Semi-Supervised Learning with Multiple Views

by

David Stuart Rosenberg

B.S. (Yale University) 2000
S.M. (Harvard University) 2002

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Statistics
and the Designated Emphasis
in
Communication, Computation, and Statistics

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Peter L. Bartlett, Chair
Professor Michael I. Jordan
Professor Dan Klein
Professor Bin Yu

Fall 2008

The dissertation of David Stuart Rosenberg is approved:

Chair Date

Date

Date

Date

University of California, Berkeley

Fall 2008

Semi-Supervised Learning with Multiple Views

Abstract

Semi-Supervised Learning with Multiple Views

by

David Stuart Rosenberg
Doctor of Philosophy in Statistics
and the Designated Emphasis
in
Communication, Computation, and Statistics

University of California, Berkeley

Professor Peter L. Bartlett, Chair

In semi-supervised learning, we receive training data comprising labeled points and unlabeled points, and we search for a *target function* that maps new unlabeled points to their appropriate labels. In the co-regularized least squares (CoRLS) algorithm for semi-supervised learning, we assume that the target function is well-approximated by some function in each of two function classes. We search for a function in each class that performs well on the labeled training data, and we restrict our search space to those pairs of functions, one from each class, whose predictions agree on the unlabeled data. By restricting the search space, we potentially get improved generalization performance for the chosen functions. Our first main contribution is a precise characterization of the size, in terms of Rademacher complexity, of the agreement-constrained search space. We find that the agreement constraint reduces the Rademacher complexity by an amount that depends on the "distance" between the function classes, as measured by a data-dependent metric. Experimentally, we find that the amount of reduction in complexity introduced by the agreement constraint correlates with the amount of improvement that the constraint gives in the CoRLS algorithm.

We next present a new framework for multi-view learning called multi-view point cloud regularization (MVPCR), which has both CoRLS and the popular “manifold regularization” algorithms as special cases. MVPCR is initially formulated as an optimization problem over multiple reproducing kernel Hilbert spaces (RKHSs), where the objective function involves both the labeled and unlabeled data. Our second main result is the construction of a single RKHS, with a data-dependent norm, that reduces the original optimization problem to a supervised learning problem in a single RKHS. Using the *multi-view kernel* corresponding to this new RKHS, we can easily convert any standard supervised kernel method into a semi-supervised, multi-view method. The multi-view kernel also allows us to refine our CoRLS generalization bound using localized Rademacher complexity theory. As a practical application of our new framework, we present the manifold co-regularization algorithm, which leads to empirical improvements over manifold regularization on several semi-supervised tasks.

Professor Peter L. Bartlett
Dissertation Committee Chair

To my parents, Michael and Karen,
and my brother, Eric,

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank many people for supporting me during my time as a Ph.D. student. I am most grateful to Peter Bartlett for the guidance and encouragement he provided me in the course of my work. His high standards of research and level of integrity have impressed and inspired me.

I have also learned much from the members of my dissertation committee. From Bin Yu, I have acquired the habit of asking how theoretical results might inform practical applications. Through my collaboration with Dan Klein, I learned a great deal about the practical side of building models for data. From Mike Jordan, I have learned the benefits of studying the deeper mathematical structures related to statistical and machine learning algorithms.

One of the greatest benefits of studying at Berkeley has been the opportunity to form relationships with my fellow graduate students. It was never difficult to find another graduate student with whom I could discuss a research idea or a mathematical challenge. I would especially like to acknowledge Guillaume Obozinski, Simon Lacoste-Julien, Romain Thibaux, Ilya Sutskever, Vince Vu, and Ron Peled, peers I have come to respect as consistent sources of engaging thought and interesting academic discussion during my time as a graduate student. I look forward to continuing these discussions as well as to future collaborations.

I would also like to thank the collaborators who I have not yet had the opportunity to mention: Ben Taskar and Vikas Sindhwani. I learned much from working with Ben and am very grateful for his help and guidance throughout our work. I am thankful to Vikas not only for recognizing an excellent collaboration opportunity, but also for providing sound advice even beyond the scope of our research endeavors.

I gratefully acknowledge the financial support of the National Science Foundation through a VIGRE Fellowship (Grant No. DMS-0130526).

Finally, I thank my parents, Michael and Karen, and my brother, Eric. Their love and support during this long period have helped sustain me on the arduous journey of my doctoral studies.

Chapter 1

Introduction

1.1 Supervised and Semi-Supervised Learning

The goal in many machine learning problems is to learn a mapping from an input space $\mathcal{X}$ to a "label" or "target" space $\mathcal{Y}$. In *supervised learning* problems, this mapping is learned from a training set $(x_1, y_1), \ldots, (x_\ell, y_\ell) \in \mathcal{X} \times \mathcal{Y}$, which we assume is an independent and identically distributed (i.i.d.) sample from some data-generating distribution $P_{\mathcal{X}\times\mathcal{Y}}$. In practice, attaining this *labeled data* from $P_{\mathcal{X}\times\mathcal{Y}}$ is often much more difficult or costly than attaining *unlabeled data* $x_{\ell+1}, \ldots, x_{\ell+u} \in \mathcal{X}$ from the input distribution $P_{\mathcal{X}}$. This motivates the *semi-supervised learning* framework, in which we learn the map from $\mathcal{X}$ to $\mathcal{Y}$ using both labeled data and a typically much larger collection of unlabeled data.

In the prediction setting, where $x \in \mathcal{X}$ is observed and $y \in \mathcal{Y}$ is predicted, it is sufficient to learn the family of predictive distributions[1] $\{P(Y \mid X = x) : x \in \mathcal{X}\}$, which we will denote by $P_{\mathcal{Y}|\mathcal{X}}$. Consider the typical scenario in which we have a relatively small labeled data set and a relatively large unlabeled data set. From the small labeled data set alone, we can only construct a rather poor empirical estimate of $P_{\mathcal{X}\times\mathcal{Y}}$, while we can potentially get a good estimate of $P_{\mathcal{X}}$ from the large unlabeled

[1] To avoid technical complications in this introduction, we assume here and for the rest of this section that the input space is finite, and thus we can talk about the family of conditional distributions $\{P(Y \mid X = x) : x \in \mathcal{X}\}$ without ambiguity or measure-theoretic complications.

data set. This leads us to the basic question of semi-supervised learning: how can we combine a poor estimate of $P_{\mathcal{X}\times\mathcal{Y}}$ with a good estimate of $P_{\mathcal{X}}$ to produce a good estimate of $P_{\mathcal{Y}|\mathcal{X}}$, or some function thereof? More concisely, how and when can *knowledge of $P_{\mathcal{X}}$ help us estimate $P_{\mathcal{Y}|\mathcal{X}}$?*

In Sections 1.2 and 1.3, we introduce a formal framework for describing and analyzing semi-supervised learning (SSL) algorithms. In Section 1.4 we present the multi-view approach to SSL, and in Section 1.5, we present the manifold smoothness approach. In Section 1.6, we present some additional details that bridge the gap between our general discussion and the particular problems we consider in this thesis. In Section 1.7, we review some related work on multi-view learning, with an emphasis on the original co-training model of Blum and Mitchell [10]. Finally, in Section 1.8 we give a brief overview of the contributions we present in subsequent chapters.

1.2 Empirical Risk Minimization

Let $V : \mathcal{Y} \times \mathcal{Y} \to \mathbf{R}$ be a nonnegative loss function, and define the *true risk* of a prediction function f as

$$R(f) = \mathbb{E}V\left(f(X), Y\right),$$

where $(X, Y) \sim P_{\mathcal{X}\times\mathcal{Y}}$. Our goal is to find a function f that has small true risk. However, in practice we cannot typically evaluate $R(f)$, since the joint distribution $P_{\mathcal{X}\times\mathcal{Y}}$ is unknown. Thus we must use a less direct method of finding a function f with small true risk. One standard approach is to minimize the *empirical risk*, defined by

$$\hat{R}_\ell(f) = \frac{1}{\ell}\sum_{i=1}^{\ell} V\left(f(x_i), y_i\right),$$

where $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ is the labeled training set. When a function f performs well with respect to empirical risk, but poorly with respect to true risk, we say that f has *overfit* the data. For example, for the squared loss $V(\hat{y}, y) = (\hat{y} - y)^2$, any function f for which $f(x_i) = y_i$ for all $i = 1, \ldots, \ell$ has zero empirical risk. However, there is no reason, in general, that such a function would have small true risk. Below, we discuss a method to control overfitting by restricting the class of functions over which we minimize the empirical risk.

