graph regularization [62], and conditional harmonic mixing (CHM) [13]; yet, the updates of information regularization differ from the mentioned algorithms as follows.

Both harmonic graph regularization and CHM result only in arithmetic averaging updates, while information regularization is asymmetric in the sense that one update is geometric. The geometric combination of label distributions means that if one probability is approximately 1, the average will be also almost 1 irrespective of the distributions of other regions. If the regions are experts that vote for the label of a common point, in information regularization it is enough for a single region to be confident to set the label of the point.

[4] The result follows from $\sum_{i=0}^{k} \binom{n}{i} \leq \left(\frac{2n}{k}\right)^k$

5.6 Extensions

5.6.1 Information regularization as post-processing

In what follows we introduce a variation of the information regularization algorithm that allows us to apply it as a post-processing step on top of any probabilistic classifier. Often we are already invested in a well-engineered supervised classifier that models the task very well given enough labeled samples. Post-processing enables us to turn the classifier into a semi-supervised learning algorithm without waisting the predictive power of the supervised classifier.

The idea is to use the probabilistic output of the supervised classifier as labeled training data for information regularization. Since the supervised classifier has the ability to assign a label to any point in the space, all training data $\mathcal{A}$ will now be labeled prior to running information regularization. It may be counter-intuitive why information regularization is still appropriate, since there are no unlabeled points. Nevertheless, with an appropriate regularization weight λ the information regularization can still correct labels that do not agree with the semi-supervised bias. For example, if the input classifier places a single negative label in a cluster of positive labels, the information regularizer will correct it to a positive label.

Thus the objective on information regularization becomes:

$$\min_{P_{Z|A} \in \mathcal{F}} - \sum_{\alpha \in \mathcal{A}} \sum_{z \in \mathcal{Z}} P^0_{Z|A}(z|\alpha) \log P_{Z|A}(z|\alpha) + \lambda I(P_{Z|A}) \tag{5.23}$$

where $P^0_{Z|A}(\cdot|\alpha)$ is the probabilistic output at α of the previous classifier. The information regularization algorithm is the same, except that it needs to be modified to accept probabilistic labels of points as input by replacing the δ function in Equation 5.13 with the actual probability $P^0_{Z|A}(\cdot|\alpha)$.

The value of the regularization parameter λ is critical in running information regularization as post-processing. If $\lambda \to 0$, the initial labels are fully trusted and no changes can be made. Post-processing makes sense only for large values of λ.

In Figure 5-7 we see an example in which information regularization corrected the output of a previous classifier to account for the a priori bias that neighbors with respect to

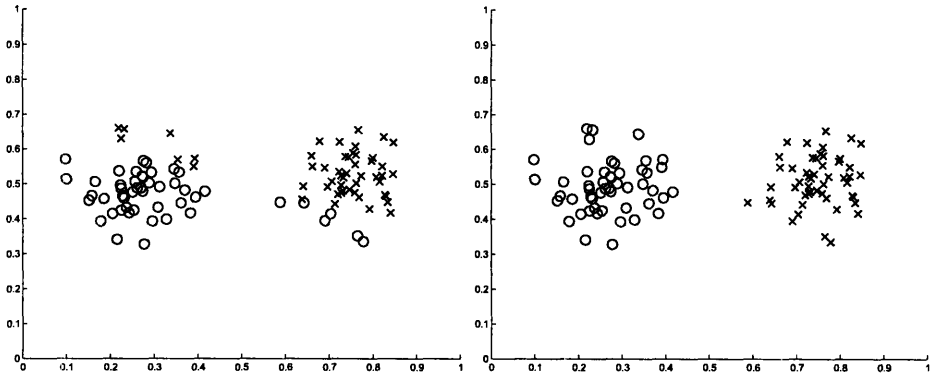

Figure 5-7: Post-processing of the probabilistic output of a supervised classifier with information regularization, in a binary classification task. Left: the output labels of the supervised classifier. Right: the output labels after information regularization, corrected to agree with the semi-supervised bias that nearby points should have similar labels.

the Euclidean metric should have similar labels.

5.6.2 Parametric models

We relax the assumption of the information regularization iteration that the distributions associated with every point and every region are unconstrained; therefore, $\mathcal{F}$ and $\mathcal{M}_R$ can now be restricted, and be defined as parametric distribution families. Before we proceed, let us reiterate the problems that may emerge due to placing restrictions on the distributions:

1. Restricting the distributions breaks the convexity of the information regularization objective.
   - we can only guarantee a local optimum
   - initialization matters
2. The restrictions may result in computationally intractable point and region updates.

If the parametric restrictions on $\mathcal{F}$ or $\mathcal{M}_R$ bring in important domain knowledge, we argue that a local optimum that considers the domain knowledge may be better than a global optimum that disregards it. Thus we agree to tolerate a certain amount of non-convexity. However, if the updates become intractable, the algorithm is compromised. Thus in our

relaxation of the original requirements for the information regularization iteration we seek parametric families that render tractable updates, ignoring non-convexity.

In the spirit of Theorem 1, we derive the general update for the distribution of region R for a generic $\mathcal{M}_R$. As in Equation 5.9, any local optimum Q^* of the information regularization objective must minimize the following as a function of Q but with fixed P^*:

$$-\sum_{\alpha\in\mathcal{A}} \pi_{AR}(\alpha, R) \sum_{z\in\mathcal{Z}} P^*_{Z|A}(z|\alpha) \log Q_{Z|R}(z|R) \tag{5.24}$$

The only difference is that the minimization is now subject to $Q_{Z|R} \in \mathcal{M}_R$. Note that the above equation can be seen as a log-likelihood of $Q_{Z|R} \in \mathcal{M}_R$ on a training set of samples from $\mathcal{Z}$ that come with frequencies $\sum_{\alpha\in\mathcal{A}} \pi_{AR}(\alpha, R) P^*_{Z|A}(z|\alpha)$.

It follows that the region update is tractable if any only if maximum likelihood estimation on distributions from $\mathcal{M}_R$ is tractable. Moreover, the region update by averaging the distribution of the points contained in it will be replaced by the maximum likelihood distribution if the samples are the points in the regions, weighted by their weight in the region.

In light of this argument, we extend information regularization to families of distributions known to be tractable for maximum likelihood estimation.

Expectation maximization

In Section 3.4.1 we have shown that with a special choice of the regions and distribution families $\mathcal{M}_R$, the information regularization objective is identical to maximum likelihood from incomplete data. Here we show that the resulting iteration for optimizing the objective is exactly the iteration of the *Expectation Maximization* (EM) algorithm [26].

In this setting all points in the training set are unlabeled, and we cover the space with a single region R that contains all points. The variable of interest Z is the pair (X, Y) that includes both a feature vector $\mathbf{x}_\alpha$ associated with each object, and the label. Also, we set $\mathcal{M}_R$ to be equal to $\{P_{XY}(\mathbf{x}, y\,;\,\boldsymbol{\theta})\,;\,\boldsymbol{\theta} \in \Theta\}$, the assumed parametric family of the joint. Since the label y is latent for all training examples, the information regularization objective is equivalent to likelihood maximization from incomplete data.

The information regularization algorithm proceeds as follows. First we randomly initialize the label distributions of all points, $P_{Y|X}(y|\mathbf{x}_\alpha)$, then we proceed with the information regularization iteration until convergence. The iteration consists of two steps. In the region update step, we set the parameters of $Q_{XY|R}$, denoted by $\boldsymbol{\theta}_R$, to the maximum likelihood estimate of a distribution from $\mathcal{M}_R$ under a training set that consists of points with labels $P_{Y|X}(y|\mathbf{x}_\alpha)$. In the point update step, we set $P_{Y|X}(y|\mathbf{x}_\alpha)$ for every point to the expected value of the label at $\mathbf{x}_\alpha$ under $Q_{XY|R}$. But this algorithm fits exactly the description of the Expectation Maximization iteration.

Thus a single region with a parametric model performs a EM-like iteration by completing the labels of unlabeled points with current estimates from the region model. This leads us to a sensible algorithm for augmenting a supervised classifier based on a generative parametric model of the joint with additional semi-supervised biases:

1. Create one region that contains all training data, such that $\mathcal{M}_R$ associated with that region with the parametric model of the generative classifier.

2. Create additional regions for other known semi-supervised biases.

The presence of a single region that covers all points solves any connectivity issues: the additional biases need not provide complete information in the form of a connected bipartite graph.

Structured labels

Here we extend the regularization framework to the case where the labels represent more structured annotations of objects. Let y be a vector of elementary labels $y = [y^1, \ldots, y^k]'$ associated with a single object α. We assume that the distribution

$$P_{Y|A}(y|\alpha) = P_{Y|A}(y^1, \ldots, y^k|\alpha)$$

, for any α, can be represented as a tree structured graphical model, where the structure is the same for all $\alpha \in \mathcal{A}$. The model is appropriate, e.g., in the context of assigning topics to documents. While the regularization principle applies directly if we leave $P_{Y|A}(y|\alpha)$

unconstrained, the calculations would be potentially infeasible due to the number of elementary labels involved, and inefficient as we would not explicitly make use of the assumed structure. Consequently, we seek to extend the regularization framework to handle distributions of the form

$$P^{\mathcal{T}}_{Y|A}(y|\alpha) = \prod_{i=1}^{k} P^{i}_{Y|A}(y^i|\alpha) \prod_{(i,j)\in\mathcal{T}} \frac{P^{ij}_{Y|A}(y^i, y^j|\alpha)}{P^{i}_{Y|A}(y^i|\alpha) P^{j}_{Y|A}(y^j|\alpha)} \tag{5.25}$$

where $\mathcal{T}$ defines the edge set of a tree. The regularization problem will be formulated over

$$\mathcal{F} = \{P^{\mathcal{T}}_{Y|A}(y|\alpha)\,;\, P^{i}_{Y|A}(y^i|\alpha), P^{ij}_{Y|A}(y^i, y^j|\alpha)\}$$

rather than unconstrained $P_{Y|A}(y|\alpha)$.

In addition, we constrain $\mathcal{M}_R$ to consist of distributions $Q_{Y|R}(y|R)$ that factor according to the same tree structure. By restricting the class of region distributions that we consider, we necessarily obtain an *upper bound* on the unrestricted information criterion. The resulting maximum likelihood region updates are simple "moment matching" updates, as if we updated each set of matching parameters independently, in parallel:

$$Q^{ij}_{Y|R}(y^i, y^j|R) \leftarrow \sum_{\alpha\in R} \pi_{A|R}(\alpha|R) P^{ij}_{Y|A}(y^i, y^j|\alpha) \tag{5.26}$$

$$Q^{i}_{Y|R}(y^i|R) \leftarrow \sum_{\alpha\in R} \pi_{A|R}(\alpha|R) P^{i}_{Y|A}(y^i|\alpha) \tag{5.27}$$

The geometric update of the point distributions is structure preserving, in the sense that if the region distributions share the same tree structure, the resulting point distribution will have the same tree structure. Therefore in effect we are leaving $\mathcal{F}$ unconstrained, and only constrain $\mathcal{M}_F$ to be structured. The structure of the point distributions is induced from the structure of the regions.

The structured extension to information regularization still has the convexity properties of the original criterion, in the sense that the optimal distributions are unique and can be found starting from any initial value.

5.7 Discussion

We have shown that restricting the information regularization framework to a finite domain with finitely many region biases results in an efficient message-passing optimization

algorithm that is guaranteed to converge to the unique optimal labels irrespective of initialization. The information regularization iteration propagates label information from the points with observed labels to the unlabeled points in a way that is consistent with the assumed semi-supervised biases.

The algorithm admits natural extensions to parametric regions biases, or parametric models of the labels. Structured labels can also be easily incorporated. The algorithm subsumes and generalizes alternative minimization iterations such as EM.

The algorithms needs a set of regions, region weights, and weights of points within each region as inputs. Learning these parameters would require a number of instances of problems belonging to the same domain. Learning π_R and $\pi_{A|R}$ from a single example (i.e. a single training data set) is difficult, but not impossible. Our experimental results in the following chapter include one such example. A full treatment of learning the regions is outside the scope of our analysis.

Chapter 6

Experiments

We illustrate the discrete version of the information regularization algorithm (Chapter 5) on a number of classification tasks. In the first experiment we blindly apply information regularization with regions derived from an Euclidean metric without knowledge of the domain from which the training set has been sampled. The following experiment demonstrates the application of information regularization to the task of categorization of web pages, where we choose the regions based on intuition about the domain. Lastly, we present information regularization applied to a named entity recognition task with a large number of objects and regions, where we provide an algorithm for region selection. With this range of experiments we hope to provide enough intuition about the performance of information regularization in practice, the sensitivity to region selection, and the ability to run on large data sets.

6.1 Generic information regularization with the Euclidean metric

We present experimental results on the performance of the discrete version of the information regularization algorithm on 6 data sets published in [14].

The benchmark is particularly challenging for semi-supervised learning because the algorithms were developed without knowledge of the domains from which data was sampled;

Table 6.1: Metrics of the data sets used in the generic experiment.

Data set	Classes	Dimension	Points
g241c	2	241	1500
g241d	2	241	1500
Digit1	2	241	1500
USPS	2	241	1500
COIL	6	241	1500
BCI	2	117	400

in fact, the data sets were preprocessed to mask any obvious link with a particular domain. Only after the publication of the book the origin of the data sets were revealed. Now that the identity of the data sets is revealed, it is instructive to comment on the dependency of the results on right or wrong assumptions. We also show that semi-supervised learning can improve significantly on supervised methods.

6.1.1 The data

We provide a brief description of the 7 data sets. The first three are artificially generated, and the rest come from real domains. Their metrics are shown in Table 6.1. The first data set is a classic instance of the cluster semi-supervised principle. The seconds dataset violates the cluster assumption, and the third has the feature that the data lies in a low-dimensional manifold. The fourth is unbalanced, and the fifth has been chosen as an example of a multi-class classification problem. The last data set is simply a noisy real-world difficult classification problem.

g241c An artificial binary-classification data set in which each class is a multivariate Gaussian of 241 dimensions.

g241d An artificial binary-classification data set in which each class is a mixture of 2 multivariate Gaussians, such that pairs of clusters from different classes overlap. This data set violates the cluster assumption.

Digit1 An artificial data set that consists of the digit "1" transformed by random rotations, translations, scalings, then rasterized in a 16×16 grid. One 241 points are kept as features, with some Gaussian noise added. The task is to predict if the digit "1" has been tilted to the right, or two the left. The data points lie into a low-dimensional manifold because of the small number of types of transformations that generated it, but does not necessarily satisfy any cluster assumption.

USPS A subset of 1500 images from the USPS digit dataset, with some rescaling, and noise applied; also, some of the features are masked so that only 241 features are available per data point. The task is binary classification, where the positive class consists of the digits "2" and "5", while the other digits go into the negative class. This is a heavily unbalanced data set, with the ratio 1 to 4 between the number of positive and negative examples.

COIL This is a data set that originated in the Columbia Object Image Library (COIL-100) [43], which is a database of 100 images taken from different angles. Processing consisted of selecting only 24 object, randomly divided into 4 classes, and choosing 241 features out of a 16×16 sub-sampling of the red channel (the original images were 128×128).

BCI The features consists of a representation of the time-series recording of 39 electroencephalography electrodes while one subject imagined performing tasks with the left hand, or with the right hand [40]. The time series were converted to features by keeping 117 fitted parameters of an auto-regressive model. The goal is to classify the recording as "left" or "right"

Each data set comes with 24 splits into labeled and unlabeled samples. 12 of them are problems with 10 labeled points, and the other 12 are problems with 100 labeled points.

6.1.2 Implementation details

In the absence of domain knowledge, we employed a generic semi-supervised prior that assumes that the distribution of the labels is correlated with Euclidean metric on the vector

space of the features. Also, we relied heavily on cross-validation to remove other implicit prior assumptions.

We implemented the discrete version of information regularization presented in detail in Chapter 5. Regions are centered at each data point, and consist of the K-nearest neighbors around the point (including the center), where the distance is measured according to the Euclidean metric. Also, regions have equal weights $\pi_R(R)$, and the weights of the points belonging to a region, $\pi_{A|R}(\alpha|R)$ are also equal.

λ, the weight of labeled training data against unlabeled data, was set to 0, meaning that the posterior labels of training data are not allowed to change from their given values. The regularization iteration proceeded until the change in parameters became insignificant.

Cross-validation

We cross-validated by 10-fold cross-validation the parameter K that governs the size of the regions, and also the choice of the thresholding function, as we will describe shortly. In order to cross-validate, we split the labeled training set in 10 equal subsets, and leave one subset out while training with the rest of the subsets. After training we compute the error rate on the subset that was left out.

Because the parameters we cross-validate should be a characteristic of the domain, not of the specific task, we average the cross-validation score across all 12 splits (of the same number of labeled samples). This alleviates the problem that the cross-validation error rate is quite noisy when only 10 labeled points are available.