Fix some class of prediction functions $\mathcal{F}$. An *empirical risk minimizer* (ERM) over $\mathcal{F}$ is any function $\hat{f}_\ell \in \mathcal{F}$ for which

$$\hat{R}_\ell(\hat{f}_\ell) \quad = \quad \min_{f \in \mathcal{F}} \hat{R}_\ell(f).$$

Since we are restricting our search space to $\mathcal{F}$, we will never find a function with true risk better than $\inf_{f\in\mathcal{F}} R(f)$. Without restricting the search space to $\mathcal{F}$, the smallest possible true risk, called the *Bayes risk*, is $\inf_f R(f)$, where the minimum is taken over all measurable functions. For ease of discussion, we assume that there exists a true risk minimizer $f_* \in \mathcal{F}$ for which

$$R(f_*) = \inf_{f \in \mathcal{F}} R(f),$$

as well as a *Bayes prediction function* y_*, for which

$$R(y_*) = \inf_f R(f).$$

The gap between the risk of f_* and the risk of y_* is called the *approximation error* of the class $\mathcal{F}$ – it is our penalty for restricting to the class $\mathcal{F}$ rather than considering all measurable functions. The gap between the risk of $\hat{f}_\ell$, the empirical risk minimizer in $\mathcal{F}$, and f_*, the true risk minimizer in $\mathcal{F}$, is called the *estimation error*, since we are using data to "estimate" the unknown function f_*. We can decompose the "excess risk" that $\hat{f}_\ell$ has over the Bayes risk $R(y_*)$ using these two types of error:

$$R(\hat{f}_\ell) - R(y_*) = \underbrace{R(\hat{f}_\ell) - R(f_*)}_{\text{estimation error}} + \underbrace{R(f_*) - R(y_*)}_{\text{approximation error}}.$$

If we take the loss function V as a given, the only choice that a practitioner needs to make when using empirical risk minimization is what function class to use.

Choosing the Function Class A big challenge in machine learning is choosing a function class $\mathcal{F}$ that gives a good tradeoff between estimation error and approximation error. In general, we expect larger function classes to have larger estimation error and more overfitting than smaller function classes. It is also obvious that making a function class larger can only decrease approximation error. In the next section, we present the *compatibility framework* for semi-supervised learning, in which we use unlabeled data to help make a good choice for $\mathcal{F}$.

1.3 Compatibility framework for Semi-Supervised Learning

In the compatibility framework, as described by Balcan and Blum in [2], we start with a preliminary choice of function class $\mathcal{F}$. With the function class fixed, the goal is to estimate the true risk minimizer $f_* \in \mathcal{F}$. We assume that the target function f_* has certain properties with respect to the unlabeled data. For example, we may assume that f_* is "smooth" with respect to the unlabeled data, in the sense that f_* gives similar predictions for any two unlabeled points that are close together in $\mathcal{X}$. Once we have set forth the properties of f_*, we observe the unlabeled data and form the "reduced" version of $\mathcal{F}$ by eliminating from $\mathcal{F}$ all functions that do not have the desired properties with respect to the unlabeled data. We then do empirical risk minimization over the reduced version of $\mathcal{F}$. Since the reduced version is smaller than the original $\mathcal{F}$, we expect to have smaller estimation error. We now introduce some notation to formalize this approach.

Let $\hat{\chi} : \mathcal{F} \to \mathbf{R}$ be a nonnegative *incompatibility functional.* The hat indicates that the $\hat{\chi}$ functional depends on the unlabeled data. We also fix some level of incompatibility $\kappa \geq 0$, and we assume that our target function f_* has an upper bound on incompatibility given by $\hat{\chi}(f_*) \leq \kappa$. Under this assumption, we can remove all functions $f \in \mathcal{F}$ for which $\hat{\chi}(f) > \kappa$ without affecting the approximation error of the resulting class. On the other hand, since we end up with a smaller function class, we can potentially reduce the estimation error.

The practical problem with the compatibility approach is that it is rarely possible to know how stringent one can make the compatibility requirement without eliminating f_* from the class $\mathcal{F}$. If we prune functions from $\mathcal{F}$ inappropriately or too aggressively, then our approximation error will grow too large and dominate the potential improvements in estimation error. In practice, the acceptable level of incompatibility κ is usually treated as an additional parameter of the algorithm, which could be tuned using cross-validation or another model selection technique.

Many standard semi-supervised learning algorithms can be expressed in this compatibility framework by an appropriate choice of incompatibility function. We refer to [2] for several examples. Below, we discuss two particular approaches that fit into this framework, namely multi-view learning and manifold regularization.

1.4 Multi-View Learning

1.4.1 Informal Introduction

In the multi-view approach to semi-supervised learning, we have several classes of prediction functions, which in this context are often called *views*. This terminology arises in contexts where an input $x \in \mathcal{X}$ can be decomposed naturally as $x = (x^1, \ldots, x^m)$, where each x^i represents a different "view" of the input x. With this decomposition of the input vector, we can define n views, where the ith view is a class of functions that depends only on x^i, while ignoring the other components of x. For example, suppose an input x is a clip from a video of a conference room. We divide x into an audio stream and a video stream, which we write as $x = (x^{\text{audio}}, x^{\text{video}})$. Define our first view $\mathcal{F}^{\text{audio}}$ to consist of prediction functions of the form $x \mapsto f(x^{\text{audio}})$, and our second view, $\mathcal{F}^{\text{video}}$, to consist of functions of the form $x \mapsto f(x^{\text{video}})$. Suppose the goal is to identify who is speaking in each video clip. Although it is certainly easier to identify who is speaking by using the x^{audio} and x^{video} signals together, a person who is familiar with the voices and appearances of the individuals in the conference room could do quite well with just one of these signals. Thus it is reasonable to assume that each of our views, both $\mathcal{F}^{\text{audio}}$ and $\mathcal{F}^{\text{video}}$, contains a function that makes the correct predictions.

We now discuss how multiple views could help us in the conference room problem. Suppose we have a very limited amount of training data, and the only time that Bob spoke, there was an accompanying sound of a truck passing outside in the audio stream x^{audio}, but no corresponding signal in the video track x^{video}. Without additional information, it would be difficult to rule out a prediction function $f^{\text{audio}}_{\text{bad}} \in \mathcal{F}^{\text{audio}}$ that identifies Bob as the speaker whenever a truck passes. However, let us now leverage the assumption that each view has a satisfactory solution to the learning problem. Since there is no evidence for a truck passing in the video signal, there is no function in $\mathcal{F}^{\text{video}}$ that can consistently make the same predictions as $f^{\text{audio}}_{\text{bad}}$. Since we assumed that each view contains a function that makes the correct predictions, and $\mathcal{F}^{\text{video}}$ does not contain any function that matches $f^{\text{audio}}_{\text{bad}}$, we can conclude that $f^{\text{audio}}_{\text{bad}}$ is not the best prediction function in $\mathcal{F}^{\text{audio}}$. That is, we know that $f^{\text{audio}}_{\text{bad}} \neq f^{\text{audio}}_{*}$, and thus we can eliminate $f^{\text{audio}}_{\text{bad}}$ from $\mathcal{F}^{\text{audio}}$. By using the assumption that each view has a good solution to the problem, we can prune out functions, such as $f^{\text{audio}}_{\text{bad}}$, that fit the training data but will not perform well in general.

In practice, to determine whether a function in $\mathcal{F}^{\text{audio}}$ and a function in $\mathcal{F}^{\text{video}}$ make the same predictions, we can compare their predictions on the unlabeled input points. To put things in the compatibility framework of Section 1.3, we can define the incompatibility of a function $f^{\text{audio}} \in \mathcal{F}^{\text{audio}}$ by the total number of disagreements it has with the function $f^{\text{video}} \in \mathcal{F}^{\text{video}}$ that it most closely matches. We can then eliminate all functions in $\mathcal{F}^{\text{audio}}$ that have a large “incompatibility.” We can perform a similar pruning for functions in $\mathcal{F}^{\text{video}}$.

Here we have seen how the assumption that each view has a good prediction function can help rule out the functions in each view that do not have a matching function in the other view. Below, we give a formal derivation of the multi-view approach to semi-supervised learning under the idealized assumption that *each view has a satisfactory solution to the learning problem.*

1.4.2 Formal Derivation of Multi-View Learning

Let $\mathcal{F}^1$ and $\mathcal{F}^2$ be our two views, that is, spaces of prediction functions mapping from $\mathcal{X}$ to $\mathcal{Y}$. For convenience, let $\mathcal{F}$ denote the space of all possible prediction functions from $\mathcal{X}$ to $\mathcal{Y}$. We consider the empirical risk minimization framework, and we denote the risk minimizers in $\mathcal{F}^1$ and $\mathcal{F}^2$ by f^1_* and f^2_* respectively. They satisfy the following equations:

$$R(f^1_*) = \min_{f^1 \in \mathcal{F}^1} R(f^1) \qquad \text{and} \qquad R(f^2_*) = \min_{f^2 \in \mathcal{F}^2} R(f^2)$$

As before, we denote the Bayes prediction function by y_*, and it satisfies

$$R(y_*) = \min_f R(f).$$

We now formalize the notion, discussed above, of two functions being "close." Suppose we have a distance metric $d(\cdot, \cdot)$ on $\mathcal{F}$ and some $\varepsilon \geq 0$ for which

$$d(f^1_*, y_*) \leq \varepsilon \qquad \text{and} \qquad d(f^2_*, y_*) \leq \varepsilon. \tag{1.1}$$

Then by the triangle inequality,

$$d(f^1_*, f^2_*) \leq 2\varepsilon. \tag{1.2}$$

This implies that f^1_* is contained in a ball of radius 2ε around f^2_*. Since we do not know f^2_*, we cannot simply limit our search to this ball. However, we can limit the search to the elements of $\mathcal{F}^1$ that are at most of distance 2ε from *some* element of $\mathcal{F}^2$, since f^1_* is one of these functions. We can make an analogous argument for f^2_*. To formalize this idea, we introduce some new notation. For any subset of functions $\mathcal{G} \subset \mathcal{F}$, let us define the *$\varepsilon$-enlargement of $\mathcal{G}$* by

$$B_\varepsilon(\mathcal{G}) = \{f \in \mathcal{F} : \inf_{g \in \mathcal{G}} d(f, g) \leq \varepsilon\}.$$

By Equation (1.2) and the discussion above, we know that

$$f^1_* \in \mathcal{F}^1 \cap B_{2\varepsilon}(\mathcal{F}^2) \qquad \text{and} \qquad f^2_* \in \mathcal{F}^2 \cap B_{2\varepsilon}(\mathcal{F}^1). \tag{1.3}$$

Since the risk minimizers in each view, f^1_* and f^2_*, are contained in these "reduced" subsets of the original views, we can also find the empirical risk minimizers over these reduced views. By Equation (1.3), the approximation error for the reduced classes is the same as for the original classes, and the estimation error for the empirical risk minimizer over these smaller spaces can only be smaller than for the original views. If our original assumptions in Equation (1.1) are correct, then restricting our function classes as described above can only help us.

We note that we can significantly relax the requirement that $d(\cdot,\cdot)$ be a metric. An approximate triangle inequality would suffice to make a similar argument. For example, suppose $d(\cdot,\cdot)$ is symmetric, and for some $\kappa > 0$ obeys

$$d(a,c) \leq \kappa \left[d(a,b) + d(b,c) \right], \quad \forall a,b,c \in \mathcal{F}. \tag{1.4}$$

Then by an argument analogous to the one above, we will have $f_*^1 \in \mathcal{F}^1 \cap B_{2\kappa\varepsilon}(\mathcal{F}^2)$ and $f_*^2 \in \mathcal{F}^2 \cap B_{2\kappa\varepsilon}(\mathcal{F}^1)$.

In Chapter 2, we consider the L^2 *disagreement functional*, which is defined for any pair of functions $(f^1, f^2) \in \mathcal{F}^1 \times \mathcal{F}^2$ as

$$d(f^1, f^2) = \sum_{i=\ell+1}^{\ell+u} \left[f^1(x_i) - f^2(x_i) \right]^2, \tag{1.5}$$

where the summation runs over the unlabeled data points. Since $\sqrt{d(\cdot,\cdot)}$ is the Euclidean distance metric on $\mathbf{R}^u$, it obeys the triangle inequality, and we can show[2] that $d(\cdot,\cdot)$ satisfies the inequality in Equation (1.4) with $\kappa = 2$.

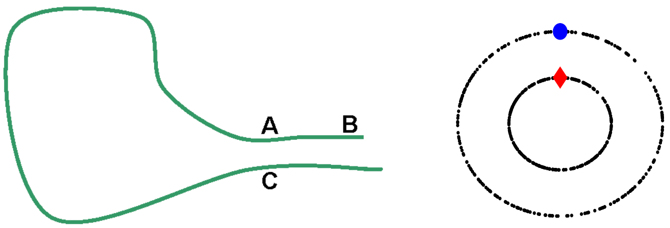
1.4.3 Connection to the Compatibility Framework

To put the multi-view learning approach described above into the compatibility framework, it is easiest to focus our attention on a single class of prediction functions. If we consider $\mathcal{F}^1$, then the measure of incompatibility that naturally corresponds to the derivation above is $\hat{\chi}(f^1) = \inf_{f^2 \in \mathcal{F}^2} d(f^1, f^2)$, where the distance chosen is based on the unlabeled data, such as in Equation (1.5). If our goal is to choose a pair of prediction functions, one from each view, then we could use the incompatibility function $\hat{\chi}\big((f^1, f^2)\big) = d(f^1, f^2)$.

[2] Squaring the triangle inequality for $\sqrt{d(\cdot,\cdot)}$, we get $\forall a,b,c \in \mathcal{F}$ that $d(a,c) \leq d(a,b) + d(b,c) + 2\sqrt{d(a,b)d(b,c)}$. Then by the arithmetic/geometric mean inequality, $\sqrt{d(a,b)d(b,c)} \leq (d(a,b) + d(b,c))/2$. Collecting terms, we get $d(a,c) \leq 2[d(a,b) + d(b,c)]$

1.5 Manifold Smoothness Approaches

In many machine learning applications, the input space $\mathcal{X}$ comes with a natural distance metric, such as the Euclidean distance when $\mathcal{X} = \mathbf{R}^d$. However, the input points themselves often suggest a different measure of distance. For instance, suppose we have a large collection of input points from $\mathcal{X}$, and they all lie on a one-dimensional manifold, as shown in Figure 1.1(a). While the points A and C in the figure are close to each other in Euclidean distance, they are far apart along the manifold. In Figure 1.1(b), the input points come from a manifold with two disjoint connected components. While two points on different components may be close in Euclidean space, we consider them to be infinitely far apart in the manifold distance.



(a) Manifold with one connected component.

(b) Manifold with two connected components.

Figure 1.1: In (a), we have a one-dimensional manifold in the plane. Note that while A and C are close in Euclidean distance, they are far apart "along the manifold." In (b), we have a one-dimensional manifold with two connected components. While the circle and the diamond points are close in Euclidean space, they is no path between them along the manifold, and thus we say they are infinitely far apart in manifold distance.

Many semi-supervised learning algorithms are built on the heuristic that prediction functions should be smooth with respect to the distance on the manifold. Stated another way, points that are close on the manifold should have the same or similar labels. To fit this idea into the compatibility framework discussed in Section 1.3, we need a compatibility functional that measures how smooth a function is, with respect to the manifold distance. One practical issue is that we do not know the data manifold in advance, and fitting a manifold to data is itself a subtle problem[3]. Below, we present compatibility functions that capture the notion of "smoothness with respect to the manifold," while avoiding the need for an explicit estimate of the manifold.

1.5.1 A Measure of Manifold Smoothness

Let $x_1, \ldots, x_n \in \mathcal{X}$ denote a "point cloud," which we typically assume to be an i.i.d. draw from the input distribution $P_{\mathcal{X}}$. In semi-supervised learning contexts, the point cloud usually consists of the unlabeled data points, as well as the input components of the labeled data points. For every pair of points $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$, let $W_{ij} \geq 0$ be a measure of how "similar" or "close" these two points are, with a larger number meaning "more similar." In typical usage, $W_{ij} = 0$ for most pairs of points, and only pairs that are "nearest neighbors" have nonzero values. Let $W = (W_{ij})_{i,j=1}^{n}$ denote the $n \times n$ matrix of these values. For a Euclidean input space $\mathcal{X} = \mathbf{R}^d$, a typical W is given by

$$W_{ij} = \begin{cases} \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma^2\right) & \text{when } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are nearest neighbors, and} \\ 0 & \text{otherwise,} \end{cases}$$

for some bandwidth parameter $\sigma^2 > 0$.