In order to determine K, we run full cross-validation for 40 values of K between 2 and 400 on a logarithmic scale, such that we try all small values of K between 2 and 18, and fewer larger values.

When it is not clear what the optimal value of K is because the cross-validation assign the same minimal score to a range of values, we take the average K across those minimal values.

When computing the cross-validation error we counted as errors all points that were graph disconnected from any labeled data points, even if their probability happened to match the true label. This encouraged to select values of K that left most points connected.

Selection of the threshold

We optimize the mapping between the soft probabilities $P_{Y|A}$ that result from the information regularization algorithm, and hard output labels. Proper selection of the threshold requires full cross-validation. However, for reasons of computational efficiency, we cross-validated only between two scenarios:

- assign the class labels by maximizing $P_{Y|A}(y|\alpha) + t_y$, following the blueprint laid out in Section 5.3.2. t_y are a set of thresholds that are optimized so that the resulting class distribution matches the class frequency on the observed labeled data

- assign the class labels simply according to the maximum of $P_{Y|A}$

The first scenario is robust to unbalances between classes in the true data distribution. The second scenario works best when the class frequencies of the observed labeled points are so noisy that are not representative at all of the true class distribution.

6.1.3 Evaluation

We computed the error rates that resulted from the information regularization algorithm and compared them against the performance of a purely supervised Support Vector Machine, and that of a Semi-supervised Transductive Support Vector Machine. The average error rates for 10 and 100 labeled training samples are presented in Table 6.2 and Table 6.3 respectively. For the performance of other semi-supervised methods on the same data sets see [14].

We notice that information regularization performs better than the supervised method, except on the data sets that violate significantly the semi-supervised prior imposed by the Euclidean metric, that is *g241d* and *BCI*. On the other 4 data sets information regularization performs better than TSVM on all but *g241c*.

Table 6.2: Average error rates obtained by Support Vector Machine (supervised), Transductive Support Vector Machine, and Information Regularization, when trained on unlabeled data and 10 labeled samples.

	g241c	**g241d**	**Digit1**	**USPS**	**COIL**	**BCI**
SVM	47.32	46.66	30.60	20.03	68.36	49.00
TSVM	24.71	50.08	17.88	25.20	67.50	49.15
inforeg	41.25	45.89	12.49	17.96	63.65	50.21

Table 6.3: Average error rates obtained by Support Vector Machine (supervised), Transductive Support Vector Machine, and Information Regularization, when trained on unlabeled data and 100 labeled samples.

	g241c	**g241d**	**Digit1**	**USPS**	**COIL**	**BCI**
SVM	23.11	24.64	5.53	9.75	22.93	34.31
TSVM	18.46	22.42	6.15	9.77	25.80	33.25
inforeg	20.31	32.82	2.44	5.10	11.46	47.47

6.2 Discussion

We conclude that information regularization has the potential to improve significantly on supervised models. Its performance however does depend on the accuracy of the implicit semi-supervised bias that the method assumes. This is not a weakness of information regularization, but in fact a strength. The power of the algorithm lies in its capability of encoding into a semi-supervised prior a wide variety of assumptions, that can be intuitively customized to the task at hand.

While in classical supervised learning blind comparisons of classifiers lacking domain knowledge are sensible, because distribution-free classifiers exist and perform well, in semi-supervised learning domain knowledge is critical. Distribution-free semi-supervised learning cannot perform on average better than pure supervised learning. It is precisely the special form of the data distribution that correlates unlabeled data with labels, and permits the transfer of information.

Therefore we do not endorse a thorough head-to-head comparison of various semi-supervised learning algorithms in the absence of domain knowledge. Each method will perform well on the data on which the assumed semi-supervised prior is relevant.

6.3 Categorization of web pages

We demonstrate the performance of the information regularization framework on a web page categorization task. This is a natural semi-supervised learning problem, because the rate at which we can gather unlabeled web pages from the Internet is much higher than the rate with which people can categorize them. Therefore in practice we will always have a significant number of uncategorized web pages. Another feature that makes the domain suitable to semi-supervised learning is the rich structure of the web pages. We not only have information about the contents of each web page, but also about the hyperlinks among them. We can use the rich structure to define a relevant semi-supervised prior.

6.3.1 The data

We perform web page categorization on a variant of the WebKB dataset [25], that consists of 4199 web pages downloaded from the academic websites of four universities (Cornell, Texas, Washington, and Wisconsin). Each web page belongs to one of four topics, "course", "faculty", "project", or "student", and the goal is to label all pages with high accuracy.

We have processed each web page to keep only the text that appears on the page, as well as the text from other pages that appears under links pointing to this page (*anchor text*). The first step in processing the documents is to treat the body and link text as bag of words. Then we perform two independent feature selections, keeping only the 100 most predictive body words, and the 500 most predictive link words. We measure how predictive a word is by the reduction in entropy with respective to the class brought by the introduction of that word.

We represent each web page by two sparse vectors. The first vector gives the count of body words that appears in the web pages, for each of the 100 selected words. The other vector gives the count of the link words appearing in anchors pointing to the page, for each of the 500 selected link words.

We receive a limited amount of labeled data, and the task is to predict categories for the rest.

6.3.2 Supervised and semi-supervised classifiers

Naïve Bayes classifier

We begin by introducing a standard supervised classifier that performs well on text domains, the *Naïve Bayes* classifier. We will use the *Naïve Bayes* classifier both as a benchmark, and as component of the semi-supervised classifier.

The *Naïve Bayes* classifier is based on a generative model of the web page that assumes within each class the words that appear on a page are generated iid from a multinomial distribution. Let $\mathbf{x} = (x^1, x^2, \ldots, x^d)$ be the feature representation of a web page, where d is the size of the vocabulary, and x^i is the number of times word i appears in the page. Let

y be the category of the web page. Then we have the following generative model for the document:

$$P_{X,Y}(\mathbf{x}, y) = P_Y(y)P_{X|Y}(\mathbf{x}|y) = P_Y(y)L(|\mathbf{x}|)\prod_{i=1}^{d} P_{W|Y}(i|y)^{x^i} \tag{6.1}$$

where $|\mathbf{x}| = \sum_{i=1}^{d} x^i$ is the total length of the document, L is a probability distribution over the length of the document, and $P_{W|Y}(\cdot|y)$ is the word distribution specific to class y.

Given a fully-labeled training set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$, it is easy to estimate the parameters of the model, P_Y and $P_{F|Y}$, by maximizing the log-likelihood of the model:

$$\sum_{j=1}^{l} \log P_{X,Y}(\mathbf{x}_j, y_j) \tag{6.2}$$

The optimal parameters are given by:

$$P_{F|Y}(i|y) = \frac{\sum_{j=1}^{l} x_j^i \delta(y, y_j)}{\sum_{j=1}^{l} |\mathbf{x}_j| \delta(y, y_j)} \tag{6.3}$$

$$P_Y(y) = \frac{1}{l}\sum_{j=1}^{l} \delta(y, y_i) \tag{6.4}$$

In practice we smooth the maximum likelihood probabilities by adding an extra word that appears in every document with a very small count. This is equivalent to placing a prior on the parameters, and guarantees that the resulting probabilities will not vanish.

To classify a document we compute $P_{Y|X}(\cdot|\mathbf{x})$ via the Bayes rule, and choose the class label that maximizes it.

Note that we can build a supervised Naïve Bayes classifier based on body words, link words, or all the words.

Semi-supervised naïve Bayes

As discussed in Chapter 2, we devise a semi-supervised benchmark by extending the Naïve Bayes generative model to unlabeled data with the EM algorithm, as in [45]. The idea is to treat y as a latent variable for the unlabeled points. Given the previous labeled data points, and the unlabeled data $\{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\}$, we maximize the following log-likelihood:

$$\sum_{j=1}^{l} \log P_{X,Y}(\mathbf{x}_j, y_j) + \sum_{j=l+1}^{n} \log P_X(\mathbf{x}_j) \tag{6.5}$$

with the EM iterative algorithm.

Note that we have shown that running the EM algorithm is equivalent to an information regularization setup with a single region that contains all points, if the label distribution of the region is constrained to be Naïve Bayes.

The semi-supervised Naïve Bayes + EM can also be run on body features, link features, or the combination of the two.

Information regularization

We identify two types of semi-supervised biases that we encode in the information regularization algorithm:

1. The body of the web page is modeled by the Naïve Bayes model relatively well.

2. Web pages that have a word in common in some anchor are more likely to belong to the same category. This bias can be expressed for every word out of the vocabulary of 500 link words.