We now define a smoothness functional S that acts on any function $f : \mathcal{X} \to \mathbf{R}$ as follows:

$$S(f) = \sum_{i,j=1}^{n} W_{ij} (f(x_i) - f(x_j))^2, \tag{1.6}$$

where the sum is over all pairs of points in the point cloud. A smaller $S(f)$ value corresponds to a "smoother" f. Note that $S(f)$ does not change if we replace the matrix W by $\left(W^T + W\right)/2$. Thus we can assume, without loss of generality, that W is symmetric. We can write $S(f)$ as a quadratic form in the vector $\boldsymbol{f} = (f(x_1), \ldots, f(x_n))^T$

[3] For example, while it is easy to construct a one-dimensional manifold that contains all our observed points, this is unlikely to capture the structure of the points that we are interested in. Intuitively, we would like a manifold that has low dimensionality, is not too "curvy," and fits the data fairly closely.

as follows:

$$
\begin{aligned}
S(f) &= \sum_{i,j=1}^{n} W_{ij}\left[f(x_i)^2 + f(x_j)^2 - 2f(x_i)f(x_j)\right] \\
&= 2\sum_{i=1}^{n}\left(\sum_{j=1}^{n} W_{ij}\right) f(x_i)^2 - 2\sum_{i,j=1}^{n} W_{ij} f(x_i) f(x_j) \\
&= 2\boldsymbol{f}^T (D - W)\,\boldsymbol{f},
\end{aligned}
$$

where D is the diagonal matrix whose ith entry is $D_{ii} = \sum_{j=1}^{n} W_{ij}$. The matrix $M := D - W$ is known as the (unnormalized) graph Laplacian of W. It is clear from Equation (1.6) that $S(f) \geq 0$ for any f, and thus $\boldsymbol{f}^T M \boldsymbol{f} \geq 0$ for all $\boldsymbol{f} \in \mathbf{R}^n$. This shows that for symmetric W, M is positive semidefinite.

We can take $S(f) = \boldsymbol{f}^T M \boldsymbol{f}$ as a compatibility function that captures the notion of manifold smoothness. We can also capture other measures of smoothness by replacing M with another matrix, such as the normalized Laplacian $I - D^{-1/2} W D^{-1/2}$ or the iterated normalized Laplacian $(I - D^{-1/2} W D^{-1/2})^p$, for some $p \in \{1, 2, \ldots\}$. Some discussion of the relative merits of these smoothness measures for semi-supervised learning may be found in [17] and the references therein.

1.6 RKHS Norm Balls and Norm Penalties

As discussed in Section 1.2, a practitioner should choose a function class $\mathcal{F}$ that makes a good tradeoff between estimation error and approximation error. However, a more practical concern is whether one can carry out the empirical risk minimization over the chosen function space. For a particular class of function spaces, known as reproducing kernel Hilbert spaces (RKHSs), solving for the empirical risk minimizer can be reduced from an optimization problem over a function space to an optimization problem over a finite dimensional space. We refer to Appendix A for a review of RKHS theory, and in particular to Appendix A.4 for a demonstration of the reduction from a function space to a finite dimensional space. Restricting to an RKHS need not hurt us in approximation error, since many commonly used RKHSs have been shown to be *universal.* We say that an RKHS is universal if any function y_* can be uniformly approximated on a compact set, arbitrarily closely, by a function in the RKHS [31, 22].

To control for overfitting, we typically take $\mathcal{F}$ to be a "norm ball" of some radius $r > 0$ in an RKHS. That is, we take $\mathcal{F}$ to be the subset of an RKHS comprising all functions with Hilbert space norm at most r. The radius r of the norm ball can be adjusted to trade off between estimation error and approximation error. For a universal function class, we can get arbitrarily small approximation error by taking r large enough.

When we require that our prediction function lie in a certain subset of a function space, such as in a norm ball, we are making a *hard constraint* on our function space. Up to this point, all our discussions have dealt with hard constraints, and corresponding optimization problems of the form

$$\min\{L(f) : f \in \mathcal{F}, \Omega(f) \leq \kappa\}, \tag{1.7}$$

where $\mathcal{F}$ is the function space over which we are optimizing, $L : \mathcal{F} \to \mathbf{R}$ is a loss functional, and $\Omega(f) \leq \kappa$ represents the hard constraint on our space. In practice, it is often more appealing to think in terms of trading off between minimizing $L(f)$ and $\Omega(f)$, than to decide, *a priori*, on a hard upper limit on $\Omega(f)$. An alternative to hard constraints is *soft constraints*, in which we add a penalty in proportion to $\Omega(f)$, rather than constraining the value of $\Omega(f)$ directly:

$$\min\{L(f) + \lambda\Omega(f) : f \in \mathcal{F}\}, \tag{1.8}$$

for some *regularization parameter* λ. The choice of λ in the soft-constraint form is analogous to the choice of κ in the hard-constraint form. As seems to be the common practice in machine learning literature, we present our algorithms in the soft-constraint formulation.

Although in practice the differences between the two formulations are largely cosmetic, for theoretical discussions we find it easier to work with the hard-constraint version. For example, if we change a hard constraint from $\Omega(f) \leq \kappa$ to $\Omega(f) \leq \kappa/2$, it is clear that the function space decreases in size, and it is often straightforward to make a precise statement about the corresponding decrease in estimation error. However, in the penalization framework, it is less clear what happens when we change the penalty term from $\lambda\Omega(f)$ to $2\lambda\Omega(f)$ – the function class does not change, although it does seem more likely that we will choose an f with smaller $\Omega(f)$. In Chapter 2, where we make a precise statement about just such a change, we must first make a particular boundedness assumption on our loss functional in order to convert to a hard-constraint formulation. Then we can use standard techniques to bound the estimation error.

1.7 Review of Related Work

The notion of capturing the same information in multiple views appeared early in the statistical literature, most notably in Hotelling's work on canonical correlation analysis (CCA) [16]. In the machine learning community, the work of Becker and Hinton [4] and de Sa [13] both considered optimizing the "agreement" between multiple views. However, it was Blum and Mitchell's paper on co-training [10], from several years later, that seems to have inspired the most research in multi-view learning over the past ten years.

1.7.1 The Co-Training Model

Blum and Mitchell's original co-training paper has three basic parts: a presentation of a formal framework for the "co-training model," a short theoretical analysis of their model, and a presentation of their "co-training algorithm," which is motivated by their formal framework. Their co-training model introduced the two principle ideas of multi-view learning: the sufficiency of each view for the prediction task and the compatibility of prediction functions with the unlabeled data. In their conclusions, they state that their framework "provides a way of looking at ... how unlabeled examples can potentially be used to prune away 'incompatible' target concepts to reduce the number of labeled examples needed to learn," which is essentially the perspective we take here. These notions, which we have discussed in Section 1.3, were developed more formally in the later work of Balcan and Blum [2]. The other two components of their paper, namely the theoretical analysis and their co-training algorithm, have not stood the test of time so well as their basic co-training model.

In Blum and Mitchell's co-training model, they suppose that their input space $\mathcal{X}$ decomposes naturally as $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, and functions on the component spaces define the two views. The principle premise of their main theorem is that, for a labeled example $(X_1, X_2, Y) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$ sampled from the true distribution, X_1 and X_2 are conditionally independent given the label Y. The theorem based on this assumption roughly states that, in the classification case, one can learn the right prediction function, up to a fixed permutation of the output labels, using the unlabeled data alone. One can then use a fixed number of labeled training examples to find the right label permutation. Blum and Mitchell themselves acknowledge in a footnote that the strength of the implications of the theorem suggests the implausibility of their conditional independence assumption.

The co-training algorithm is most closely related to the self-training method introduced by Yarowsky [32]. In self-training, a learning algorithm is trained on a labeled training set, and predictions are made on the unlabeled training set (or a random subset of the unlabeled training set). A certain number of the most confidently labeled points are added to the labeled training set with the predicted labels. The learning algorithm is then trained on the augmented labeled training set, and the process repeats.