We formalize these biases by the set of regions $\mathcal{R}$ that defines the information regularization algorithm. The first type of bias needs a single region that covers all the points, R_0. The second type of biases requires one region for every link word, $R^i, i = 1 \ldots d_l$. The link region R^i contains all web pages that were linked to by at least one anchor that contained the word indexed by i. The bipartite graph of the information regularizer is depicted in Figure 6-1.

In order to express the first type of bias, the variable of interest of R_0 must be $Z = (X, Y)$. We restrict the family of distributions $Q_{Z|R}(\cdot|R_0)$ associated with region R_0 to Naïve Bayes distributions:

$$\mathcal{M}_{R_0} = \{Q_Z \,;\, Q_Z(\mathbf{x}, y) = Q_Y(y)Q_{X|Y}(\mathbf{x}|y) = Q_Y(y)L(|\mathbf{x}|)\prod_{i=1}^{d} Q_{W|Y}(i|y)^{x^i}\} \quad (6.6)$$

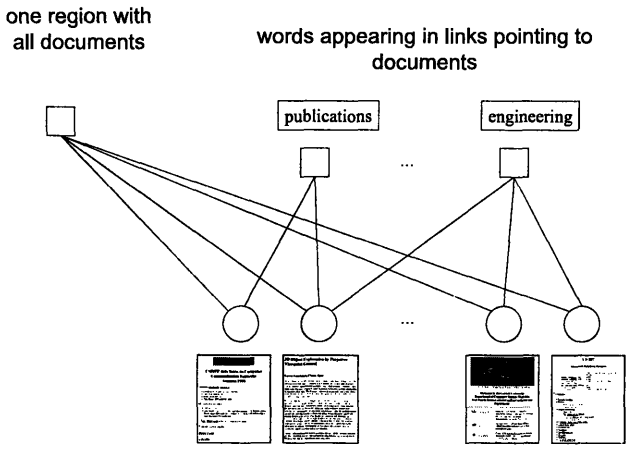

Figure 6-1: The bipartite graph structure of the information regularizer for categorization of web pages. A single region constrained to a Naïve Bayes model contains all the points. The rest of the regions correspond to words in the anchors of the web page.

The second type of regions hold distributions over the variable of interest Y[1] from unrestricted families $\mathcal{M}_{R^i}$.

We assign the weights π_R such that we can trade off the relative value between the two types of regions. We express this trade-off by a parameter $\eta \in [0, 1]$. If η is 0, we only rely on the region of type 1 and its model. If η is 1, we only rely on the regions of type2. Otherwise, regions of type 2 are weighted the same among themselves.

Because the single region of type 1 emulates Naïve Bayes and EM exactly, setting $\eta = 0$ is equivalent to Naïve Bayes + EM run on the body features. Also, we can predict that when $\eta = 1$ the performance is poor, because the link features are sparse and without R_0 they leave the graph disconnected.

This setup of the information regularizer results in the following algorithm:

1. Assign distributions $Q_{Y|R}(\cdot|R^i)$ to every region R^i. Assign a distribution $Q_{XY|R}(\cdot|R_0\,;\,\theta)$ to the region of type 1.

[1]We could write the specification of the information regularizer such that all regions have the same variable of interest (X, Y). We opt for different variables of interest only to simplify the argument – the approaches are equivalent.

2. Initialize $P_{Y|A}(y|\alpha),\ \alpha \in \mathcal{A}$, α unlabeled, with uniform distributions.

3. **Region update:**

   - Recompute the Naïve Bayes parameters θ by maximum likelihood on a data set on which $(\mathbf{x}_\alpha, y)$ appears according to $P_{Y|A}(y|\alpha)$. This is the M step of Naïve Bayes + EM
   - Recompute all region distributions for type 2 regions by averaging point distributions:

$$Q_{Y|R^i}(y|R^i) \leftarrow \sum_{\alpha \in \mathcal{A}} \pi_{A|R^i}(\alpha|R^i) P_{Y|A}(y|\alpha)$$

4. **Point update:** Recompute all point distributions of unlabeled points by geometrically averaging region distributions:

$$P_{Y|A}(y|\alpha) \leftarrow \frac{1}{\tilde{Z}} \exp\left((1-\eta) \log Q_{Y|R}(y|R_0\,;\,\theta) + \eta \sum_{R \in \mathcal{R}, R \neq R_0} \pi_{R|A}(R|\alpha) \log Q_{Y|R}(y|R) \right)$$

5. Return to Step 3 until convergence.

6.3.3 Results

Table 6.4 shows a comparison of the supervised Naïve Bayes, the semi-supervised Naïve Bayes + EM, and the information regularization algorithm, for various sizes of the labeled training data. Each error rate is obtained as an average over 50 random selections of the labeled data. All results use $\eta = 0.9$.

Information regularization achieved between 1% and 3% error rate improvement over any of the semi-supervised algorithms. Note that when the number of labeled samples becomes sizable, supervised naïve Bayes outperforms information regularization, which is to be expected because semi-supervised algorithms are usually more sensitive to model mismatch, and may loose their advantage when labeled data is enough to train the supervised model well.

In Figure 6-2 we show the performance of information regularization as a function of η, averaged over 50 runs, for 25 labeled training points. We can see significant improvement over the purely supervised method (that does not depend on η, as well as a gradual

Table 6.4: Error rates of Naïve Bayes, the semi-supervised Naïve Bayes + EM, and the information regularization on the web page categorization data. Each result is obtained as an average of 50 runs with random sampling of the labeled training data.

	number of labeled samples							
	10	20	40	80	160	320	640	1280
inforeg	18.33	16.29	16.35	16.34	16.15	15.67	15.08	13.84
nb + EM (body)	22.48	19.94	19.93	19.86	19.56	19.13	18.68	17.41
nb + EM (link)	60.53	60.58	60.71	60.63	60.60	59.61	43.90	43.62
nb + EM (body + link)	20.93	20.03	19.92	19.69	19.08	18.13	17.25	15.48
NB (body)	32.68	25.38	20.95	18.14	16.66	15.35	15.07	14.52
NB (link)	57.01	57.49	55.22	53.06	50.72	48.63	47.09	45.44
NB (link + body)	31.67	24.36	19.66	17.07	15.59	14.18	13.74	12.84

improvement over the semi-supervised NB + EM. If η is very large, performance drops significantly because connectivity breaks.

6.4 Semi-supervised named entity recognition

We apply information regularization to a *named entity recognition* task on the data published in [17]. In named entity recognition, the goal is to identify the category of each proper name in a document based on the spelling of the proper name, and the context around it. For example, if the entity begins with *Mr.*, we have a strong reason to believe it names a person; and if the words preceding the entity are *city of*, it is very likely that it names a location. Identifying such specific rules that depend on the actual words contained in the entity and around it seems to require a large number of labeled training samples, as the number of possible rules is very large. We show that even with a limited labeled training set, if we have enough unlabeled data we can achieve low error rates by placing a reasonable semi-supervised prior through information regularization.

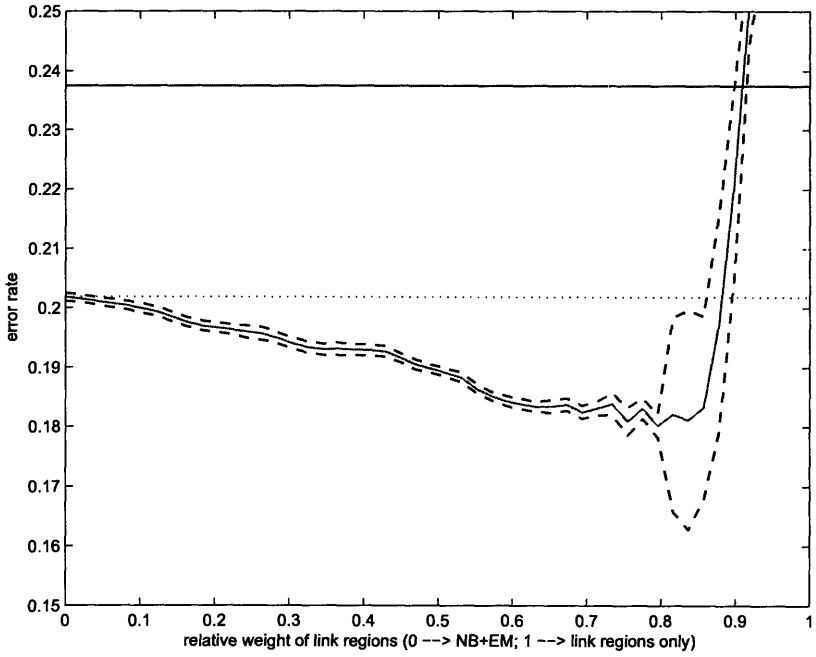

Figure 6-2: Average error rate of information regularization on the WebKB web page categorization task, as a function of η. $\eta = 0$ is equivalent to naïve Bayes + EM semi-supervised learning, while $\eta = 1$ uses only the link regions. The dotted lines indicate one standard deviation variation on the 47 experiments. The horizontal line is the error rate achieved by supervised naïve Bayes. There were 25 labeled samples.