The co-training algorithm is a multi-view version of self-training: we train one classifier for each view, and we may add an unlabeled point to the labeled training set if the prediction function from *either* view labels the point confidently. The issues of how to choose which points to add to the labeled training set at each iteration, and how many iterations to run, are discussed to some extent in [10].

Blum and Mitchell conclude their paper with the following open question: "To what extent [can] the consistency constraints in the model and the mutual independence assumption ... be relaxed and still allow provable results on the utility of co-training from unlabeled data." The work in this thesis, especially Chapter 2, addresses the latter concern, of proving the utility of the co-training model without an appeal to probabilistic independence assumptions. The former concern, of whether one needs each view to contain a "good" prediction function, is addressed to some extent in the recent work of Sridharan and Kakade [30].

In later work, Dasgupta et al. presented in [12] some additional results based on the conditional independence assumptions used by Blum and Mitchell. In [1], Balcan et al. replace the conditional independence assumption with an assumption that the distribution has a particular "expanding" property on a graph. They show that this assumption gives a more direct motivation for Blum and Mitchell's iterative co-training algorithm. However, their results also require the strong assumption that neither view produces a prediction function that is ever both confident and wrong.

1.7.2 Co-Regularization

Another family of algorithms inspired by the co-training model uses an approach commonly called *co-regularization*. In co-regularization, which we discuss in detail in Chapter 2, we directly penalize disagreement among the prediction functions selected from each view. The idea of using this approach in a semi-supervised framework seems to have been developed independently by several research groups. The earlier references include [23, 18, 28, 14, 11]. The approach we consider most closely follows that of Sindhwani et al. [28] and Brefeld et al. [11]. We also independently developed the co-regularization approach, under the name *kernel co-training*, and presented it along with a theoretical analysis in [24].

1.8 Overview of Contributions

In Chapter 2, we present a two-view learning algorithm called co-regularized least squares (CoRLS), in which the views are reproducing kernel Hilbert spaces (RKHS's), and the disagreement penalty is the average squared difference in predictions on the unlabeled data. The final predictor is the pointwise average of the predictors from each view. We call the set of predictors that can result from this procedure the co-regularized hypothesis class. Our main result is a tight bound on the Rademacher complexity of the co-regularized hypothesis class in terms of the kernel matrices of each RKHS. We find that the co-regularization reduces the Rademacher complexity by an amount that depends on the distance between the two views, as measured by a data-dependent metric. We then use standard techniques to bound the gap between training error and test error for the CoRLS algorithm. Experimentally, we find that the amount of reduction in complexity introduced by co-regularization correlates with the amount of improvement that co-regularization gives in the CoRLS algorithm.

In Chapter 3, we present a significant generalization of the CoRLS algorithm. This generalization allows more than two views as well as a much broader range of compatibility constraints. One of the main results of this section is a reformulation of this multi-view algorithm as a simple optimization problem in a new data-dependent RKHS. With this reformulation, we can apply standard results of RKHS theory to generalize our Rademacher complexity bounds of Chapter 2 to the generalized algorithms of Chapter 3. Furthermore, we can apply the localized Rademacher complexity theory to get estimation error bounds for the empirical risk minimizer over a ball in this new RKHS.

In Chapter 4, we present the *manifold co-regularization* algorithm, which is a two-view algorithm designed to incorporate the manifold smoothness ideas introduced in Section 1.5. In manifold co-regularization, the first view is an RKHS of functions defined on the input space $\mathcal{X}$, just as in CoRLS. The second view, however, comprises functions whose domain is restricted to the set of training points, which we will denote by

$$\mathcal{M} = \{x_1, \ldots, x_\ell, x_{\ell+1}, x_{\ell+u}\}.$$

The norm of a function f in this space is taken to be $\sqrt{S(f)}$, where $S(f) = \boldsymbol{f}^T M \boldsymbol{f}$ is an incompatibility function measuring manifold smoothness, such as one of those presented in Section 1.5 above. If necessary, we add a small multiple of the identity matrix to M to ensure that $\sqrt{S(f)}$ defines a norm. With these two views, we then apply the CoRLS algorithm to find a pair of functions, one from each view, that "agree" and have small empirical risk. For prediction on the unlabeled training points, we can take the average of the predictions of the functions from each view. For prediction on an arbitrary point in $\mathcal{X}$, we can use the function from the first view, which is defined on all of $\mathcal{X}$. The manifold co-regularization algorithm has a parameter λ that controls the magnitude of the disagreement penalty. In Section 4.3, we show that as a limiting case of $\lambda \to \infty$, we get the manifold regularization algorithm presented in [6]. We also present some experiments in which manifold co-regularization algorithm showed significant improvement over the manifold regularization algorithm for most data sets.

Chapter 2

Least-Squares Co-Regularization

2.1 Overview

In this Chapter, we first consider the co-regularized least squares (CoRLS) algorithm, a two-view, semi-supervised version of regularized least squares (RLS). The algorithm was first discussed in [28], and similar approaches were given earlier in [23] and [18]. We also developed this algorithm independently under the name *kernel co-training* [24]. The CoRLS algorithm was extended to more than two views in [11].

Although CoRLS has been shown to work well in practice for both classification [28] and regression [11] problems, many would-be users of the algorithm are deterred by the requirement of choosing two views, as this often seems an arbitrary process. We attempt to improve this situation by showing how the choice of views affects generalization performance, even in settings where we cannot make any probabilistic assumptions about the views.

In [2], Balcan and Blum present a theoretical framework for semi-supervised learning that nicely contains multi-view learning as a special case. Although their results do not assume independent views, their sample complexity results are in terms of the complexity of the space of "compatible predictors," which in the case of multi-view learning corresponds to those predictors that have matching predictors in the other views. To apply these results to a particular multi-view learning algorithm, one must compute the complexity of the class of compatible predictors. This problem is addressed to some extent in [14], in which they compute an upper bound on the Rademacher complexity of the space of compatible predictors. However, their bound is given as the solution to an optimization problem and is difficult to reason about directly.

In CoRLS, the two views are reproducing kernel Hilbert spaces (RKHS's), call them $\mathcal{H}^1$ and $\mathcal{H}^2$. We find predictors $f_*^1 \in \mathcal{H}^1$ and $f_*^2 \in \mathcal{H}^2$ that minimize an objective functional of the form

$$\begin{gathered}\text{Labeled Loss}(f^1, f^2) + \text{Complexity}(f^1, f^2) \\ +\lambda \sum_{x \in \{\text{unlabeled points}\}} [f^1(x) - f^2(x)]^2.\end{gathered}$$

The last term is the "co-regularization" term, which encourages the selection of a pair of prediction functions (f_*^1, f_*^2) that agree on the unlabeled data. Although we consider a more general setting in Chapter 3, in this chapter we follow [11, 14] and consider the average prediction function $\varphi^* := (f_*^1 + f_*^2)/2$, which comes from the class

$$\mathcal{J}^{\text{ave}} := \left\{x \mapsto \left[f^1(x) + f^2(x)\right]/2 \,:\, (f^1, f^2) \in \mathcal{H}^1 \times \mathcal{H}^2\right\}.$$

For typical choices of $\mathcal{H}^1$ and $\mathcal{H}^2$, this class is too large to admit uniform generalization bounds. In Section 2.4.1, however, we see that under a boundedness condition on the labeled loss, the complexity and co-regularization terms force φ^* to come from a much smaller class $\mathcal{J}_\lambda$, where $\lambda \geq 0$ is the coefficient of the co-regularization term in the objective function. As λ increases, we are increasing the amount of co-regularization, and it is clear that the size of $\mathcal{J}_\lambda$ decreases, and we'd expect this to improve generalization. We make this precise in Theorem 2.4.2, where we use standard arguments to bound the gap between training error and test error in terms of the Rademacher complexity of $\mathcal{J}_\lambda$.

The main contribution of this chapter is Theorem 2.4.3, which gives an explicit expression for the Rademacher complexity of $\mathcal{J}_\lambda$, up to a small constant factor. For ordinary kernel RLS (i.e. single view and fully supervised), it is known that the squared Rademacher complexity is proportional to the trace of the kernel matrix (see e.g. [26, Thm 7.39, p. 231]). We find that for the two-view case without co-regularization (i.e. $\lambda = 0$), $\mathcal{J}_\lambda$ has squared Rademacher complexity equal to the average of the traces of the two labeled-data kernel matrices. When $\lambda > 0$, the co-regularization term reduces this quantity by an amount that depends on how different the two views are, and in particular on the average distance between the two views' representations of the labeled data, where the distance metric is determined by the unlabeled data.

2.2 Formal Setup and Notation

We consider the case of two views, though both the algorithm and much of the analysis are extended to multiple views in Chapter 3. Our views are RKHS's $\mathcal{H}^1$ and $\mathcal{H}^2$ of functions mapping from an arbitrary space $\mathcal{X}$ to $\mathcal{Y}$, which we take to be an arbitrary subset of the real line. The CoRLS algorithm takes labeled points $(x_1, y_1), \ldots, (x_\ell, y_\ell) \in \mathcal{X} \times \mathcal{Y}$, and unlabeled points $x_{\ell+1}, \ldots, x_{\ell+u} \in \mathcal{X}$, and solves the following minimization problem:

$$\begin{aligned}(f_*^1, f_*^2) = \operatorname*{arg\,min}_{f^1 \in \mathcal{H}^1, f^2 \in \mathcal{H}^2} \; & \hat{L}(f^1, f^2) + \gamma_1 ||f^1||^2_{\mathcal{H}^1} + \gamma_2 ||f^2||^2_{\mathcal{H}^2} \\ & + \lambda \sum_{i=\ell+1}^{\ell+u} [f^1(x_i) - f^2(x_i)]^2 \end{aligned} \tag{2.1}$$

for some and regularization parameters γ_1, γ_2, and λ, as well as some loss functional $\hat{L}$ that depends on f^1 and f^2 only through their evaluations on the labeled input points $x_1, \ldots, x_\ell$. The final output is $\varphi_* := (f_*^1 + f_*^2)/2$.

In [28, 11], the loss functional considered was

$$\hat{L}(f^1, f^2) = \frac{1}{2\ell} \sum_{i=1}^{\ell} \left(\left[f^1(x_i) - y_i\right]^2 + \left[f^2(x_i) - y_i\right]^2 \right) \tag{2.2}$$

If we use this loss and set $\lambda = 0$ in Equation (2.1), the objective function decouples into two single-view, fully-supervised kernel regularized least squares (RLS) regressions. We also propose the loss functional

$$\hat{L}(f^1, f^2) = \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\frac{f^1(x_i) + f^2(x_i)}{2} - y_i \right)^2 \tag{2.3}$$

as one that seems natural when the final prediction function is $\frac{1}{2}(f^1 + f^2)$, as in [11, 14]. Our analysis applies to both of these loss functionals, as well as many more that depend on the labeled data only and that satisfy a boundedness condition specified in Section 2.4.1. Thus we take the "S" in CoRLS to refer to the squares in the complexity and co-regularization terms, which our analysis requires, rather than to the squares in the loss functional, which we do not require.

2.3 Notation and Preliminaries

We will denote the reproducing kernels corresponding to $\mathcal{H}^1$ and $\mathcal{H}^2$ by $k^1 : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ and $k^2 : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$, respectively. It is convenient to introduce notation for the "span of the data" in each space:

$$\mathcal{L}^1 := \operatorname{span}\{k^1(x_i, \cdot)\}_{i=1}^{\ell+u} \subset \mathcal{H}^1 \qquad \mathcal{L}^2 := \operatorname{span}\{k^2(x_i, \cdot)\}_{i=1}^{\ell+u} \subset \mathcal{H}^2.$$

Proposition 2.3.1. *If the CoRLS optimization problem of Equation (2.1) has a solution, then it has a solution (f_*^1, f_*^2) contained in $\mathcal{L}^1 \times \mathcal{L}^2$. That is,*

$$f_*^1(\cdot) = \sum_{i=1}^{u+\ell} \alpha_i k^1(x_i, \cdot) \in \mathcal{L}^1 \qquad f_*^2(\cdot) = \sum_{i=1}^{u+\ell} \beta_i k^2(x_i, \cdot) \in \mathcal{L}^2$$

for some $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{u+\ell}) \in \mathbf{R}^{u+\ell}$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{u+\ell}) \in \mathbf{R}^{u+\ell}$.

The proof of this proposition is analogous to the proof of the Representer Theorem (c.f. Theorem A.3.1).

Proof. Consider any $f^1 \in \mathcal{H}^1$ and $f^2 \in \mathcal{H}^2$, and let $f_\|^1 \in \mathcal{L}^1$ and $f_\|^2 \in \mathcal{L}^2$ denote their projections into $\mathcal{L}^1$ and $\mathcal{L}^2$, respectively. By Lemma A.2.1, $\|f_\|^1\|_{\mathcal{H}^1}^2 \leq \|f^1\|_{\mathcal{H}^1}^2$ and $\|f_\|^2\|_{\mathcal{H}^2}^2 \leq \|f^2\|_{\mathcal{H}^2}^2$. By Lemma A.2.2, $f^1(x_i) = f_\|^1(x_i)$ and $f^2(x_i) = f_\|^2(x_i)$, for $i = 1, \ldots, \ell+u$. Since $\hat{L}$ only depends on f^1 and f^2 via their evaluations on $x_1, \ldots, x_\ell$, it is clear that the objective function in Equation (2.1) will never increase if we replace (f^1, f^2) by $(f_\|^1, f_\|^2)$. Thus the projection into $\mathcal{L}^1 \times \mathcal{L}^2$ of any solution to Equation (2.1) is itself a solution. □

Define the *kernel matrices* $K^1, K^2 \in \mathbf{R}^{(u+\ell)\times(u+\ell)}$ to be the matrices whose (i,j)'th elements are given by $(K^1)_{ij} = k^1(x_i, x_j)$ and $(K^2)_{ij} = k^2(x_i, x_j)$. It will be convenient to partition these matrices into blocks corresponding to labeled and unlabeled points, as follows:

$$K^1 = \begin{pmatrix} K^1_{UU} & K^1_{UL} \\ K^1_{LU} & K^1_{LL} \end{pmatrix} \qquad K^2 = \begin{pmatrix} K^2_{UU} & K^2_{UL} \\ K^2_{LU} & K^2_{LL} \end{pmatrix}.$$

We will denote an arbitrary element of $\mathcal{L}^1$ by $f^1_{\boldsymbol{\alpha}} = \sum_{i=1}^{u+\ell} \alpha_i k^1(x_i, \cdot)$, and an arbitrary element of $\mathcal{L}^2$ by $f^2_{\boldsymbol{\beta}} = \sum_{i=1}^{u+\ell} \beta_i k^2(x_i, \cdot)$. We can now rewrite the co-regularization term for any pair of functions $(f^1, f^2) = (f^1_{\boldsymbol{\alpha}}, f^2_{\boldsymbol{\beta}}) \in \mathcal{L}^1 \times \mathcal{L}^2$ as

$$\begin{aligned} \sum_{i=\ell+1}^{\ell+u} [f^1(x_i) - f^2(x_i)]^2 &= \sum_{i=\ell+1}^{\ell+u} [f^1_{\boldsymbol{\alpha}}(x_i) - f^2_{\boldsymbol{\beta}}(x_i)]^2 \\ &= \left\| \begin{pmatrix} K^1_{UU} \; K^1_{UL} \end{pmatrix} \boldsymbol{\alpha} - \begin{pmatrix} K^2_{UU} \; K^2_{UL} \end{pmatrix} \boldsymbol{\beta} \right\|^2 \\ &= (\boldsymbol{\alpha}' \; \boldsymbol{\beta}') \begin{pmatrix} K^1_{UU} \\ K^1_{LU} \\ \text{-}K^2_{UU} \\ \text{-}K^2_{LU} \end{pmatrix} \begin{pmatrix} K^1_{UU} \; K^1_{UL} \; \text{-}K^2_{UU} \; \text{-}K^2_{UL} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix}. \end{aligned}$$