The idea is to define information regularization regions based on context and spelling features of the entities, such as the words that make up the entity, its capitalization, or the words that modify it. Therefore we consider a one-to-one mapping between features and information regularization regions, and entities belong to a region if they have the corresponding feature enabled. We consider a large number of features, and provide an algorithm for selecting the ones that are relevant for information regularization. The mechanism by which we select regions is by controlling πR, the weight of each region relative to the others.

6.4.1 The data

The task consists of classifying the named entities gathered from $971,746$ sentences from New York Times. The named entity extraction has already been performed by a statistical parser [17], and we only need to assign each entity to one of four categories: *person*, *organization*, *location*, and *other*. The parser extracted a total of $90,305$ entities. Out of these 1000 entities have been labeled by human annotators, and they will be our test set, while the rest of the $89,305$ unlabeled entities are the unlabeled training set. Note that none of the labeled entities will be used during training; instead we will get our label information for training from a set of labeled rules, as described below.

In principle we could also use the remaining 1000 test entities as unlabeled data during training, without violating the separation between the training and the testing procedures. We keep the unlabeled features of the test data entirely separate because we want to illustrate that information regularization can depart from the transductive paradigm.

We extract various features about each named entity, on which we will base our classifier. Each feature is a binary property that can either be present or absent. In other words, we represent each entity by the list of features that are enabled for the entity.

There are several types of extracted features, pertaining to the spelling of the entity, or to its context, as listed below [17]:

- **The exact words in the entity**: each entity has a property identified with its exact spelling exact spelling, so that multiple instances of the same entity in different

contexts share this property.

- **Individual words in the entity**: we label each entity by the list of words contained in it.
- **All capitals**: this feature is on if the entity consists only of capital letters (such as IBM).
- **All capitals or periods**: this feature is on if the entity consists only of capital letters and periods, with at least one period (I.B.M.)
- **Non-alphanumeric characters**: the word obtained by removing all letters from the entity (A.T.&T. $\rightarrow$ ..&.).
- **Context words**: we attach to each entity a property that consists of the context words that modify the entity, obtained from the parser.
- **Context type**: *prepositional* or *appositive*, depending on how the context words modify the entity in the parse.
- **Temporal entity**: this type of feature contains a single label that is on for entities that contain a day of week, or the name of a month among its words.

We have extracted a total of $68,796$ features, but only $25,674$ of them are enabled for at least two entities. Features that are not enabled for at least two entities do not affect the running time of the information regularization algorithm, because they do not participate in the exchange of messages (if we remove them the labels of all points will converge to the same values).

There is no labeled training data in the form of labeled entities, but we do have a set of eight hand-labeled features that we know are indicative of the category of those entities that have the feature enabled. The training labeled features are shown in Table 6.5.

6.4.2 Information regularization approach

We provide a classification algorithm for named entities by information regularization. Specifically, the algorithm is based on the discrete version of information regularization

Table 6.5: Seed features used as training labels for semi-supervised named entity recognition.

feature	**category**
entity is New-York	Location
entity is U.S.	Location
entity is California	Location
entity contains Mr.	Person
entity contains Incorporated	Organization
entity is Microsoft	Organization
entity is I.B.M.	Organization
temporal entity	Other

introduced in Chapter 5.

Each feature potentially introduces a label-similarity bias of all the entities that have that feature enabled. To take into account these similarity biases we define a bipartite graph whose nodes on one side are all the entities, and on the other side are all features (Figure 6-3). We join an entity with a feature if that entity is enabled for the feature. There are $90,305$ points (the entities), and $68,796$ regions (the features); however, only $25,674$ regions are non-degenerate and contain at least two points.

Not all features are created equal in terms of the encoded similarity bias. For example, while all entities with the feature "entity is Microsoft" enabled are likely to be names of organizations, the feature "prepositional", shared by half of the entities, clearly does not correlate with the label – it is unlikely that the majority of entities among half the examples have the same label. Thus the relative weighting π_R of the features is very important for the regularizer to perform. We have no choice but to tackle the famous *region selection* problem. Other than that, we can safely assume that the points (entities) within a region (enabled feature) are to be treated equally – $\pi_{A|R}(\cdot|R)$ is a uniform distribution, for every R.

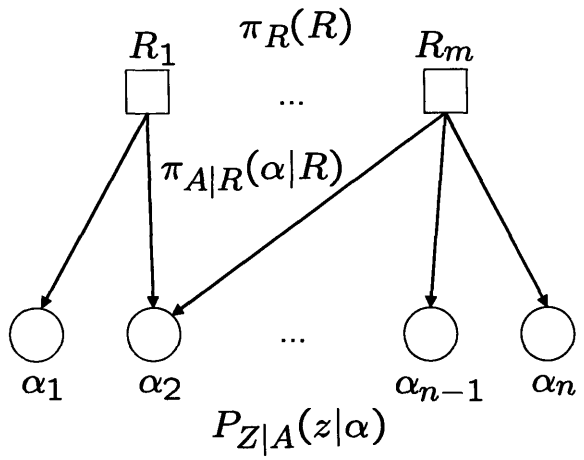

Figure 6-3: Graph representation of the information regularization model. Each circle is an instance of a named entity (90,305 circles), and each square is a feature (25,674 that contain at least 2 entities). An edge means that the entity has the feature enabled. (This is the same as Figure 5-1, reproduced here for convenience.)

6.4.3 Region selection

In the generic analysis of the information regularization algorithm from the previous chapter we avoided the question of selecting the regions of the information regularizer, as the type of regions that works with a task is a property of the domain, and should ideally be learned from more than one task, in order to generalize. The ideal set of regions for a single task would always be the regions that group together all points of a particular class. However, we cannot find these regions without solving the classification problem in the first place. At best we can select a set of regions based on a sensible criterion that we think it correlates regions with labels as well as possible.

In this named entity recognition task, the assumption we make is that the set of regions (and associated weights) that best describes the task is the one that results in soft labels $P_{Y|A}$ of minimal entropy. The smaller the entropy of the soft labels, the more precise they are, and the more confident we are in the true category of the entities. Thus if $P^*_{Y|A}(\pi_R)$ is the set of labels that minimizes the information regularizer for a particular region weighting

π_R, we choose the weighting that minimizes the average entropy of the points:

$$\pi_R^* = \arg\min_{\pi_R} \sum_{\alpha \in \mathcal{A}} \sum_{y \in \mathcal{Y}} -P^*_{Y|A}(\pi_R)(y|\alpha) \log P^*_{Y|A}(\pi_R)(y|\alpha) \tag{6.7}$$

$$P^*_{Y|A}(\pi_R) = \arg\min_{P_{Y|A}} \sum_{R \in \mathcal{R}} \min_{Q_{Y|R}(\cdot|R) \in \mathcal{M}_R} \sum_{\alpha \in \mathcal{A}} \pi_R(R)\pi_{A|R}(\alpha|R)\mathrm{KL}\left(P_{Y|A}(\cdot|\alpha) \,\|\, Q_{Y|R}(\cdot|R)\right) \tag{6.8}$$

In the computation of entropy, we assign a uniform label distribution to points that are not covered by any region. Therefore uncovered entropy have the highest entropy, thus the set of regions of minimal global entropy will likely cover all points. If a point remains uncovered and there exists a region that

- has not been selected
- covers the point
- is linked to at least one labeled training region by a path of selected regions that overlap

than we could further reduce the entropy by selecting this region with an infinitesimal weight π_R.

Greedy approximation

Optimal region selection according to the above entropy criterion is expensive from a computational perspective, given the large number of points and potential regions. We resort to an efficient greedy approximation. The idea is to start from a minimal set of seed regions (those for which we have labels), and enlarge $\mathcal{R}$ incrementally while updating the weights of the existing regions, so that each operation minimizes the entropy greedily.

Suppose that given a set of weights π_R and a region R_0 we can compute $g(R_0, \pi_R)$, the optimal value of $\pi_R(R_0)$ while keeping $\pi_R(R)$ fixed[2] for all $R \neq R_0$. Then we can run the following greedy algorithm to find a good set of region weights:

[2]We let $\sum_{R \in \mathcal{R}} \pi_R(R)$ be unconstrained. The only restriction on the weights is that they are non-negative.

1. $\mathcal{S}$ will be the of regions $R \in \mathcal{R}$ of positive $\pi_R(R)$. We initialize $\mathcal{S}$ to be the set of labeled training regions. We fix $P_{Y|R}(y|R) = \delta(y, \hat{y}_R)$, the training label. We also fix $\pi_R(R) = \infty$ for the labeled regions (in other words, we trust the labeled regions completely – if an entity belongs to some labeled regions, its label distribution will be determined only from the labeled regions to which it belongs.).

2. For each $R_0 \in \mathcal{R} \setminus \mathcal{S}$ compute $g(R_0, \pi_R)$, as well as the resulting drop in entropy with the addition of region R_0 with the computed weight. Add to $\mathcal{S}$ the k regions that achieve the highest entropy drop, where k is specified in advance.

3. For each $R \in \mathcal{S}$ that is not a labeled training region update

$$\pi_R(R) \leftarrow g(R, \pi_R) \tag{6.9}$$

4. Repeat from 2. until all points in $\mathcal{A}$ are covered by at least one region, the cardinality of $\mathcal{S}$ is sufficiently large, and the update in Step 2. indicates that the weights have converged (early stopping for computational savings is also OK).

The above greedy algorithm, that starts from a small set of rules and expands it incrementally, is likely to perform better and be faster than an algorithm that starts from all rules, and keeps removing them. This is because we initialize with relevant regions, and we only add regions that are relevant given relevant regions. Thus $\mathcal{S}$ is should stay relevant as it increases in size. On the other hand, starting from all regions means that most of them will be irrelevant in the beginning, and the greedy judgment is noisy and questionable.

Note that if set of possible regions covers all entities, than the described greedy iteration will eventually produce a set of regions $\mathcal{S} \subset \mathcal{R}$ that also covers all entities. This is because it is always beneficial to add a region with new points if its weight is small enough. When the weight approaches 0 it will have no effect on the points that are already covered, but it will move $P_{Y|A}(\cdot|\alpha)$ for the newly covered points away from the uniform default, decreasing the overall entropy.

Computation of the optimal weight of a single region

Unfortunately, even the computation of the optimal weight of a single region given that the weights of all other regions are fixed is not efficient enough. According to the blueprint of the greedy algorithm, the computation of $g(R_0, \pi_R)$ must be performed for each $R_0 \in \mathcal{R}$ at every greedy iteration. Thus each addition of k regions to $\mathcal{S}$ involves potentially $25,674^3$ evaluations of $g(R_0, \pi_R)$. We do not have a choice but to restrict the computation of $g(R_0, \pi_R)$ to a fast approximation.

Let us analyze what the exact evaluation of $g(R_0, \pi_R)$ would entail. According to equation (6.8), computing $P^*_{Y|A}(\pi_R)$, the optimal labels of all entities, involves a full run of the information regularization iterative algorithm described in the previous chapter. Every slight change in π_R, even in the weight of a single region, involves running the information regularization on all regions again, because the weight of a single region affects all labels. Thus evaluating a single $g(R_0, \pi_R)$ exactly is necessarily less efficient than information regularization. Then clearly we cannot evaluate $g(R_0, \pi_R)$ exactly $25,000$ times, for every couple of regions we would add to $\mathcal{S}$.

The key to making the evaluation of $g(R_0, \pi_R)$ efficient is to break the dependency between $\pi_R(R_0)$ and the labels of all points and all regions. Therefore, we make the following important approximation: we assume that only $P_{Y|R}(\cdot|R_0)$ and $P_{Y|A}(\cdot|\alpha)$ for $\alpha \in R_0$ are allowed to vary. The labels of all other points and regions are held constant. We then optimize the average entropy of the labels as a function of the weight of region R_0. Thus we use the following approximation of $g(R_0, \pi_R)$

$$g(R_0, \pi_R) \approx \arg \min_{\pi_R(R_0)} \sum_{\alpha \in R_0} \sum_{y \in \mathcal{Y}} -P^*_{Y|A}(\pi_R(R_0))(y|\alpha) \log P^*_{Y|A}(\pi_R(R_0))(y|\alpha) \quad (6.10)$$

where

$$P^*_{Y|A}(\pi_R(R_0)) = \arg \min_{P_{Y|A}(\cdot|\alpha), \alpha \in R_0,\ P_{Y|R}(\cdot|R_0)} \sum_{R \in \mathcal{R}} \min_{Q_{Y|R}(\cdot|R) \in \mathcal{M}_R} \sum_{\alpha \in \mathcal{A}} \pi_R(R) \pi_{A|R}(\alpha|R) \mathrm{KL}\left(P_{Y|A}(\cdot|\alpha) \,\|\, Q_{Y|R}(\cdot|R)\right) \quad (6.11)$$

[3] In practice we can save computation by considering as candidates only the regions that have at least one element in common with already selected regions.

Note that we only only need to evaluate the average label entropy on R_0, because all other entity labels are held constant, thus the only variation in global average entropy comes from the labels of points from R_0.

The computation of $P^*_{Y|A}(\pi_R(R_0))$ now involves only the points in R_0 and it is much faster to carry out than running the full information regularization. In particular, for a specific $\pi_R(R_0)$ the information regularization iteration for computing reduces to the following set of updates:

$$P_{Y|R}(y|R_0) = \frac{1}{|R_0|} \sum_{\alpha \in R_0} P_{Y|A}(y|\alpha) \tag{6.12}$$

$$\log P_{Y|A}(y|\alpha) = \frac{\pi_R(R_0)}{\pi_R(R_0) + t(\alpha)} \log P_{Y|R}(y|R_0) + \frac{C(y,\alpha)}{\pi_R(R_0) + t(\alpha)} + \text{const.} \tag{6.13}$$

where $t(\alpha)$ and $C(y,\alpha)$ are constants determined from the neighbors of R_0 according to:

$$C(y,\alpha) = \sum_{R \ni \alpha,\ R \neq R_0} \frac{|R_0|}{|R|} \pi_R(R) \log P_{Y|R}(y|R) \tag{6.14}$$

$$t(\alpha) = \sum_{R \ni \alpha,\ R \neq R_0} \frac{|R_0|}{|R|} \pi_R(R) \tag{6.15}$$

and the constant const. is such that for each α, $P_{Y|A}(y|\alpha)$ sums to 1 over $\mathcal{Y}$.

For a fixed $\pi_R(R_0)$ we can perform the iterations (6.12) and (6.13) until convergence, to find $P^*_{Y|A}(\pi_R(R_0))$. Then we can evaluate the entropy over R_0, that need to be minimized as a function of $\pi_R(R_0)$. There are many ways of minimizing this objective, including gradient descent, and Newton's method. We opt for a simple line binary search over the values of $\pi_R(R_0)$. Since the objective is neither convex, nor monotonic, we need extra care not to be trapped in a poor local optimum.

It is worth understanding what is the label configuration if $\pi_R(R_0)$ takes extreme values. If $\pi_R(R_0) \to 0$, then region R_0 has no impact whatsoever on points from R_0 that belong to other enabled regions also. It is as if we run information regularization without R_0 to assign labels to points from R_0 that belong to other regions. Than we set $P_{Y|R}(\cdot|R_0)$ to the average label of those points. Then we copy $P_{Y|R}(\cdot|R_0)$ to the labels of the points that are only covered by this region. The change in average entropy comes only from setting the labels of points unique to R_0.

On the other hand, if $\pi_R(R_0) \rightarrow \infty$, then we assign complete confidence to region R_0. At convergence, the label distributions of all $\alpha \in R_0$, as well as $P_{Y|R}(\cdot|R_0)$ will all be equal. The configuration is equivalent to a situation in which all points from R_0 collapse to a single point, that belongs to all regions that intersected R_0. The label of that point is set by geometric averaging, while the label of R_0 is set to the label of the point.

6.5 Results

We compare the performance of the information regularization algorithm with region selection on the named entity recognition task with the error rates obtained by Yarowsky [58] and Collins at al [17], shown in Table 6.6. The baseline is a supervised decision list classifier:

$$\hat{y}(\mathbf{x}) = \arg \max_{R \ni \mathbf{x}, y \in \mathcal{Y}} \frac{Count(R, y) + \epsilon}{Count(R) + |\mathcal{Y}|\epsilon} \tag{6.16}$$

where $Count(R, y)$ is the number of labeled training entities of observed class y present in region R, and $Count(R)$ is the total of labeled training entities in R. $\epsilon = 0.1$ is a smoothing parameter. In other words, the decision list classifier estimates a label distribution for each region based only on the labeled data, and assigns labels to other entities according to a maximum rule.