By Lemma A.3.2, we have $\|f^1_{\boldsymbol{\alpha}}\|^2_{\mathcal{H}^1} = \boldsymbol{\alpha}' K^1 \boldsymbol{\alpha}$ and $\|f^2_{\boldsymbol{\beta}}\|^2_{\mathcal{H}^2} = \boldsymbol{\beta}' K^2 \boldsymbol{\beta}$. Thus the optimization problem in Equation (2.1) is equivalent to

$$\begin{aligned} (\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*) = \operatorname*{arg\,min}_{(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbf{R}^{u+\ell} \times \mathbf{R}^{u+\ell}} & \hat{L}(f^1_{\boldsymbol{\alpha}}, f^2_{\boldsymbol{\beta}}) + \gamma_1 \boldsymbol{\alpha}' K^1 \boldsymbol{\alpha} + \gamma_2 \boldsymbol{\beta}' K^2 \boldsymbol{\beta} \\ & + \lambda \left\| \begin{pmatrix} K^1_{UU} \; K^1_{UL} \end{pmatrix} \boldsymbol{\alpha} - \begin{pmatrix} K^2_{UU} \; K^2_{UL} \end{pmatrix} \boldsymbol{\beta} \right\|^2, \end{aligned} \tag{2.4}$$

where

$$f^1_*(\cdot) = \sum_{i=1}^{n} (\boldsymbol{\alpha}_*)_i \, k(x_i, \cdot) \qquad \text{and} \qquad f^2_*(\cdot) = \sum_{i=1}^{n} (\boldsymbol{\beta}_*)_i \, k(x_i, \cdot)$$

For each of the loss functionals presented in the beginning of this section, the whole objective function is quadratic in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and thus a solution (f^1_*, f^2_*) can be found by differentiating and solving a system of linear equations. For example, See [28, 11] for more details.

2.4 Results

2.4.1 Bounding the CoRLS Function Class

In this section, we assume the loss functional $\hat{L} : \mathcal{H}^1 \times \mathcal{H}^2 \to [0, \infty)$ satisfies

$$\hat{L}(0, 0) \leq 1.$$

That is, $\hat{L}(f^1, f^2) \leq 1$ for $f^1 \equiv 0$ and $f^2 \equiv 0$. This is true, for example, for the loss functionals given in Equation (2.2) and Equation (2.3) of Section 2.2, provided that $\mathcal{Y} \subset [-1, 1]$. Assuming $\hat{L}(0, 0) \leq 1$, we now derive the "co-regularized" function class $\mathcal{J}_\lambda \subset \mathcal{J}_{\text{ave}}$, from which the CoRLS predictors are drawn. Since λ is fixed in the derivation and discussion that follows, we will drop the subscript and write $\mathcal{J}$ for the co-regularized function class.

Recall that our original problem was to minimize

$$Q(f^1, f^2) := \hat{L}(f^1, f^2) + \gamma_1 \|f^1\|^2_{\mathcal{H}^1} + \gamma_2 \|f^2\|^2_{\mathcal{H}^2} + \lambda \sum_{i=\ell+1}^{\ell+u} [f^1(x_i) - f^2(x_i)]^2$$

over $\mathcal{H}^1 \times \mathcal{H}^2$. Plugging in the trivial predictors $f^1 \equiv 0$ and $f^2 \equiv 0$ gives the following upper bound:

$$\min_{f^1, f^2 \in \mathcal{H}^1 \times \mathcal{H}^2} Q(f^1, f^2) \leq Q(0, 0) = \hat{L}(0, 0) \leq 1$$

Since all terms of $Q(f^1, f^2)$ are nonnegative, we conclude that any (f^1_*, f^2_*) minimizing $Q(f^1, f^2)$ is contained in

$$\mathcal{H} := \left\{ (f^1, f^2) \,:\, \gamma_1 \|f^1\|^2_{\mathcal{H}^1} + \gamma_2 \|f^2\|^2_{\mathcal{H}^2} + \lambda \sum_{i=\ell+1}^{\ell+u} (f^1(x_i) - f^2(x_i))^2 \leq 1 \right\},$$

and the final predictor for the CoRLS algorithm is chosen from the class

$$\mathcal{J} := \{ x \mapsto [f(x) + g(x)]/2 \,:\, (f, g) \in \mathcal{H} \}.$$

Note that the function classes $\mathcal{H}$ and $\mathcal{J}$ do not depend on the labeled data, and thus are deterministic after conditioning on the unlabeled data.

2.4.2 Setup for the Theorems

We will use empirical Rademacher complexity as our measure of the size of a function class. We refer to Appendix B for a review of Rademacher complexity. For convenience, we restate the definition here: The *empirical Rademacher complexity* of a function class $\mathcal{F} = \{\varphi : \mathcal{X} \to \mathcal{Y}\}$ for a sample $x_1, \ldots, x_\ell \in \mathcal{X}$ is defined as

$$\hat{R}_\ell(\mathcal{F}) = \mathbb{E}^\sigma \left[\sup_{\varphi \in \mathcal{F}} \left| \frac{2}{\ell} \sum_{i=1}^{\ell} \sigma_i \varphi(x_i) \right| \right],$$

where the expectation is with respect to $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_\ell\}$, and the σ_i are i.i.d. Rademacher random variables[1].

While it is typical to assume that the unlabeled points

$$x_{\ell+1}, \ldots, x_{\ell+u} \in \mathcal{X}.$$

are drawn i.i.d. from the marginal distribution $P_\mathcal{X}$, here we need only assume that the unlabeled points are independent of the labeled points. More concretely, whether or not we condition on the unlabeled points, the labeled points

$$(x_1, y_1), \ldots, (x_\ell, y_\ell) \in \mathcal{X} \times \mathcal{Y}$$

should be an i.i.d. draw from a distribution $P_{\mathcal{X} \times \mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}$. Beyond this independence from the labeled data, we need not assume anything about the unlabeled points – our claims remain true no matter whether the unlabeled points are generated randomly, by design, or adversarially.

For a given loss function $V : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$, and for any choice of $\varphi \in \mathcal{J}$, we are interested in bounds on the expected loss $\mathbb{E}V(\varphi(X), Y)$. Typically, V would be the loss used to define the labeled empirical risk functional $\hat{L}$ in the CoRLS objective function of Equation (2.1). For example,

$$\hat{L}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} V(f(x_i), y_i).$$

However, the theorem below does not require this relationship between V and $\hat{L}$.

[1] We say σ is a Rademacher random variable if $P(\sigma = 1) = P(\sigma = -1) = \frac{1}{2}$.

Our generalization bound in Theorem 2.4.2 is based on the following well-known theorem (see e.g. [26, Thm 4.9, p. 96]):

Theorem 2.4.1. *Fix $\delta \in (0,1)$, and let $\mathcal{Q}$ be a class of functions mapping from $\mathcal{X} \times \mathcal{Y}$ to $[0,1]$. With probability at least $1-\delta$ over the sample $(X_1, Y_1), \dots, (X_\ell, Y_\ell)$ drawn i.i.d. from $P_{\mathcal{X}\times\mathcal{Y}}$, every $q \in \mathcal{Q}$ satisfies*

$$\mathbb{E}q(X,Y) \leq \frac{1}{\ell}\sum_{i=1}^{\ell} q(X_i, Y_i) + \hat{R}_\ell(\mathcal{Q}) + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}.$$

The expression $\mathbb{E}q(X,Y)$ is deterministic, but unknown to us because we do not know the data generating distribution $P_{\mathcal{X}\times\mathcal{Y}}$. The terms $\frac{1}{\ell}\sum_{i=1}^{\ell} q(X_i, Y_i)$ and $\hat{R}_\ell(\mathcal{Q})$ are random, but with probability at least $1-\delta$, the inequality holds for the observed values of these random quantities and for every $q \in \mathcal{Q}$.

2.4.3 Theorems

In Theorem 2.4.2, we give generalization bounds for the class $\mathcal{J}$ in terms of the empirical Rademacher complexity $\hat{R}_\ell(\mathcal{J})$. In Theorem 2.4.3, we give upper and lower bounds on $\hat{R}_\ell(\mathcal{J})$ that can be written explicitly in terms of blocks of the kernel matrices K^1 and K^2.