The Yarowsky and Collins algorithms are described in [17], and consist of the application of the supervised decision list classifier on a training set generated by labeling the unlabeled data incrementally starting at the entities with known labels, and iteratively propagating across the regions. Collins separates the spelling and context features, and propagates on the two sets of features alternatively, in the spirit of co-training.

We ran information regularization by greedily adding 20 regions at a time. The labeled regions had their weight fixed to infinity, so that if an entity belongs to some labeled region, its label will be determined solely from the labeled regions to which it belongs. The optimal weights of the other regions were computed by binary search on the interval bounded by 0 and twice the maximum of the weights of other unlabeled regions. Figure 6-4 shows the performance of the information regularization with region selection algorithm, that achieves an error rate of 14% at 2000 regions. The comparison with Collins' algorithm

Table 6.6: Error rates on the named entity recognition task.

Algorithm	**Error rate**
Supervised Decision List	0.54
Semi-supervised EM	0.17
Yarowsky [58]	0.19
Collins et al [17]	0.09
Information Regularization	0.14

is somewhat unfair because we categorize the entities into 4 classes, while the other algorithms categorize them only into 3 classes, by omitting the "other" class altogether. This is done by training only on 3 classes, and excluding the test points labeled with "other" from the computation of error rates.

From what we observed the main source of error in the information regularization run was the incorrect treatment of location features, such as "Japan", or "York". These features, along with many other location features, were labeled as "organization" features. The reason is that some generic features that contained many entities, such as ".." or "ALLCAPS", appeared mostly in organizations, and they were labeled as such. However, some location entities also contained "..", so that the "organization" label propagates to some "location" entities that were of unknown label at the time. Once a few "location" entities were labeled as "organization", the wrong label quickly propagated to the entire cluster.

Once possible remedy to the source of error described above is to artificially weight less large regions for which we decided their label based on a small number of entities contained in them, because they likely span many classes and are likely to hurt if weighted to much. Just because the 10% of the entities for which we have some label information from a large region seem to share the same label does not mean that all points in the region should share that label. However, if the region is small and 90% of its points share the same label, then it is likely that the region is a good indicative of the label.

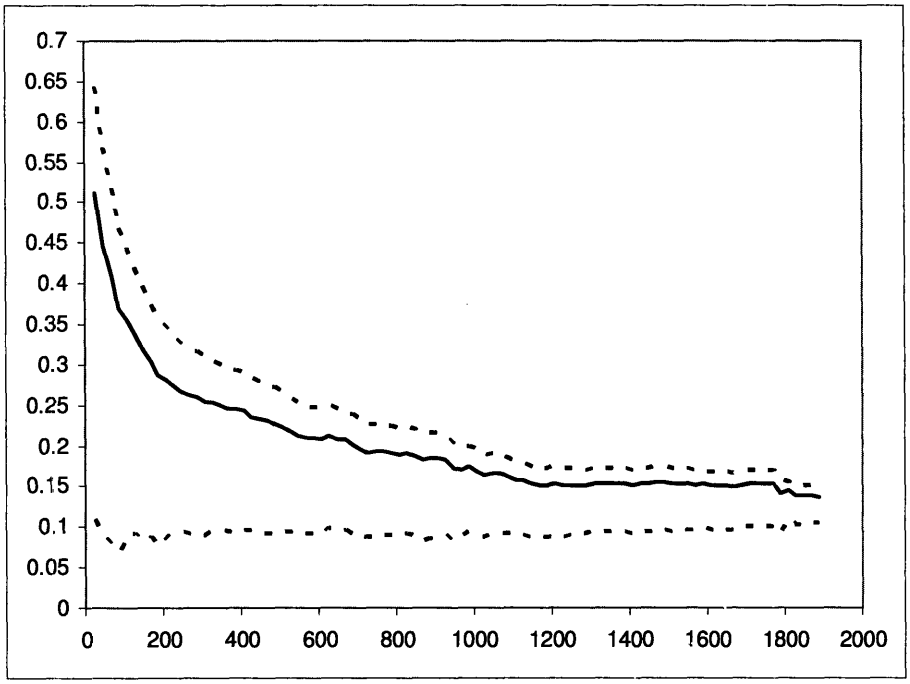

Figure 6-4: Error rate of information regularization on the named entity recognition task as a function of the number of regions included in the regularizer. Bottom dotted line: error rate on entities covered by at least one region. Top dotted line: error rate on all entities, by treating all uncovered entities as errors. Solid line: estimated error rate by selecting labels of uncovered entities uniformly at random.

Chapter 7

Contribution

We introduced a framework for semi-supervised learning based on the the principle of information regularization, originally described in [53]. A central concept in information regularization is that of a semi-supervised bias, an unlabeled subset of the training set that consists of objects deemed to be similar in a way that is relevant to classification. Information regularization represents the semi-supervised biases by a collection of regions that covers the training data, and a probability distribution over the selection of the regions. The regions can be defined from a similarity metric on the vector space of features, or from relations among the objects.

Given a set of regions, the framework defines an information regularizer that penalizes joint distributions that do not satisfy the similarity biases of individual regions. The regularizer can be applied to a supervised loss function to obtain an objective whose minimization results in semi-supervised classification.

We demonstrated the convexity of the information regularization objective, and provided an iterative message passing algorithm on the bipartite graph of objects and regions that optimizes the criterion.

We showed that the information regularization algorithm can be applied in both a purely non-parametric setting, and in a situation in which we enforce parametric constraints on the joint.

When the feature space is continuous, we obtained an inductive classification algorithm by taking the limit of the information regularizer when the number of regions is infinite,

and their size approaches 0.

The information regularization framework is flexible enough to subsume the expectation maximization algorithm for semi-supervised learning, by defining a single region that contains all points, with a specially restricted region distribution. It can also obtain the objective of harmonic graph regularization by defining regions with pairs of points, one region for each edge in the graph. A variant of co-training can be achieved with the information regularization objective.

We demonstrated the performance of information regularization on categorization of web pages, and on named entity recognition.

7.1 Topics of further research

An important issue with most current semi-supervised algorithms is that they do not address well learning the semi-supervised biases. In the context of information regularization this translates into learning the region set, and the region weights. In other semi-supervised algorithms it may mean learning a label-similarity metric. We made an attempt on learning the region set in the context of named entity recognition, but the resulting algorithm is largely heuristic. There is a need for a thorough treatment of the topic. We envision that in order to learn the similarity metric reliably one would need a collection of tasks (training sets) from the domain on which the metric should be valid.

Appendix A

Notation

The following is a list of symbols defined in the thesis and their meaning:

$\mathcal{A}, \alpha$	the set of all data points available to the semi-supervised learning algorithm. α is a generic element of $\mathcal{A}$
$\mathbf{x}, \mathcal{X}$	the *feature vector representation* of a data point, and the set of all possible feature vectors ($\mathbf{x} \in \mathcal{X}$)
$\mathbf{x}_\alpha$	feature representation of the object α
$\mathcal{Y}, y$	the set of possible *class labels*, and a generic label ($y \in \mathcal{Y}$)
y_α	class label of the object α
$z, \mathcal{Z}$	quantity to be predicted about each data point in the most general framework; for example z may be y or $(\mathbf{x}, y)$
$\mathcal{R}, R$	a collection of *regions* from the set of available objects. $R \in \mathcal{R}$ is a generic region. Note that $R \subset \mathcal{A}$
$\boldsymbol{\theta}, \Theta$	*parameter vector*, and the set of all parameters
$\mathcal{D}$	$= \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is a semi-supervised *training set* of n objects, of which the first l are labeled, and the next u are unlabeled. Thus $(\mathbf{x}_{\alpha_i}, y_{\alpha_i})$ is observed for $1 \leq i \leq l$, and $\mathbf{x}_{\alpha_j}$ is observed for $l < j \leq n$. Note that $\mathcal{D} \subset \mathcal{A}$ and $n = l + u$.
π_R	probability distribution over $\mathcal{R}$. It represents the relative importance of the regions in the regularizer. In a typical setting it is given a priori.

$\pi_{A|R}$ probability distribution over the elements of region R. It represents the relative contribution of each element to the model associated with R. In a typical setting it is given a priori.

$P_{Z|A}$ probability distribution associated with each data point that needs to be estimated. It represents the a posteori confidence in the value of quantity z for data point α.

P_A the probability distribution that generates data points. It can be estimated from observed unlabeled samples.

$\mathcal{F}$ constrained family of distribution to which $P_{Z|A}$ is forced to belong. It represents any hard constraints on $P_{Z|A}$ known a priori.

$Q_{Z|R}$ probability distribution associated with each region that represents the a posteriori confidence in the quantity z on average across the region.

$\mathcal{M}_R$ constrained family of distributions over $\mathcal{Z}$ associated with region R. In other words, $Q_{Z|R}$ is forced to belong to $\mathcal{M}_R$. The family encodes the semi-supervised bias over $P_{Z|A}$ induced by region R.

$P_{\mathcal{F}}$ the *task prior*. It is a distribution over $\mathcal{F}$ that encodes a priori biases about the possible $P_{Z|A}$. Any semi-supervised method assumes implicitly or explicitly a task prior. $P_{\mathcal{F}}$ is specific to the class of problems, but not on the particular instance of the problem we need to solve.

Index

Bibliography

[1] Steven Abney. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30(3), 2004.

[2] Shun-Ichi Amari. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9):1379–1408, 1995.

[3] Maria-Florina Balcan, Avrim Blum, Pakyan Choi, John Lafferty, Brian Pantano, Mugizi Robert Rwebangira, and Xiaojin Zhu. Person identification in webcam images: An application of semi-supervised learning. In *ICML2005 Workshop on Learning with Partially Classified Training Data*, 2005.

[4] Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 89–96. MIT Press, Cambridge, MA, 2005.

[5] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In John Shawe-Taylor and Yoram Singer, editors, *Proceedings of the 17th Annual Conference on Learning Theory*, volume 3120, pages 624–638. Springer, 2004.

[6] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning, Special Issue on Clustering*, 56:209–239, 2004.

[7] James O. Berger. *Statistical decision theory and bayesian analysis*. Springer, 2nd edition edition, 1985.

[8] Richard E. Blahut. Computation of channel capacity and rate distortion functions. In *IEEE Transactions on Information Theory*, volume 18, pages 460–473, July 1972.

[9] Arvim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Conference on Computational Learning Theory*, pages 92–100, Madison, WI, 1998.

[10] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pages 18–26, 2001.

[11] Avrim Blum, John Lafferty, Mugizi Robert Rwebangira, and Rajashekar Reddy. Semi-supervised learning using randomized mincuts. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 13, New York, NY, USA, 2004. ACM Press.

[12] Ulf Brefeld and Tobias Scheffer. Co-em support vector learning. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 16, New York, NY, USA, 2004. ACM Press.

[13] Chris J.C. Burges and John C. Platt. Semi-supervised learning with conditional harmonic mixing. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-supervised learning*. MIT Press, Cambridge, MA, to appear 2006.

[14] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-supervised Learning*. MIT Press, to appear 2006.

[15] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 585–592. MIT Press, Cambridge, MA, 2003.

[16] Ira Cohen, Fabio G. Cozman, Nicu Sebe, Marcelo C. Cirelo, and Thomas S. Huang. Semi-supervised learning of classifiers: theory, algorithms for Bayesian Network

Classifiers and application to Human-Computer Interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1553–1567, December 2004.

[17] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 189–196, 1999.

[18] D. Cooper and J. Freeman. On the asymptotic improvement in the outcome of supervised learning provided by additional nonsupervised learning. *IEEE Transactions on Computers*, C-19:1055–1063, 1970.

[19] Adrian Corduneanu. Stable mixing of complete and incomplete information. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2002.

[20] Adrian Corduneanu and Tommi Jaakkola. Continuation methods for mixing heterogeneous sources. In Adnan Darwiche and Nir Friedman, editors, *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 111–118. Morgan Kaufmann, 2002.

[21] Adrian Corduneanu and Tommi Jaakkola. On information regularization. In Christopher Meek and Uffe Kjærulff, editors, *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, pages 151–158. Morgan Kaufmann, 2003.

[22] Adrian Corduneanu and Tommi Jaakkola. Data-dependent regularization. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-supervised learning*. MIT Press, Cambridge, MA, to appear 2006.

[23] Adrian Corduneanu and Tommi S. Jaakkola. Distributed information regularization on graphs. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 297–304. MIT Press, Cambridge, MA, 2005.

[24] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[25] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the AAAI-98, 15th Conference of the American Association of Artificial Intelligence*, pages 509–516, Madison, US, 1998. AAAI Press, Menlo Park, US.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[27] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1300–1309, Stockholm, Sweden, 1999.

[28] Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707, 2002.

[29] Gad Getz, Noam Shental, and Eytan Domany. Semi-supervised learning — a statistical physics approach. In *Proceedings of the 22nd ICML Workstop on Learning with Partially Classified Training Data*, Bonn, Germany, 2005.

[30] Zoubin Ghahramani and Michael Jordan. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems*, volume 6, pages 120–127, 1994.

[31] Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, pages 327–334. Morgan Kaufmann, San Francisco, CA, 2000.

[32] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[33] Jiayuan Huang. A combinatorial view of graph laplacians. Technical Report 144, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, August 2005.

[34] Thorsten Joachims. Transductive inference for text classification using support vector machines. In Ivan Bratko and Saso Dzeroski, editors, *Proceedings of 16th International Conference on Machine Learning*, pages 200–209, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.

[35] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference of Machine Learning (ICML)*, 2003.

[36] Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. In S. J. Hanson, G. A. Drastal, and R. L. Rivest, editors, *Computational Learning Theory and Natural Learning Systems, Volume I: Constraints and Prospect*, volume 1. MIT Press, Bradford, 1994.

[37] Charles Kemp, Thomas L. Griffiths, Sean Stromsten, and Joshua B. Tenenbaum. Semi-supervised learning with trees. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[38] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[39] Balaji Krishnapuram, David Williams, Ya Xue, Alexander Hartemink, Lawrence Carin, and Mario Figueiredo. On semi-supervised classification. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 721–728. MIT Press, Cambridge, MA, 2005.

[40] T. N. Lal, M. Schröder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf. Support vector channel selection in BCI. *IEEE Transactions on Biomedical Engineering*, 51(6):1003–1010, 2004.

[41] C. McDiarmid. On the method of bounded differences. In *Survey in Combinatorics 1989*, pages 148–188. Cambridge University Press, Cambridge, UK, 1989.

[42] Atsuyoshi Nakamura and Naoki Abe. Polynomial learnability of stochastic rules with respect to the KL-divergence and quadratic distance. *IEICE Transactions on Information and Systems*, E84-D(3):299–316, March 2001.

[43] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-006-96, Columbia University, February 1996.

[44] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93, New York, NY, USA, 2000. ACM Press.

[45] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[46] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI '99/IAAI '99: Proceedings of the 16th national conference on Artificial intelligence and the 11th Innovative applications of artificial intelligence conference*, pages 474–479, Menlo Park, CA, 1999. American Association for Artificial Intelligence.

[47] Charles Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, volume 1, pages 29–36, January 2005.

[48] Lawrence K. Saul, Kilian Q. Weinberger, Jihun H. Ham, Fei Sha, and Daniel D. Lee. Spectral methods for dimensionality reduction. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-supervised learning*. MIT Press, Cambridge, MA, to appear 2006.

[49] Dale Schuurmans and Finnegan Southey. An adaptive regularization criterion for supervised learning. In *ICML '00: Proceedings of the Seventeenth International Confer-*

ence on Machine Learning, pages 847–854, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[50] Behzad M. Shahshahani and David A. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, September 1994.

[51] Chris Stauffer. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[52] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 945–952, Cambridge, MA, 2002. MIT Press.

[53] Martin Szummer and Tommi Jaakkola. Information regularization with partially labeled data. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1025–1032. MIT Press, Cambridge, MA, 2003.

[54] Martin Szummer and Tommi S. Jaakkola. Kernel expansions with unlabeled examples. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 626–632. MIT Press, 2001.

[55] Ben Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, 2002.

[56] Ben Taskar, Eran Segal, and Daphne Koller. Probabilistic classification and clustering in relational data. In Bernhard Nebel, editor, *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, pages 870–878, Seattle, US, 2001.

[57] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[58] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[59] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[60] Dengyong Zhou, Bernhard Schölkopf, and Thomas Hofmann. Semi-supervised learning on directed graphs. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1633–1640. MIT Press, Cambridge, MA, 2005.

[61] Xiaojin Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, May 2005.

[62] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic function. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20, pages 912–919, 2003.

[63] Xiaojin Zhu, Jaz Kandola, Zoubin Ghahramani, and John Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1641–1648. MIT Press, Cambridge, MA, 2005.

[64] Xiaojin Zhu and John Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In Luc De Raedt and Stefan Wrobel, editors, *Proceedings of the 22nd International Machine Learning Conference*. ACM Press, 2005.