Theorem 2.4.2. *Suppose that $V : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ satisfies the following uniform Lipschitz condition: for all $y \in \mathcal{Y}$ and all $\hat{y}_1, \hat{y}_2 \in \mathcal{Y}$ with $\hat{y}_1 \neq \hat{y}_2$,*

$$\frac{|V(\hat{y}_1, y) - V(\hat{y}_2, y)|}{|\hat{y}_1 - \hat{y}_2|} \leq B.$$

Then conditioned on the unlabeled data, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sample of labeled points $(X_1, Y_1), \dots, (X_\ell, Y_\ell)$ drawn i.i.d. from $\mathcal{D}$, we have for any predictor $\varphi \in \mathcal{J}$ that

$$\begin{aligned}\mathbb{E}_{\mathcal{D}} V(\varphi(X), Y) \leq{}& \frac{1}{\ell}\sum_{i=1}^{\ell} L(\varphi(X_i), Y_i) + 2B\hat{R}_\ell(\mathcal{J}) \\ &+ \frac{1}{\sqrt{\ell}}\left(2 + 3\sqrt{\ln(2/\delta)/2}\right)\end{aligned}$$

Note that for $\mathcal{Y} = [-1, 1]$, the conditions of this theorem are satisfied by a modified squared loss function

$$V(\hat{y}, y) = (\tau(\hat{y}) - y)^2/4,$$

where $\tau(y) = \min(1, \max(-1, y))$.

We now give upper and lower bounds on $\hat{R}_\ell(\mathcal{J})$:

Theorem 2.4.3. *The empirical Rademacher complexity of the CoRLS function class $\mathcal{J}$ is bounded above and below as follows:*

$$\frac{1}{\sqrt[4]{2}} \frac{U}{\ell} \le \hat{R}_\ell(\mathcal{J}) \le \frac{U}{\ell},$$

where

$$U^2 = \gamma_1^{-1} \operatorname{tr}(K^1_{LL}) + \gamma_2^{-1} \operatorname{tr}(K^2_{LL}) - \lambda \operatorname{tr}\left(J' \left(I + \lambda S\right)^{-1} J\right),$$

with I the identity matrix, and

$$J = \gamma_1^{-1} K^1_{UL} - \gamma_2^{-1} K^2_{UL} \qquad S = \gamma_1^{-1} K^1_{UU} + \gamma_2^{-1} K^2_{UU}.$$

2.5 Discussion

2.5.1 Unlabeled Data Improves the Bound

The regularization parameter λ controls the amount by which the unlabeled data constrain the hypothesis space. It is obvious from the definition of the hypothesis class $\mathcal{J}$ that if $\lambda_1 \ge \lambda_2 \ge 0$, then $\mathcal{J}_{\lambda_1} \subseteq \mathcal{J}_{\lambda_2}$, and thus $\hat{R}_\ell(\mathcal{J}_{\lambda_1}) \le \hat{R}_\ell(\mathcal{J}_{\lambda_2})$. That is, increasing λ reduces the Rademacher complexity $\hat{R}_\ell(\mathcal{J}_\lambda)$. The amount of this reduction is characterized by the last term in the expression for U^2 in Theorem 2.4.3 above. We denote this term, as a function of λ, as follows:

$$\Delta(\lambda) := \lambda \operatorname{tr}\left(J' \left(I + \lambda S\right)^{-1} J\right).$$

When $\lambda = 0$, the algorithm ignores the unlabeled data, and the reduction is indeed $\Delta(0) = 0$. As we would expect, $\Delta(\lambda)$ is nondecreasing in λ and has a finite limit as $\lambda \to \infty$. We collect these properties in a proposition:

Proposition 2.5.1. $\Delta(0) = 0$, *$\Delta(\lambda)$ is nondecreasing on $\lambda \geq 0$, and*

$$\lim_{\lambda \to \infty} \Delta(\lambda) = \operatorname{tr}(J' S^{-1} J),$$

provided that S is invertible.

Proof. The limit claim is clear if we write the reduction as $\Delta(\lambda) = \operatorname{tr}(J'(\lambda^{-1} I + S)^{-1} J)$. Since K_{UU}^1 and K_{UU}^2 are Gram matrices, their positive combination S is positive semidefinite. Thus we can write $S = Q'DQ$, with diagonal $D \geq 0$ and orthogonal matrix Q. Then

$$\begin{aligned} \Delta(\lambda) &= \operatorname{tr}\left(J'Q'\left(\lambda^{-1} I + D\right)^{-1} QJ\right) \\ &= \sum_{i=1}^{\ell} \sum_{j=1}^{u} (QJ)_{ij}^2 \left(\lambda^{-1} + D_{jj}\right)^{-1} \end{aligned}$$

From this expression, it is clear that $\Delta(\lambda)$ is nondecreasing in λ on $(0, \infty)$. Since $\Delta(\lambda)$ is continuous at $\lambda = 0$, it is nondecreasing on $[0, \infty)$. □

2.5.2 Interpretation of Improvement

Here we present an interpretation of the reduction in complexity $\Delta(\lambda)$. It will be convenient to denote the ith columns of K_{UL}^1 and of K_{UL}^2 by $\boldsymbol{k}_{Ux_i}^1$ and $\boldsymbol{k}_{Ux_i}^2$, respectively. Recalling that x_i is the ith labeled point, for $i = 1, \ldots, \ell$, we see that the entries of these columns are given as follows:

$$\boldsymbol{k}_{Ux_i}^1 = \begin{pmatrix} k^1(x_i, x_{\ell+1}) \\ k^1(x_i, x_{\ell+2}) \\ \vdots \\ k^1(x_i, x_{\ell+u}) \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{k}_{Ux_i}^2 = \begin{pmatrix} k^2(x_i, x_{\ell+1}) \\ k^2(x_i, x_{\ell+2}) \\ \vdots \\ k^2(x_i, x_{\ell+u}) \end{pmatrix}$$

We interpret these columns as giving two different representations of the ith labeled data point by its kernel "inner product" with each of the u unlabeled points. We can now write the reduction $\Delta(\lambda)$ as follows:

$$\begin{aligned}
\Delta(\lambda) &= \lambda \operatorname{tr}\left(\left(\gamma_1^{-1}K_{UL}^1 - \gamma_2^{-1}K_{UL}^2\right)' (I+\lambda S)^{-1} \left(\gamma_1^{-1}K_{UL}^1 - \gamma_2^{-1}K_{UL}^2\right)\right) && (2.5)\\
&= \lambda \sum_{i=1}^{\ell} \left(\gamma_1^{-1}\boldsymbol{k}_{Ux_i}^1 - \gamma_2^{-1}\boldsymbol{k}_{Ux_i}^2\right)' (I+\lambda S)^{-1} \left(\gamma_1^{-1}\boldsymbol{k}_{Ux_i}^1 - \gamma_2^{-1}\boldsymbol{k}_{Ux_i}^2\right) && (2.6)\\
&= \sum_{i=1}^{\ell} \rho^2\left(\gamma_1^{-1}\boldsymbol{k}_{Ux_i}^1\ ,\ \gamma_2^{-1}\boldsymbol{k}_{Ux_i}^2\right), && (2.7)
\end{aligned}$$

where, $\rho(\cdot,\cdot)$ is a metric on the space $\mathbf{R}^u$ defined by

$$\begin{aligned}
\rho^2(\boldsymbol{s},\boldsymbol{t}) &= \lambda(\boldsymbol{s}-\boldsymbol{t})' (I+\lambda S)^{-1} (\boldsymbol{s}-\boldsymbol{t})\\
&= (\boldsymbol{s}-\boldsymbol{t})' (I/\lambda + S)^{-1} (\boldsymbol{s}-\boldsymbol{t})
\end{aligned}$$

where I is the $u \times u$ identity matrix, and recall that

$$S = \gamma_1^{-1}K_{UU}^1 + \gamma_2^{-1}K_{UU}^2$$

is a weighted sum of the unlabeled data kernel matrices.

We see that the complexity reduction $\Delta(\lambda)$ grows with the ρ-distance between the two different (scaled) representations of the labeled points. Note that the metric ρ is determined by S, which is the weighted sum of the Gram matrices of the two kernels on unlabeled data. For very small λ, this distance is essentially measured using the Euclidean norm, and the reduction itself is quite small. As λ grows, the distance approaches that determined by S^{-1}, where S is the sum of the two unlabeled data kernel matrices. Loosely summarized, the reduction $\Delta(\lambda)$ is proportional to the difference between the representations of the labeled data in the two different views, where the measure of difference is determined by the unlabeled data.

2.6 Proofs

2.6.1 Proof of Theorem 2.4.2.

Define the loss class

$$\mathcal{Q} = \{(x,y) \mapsto V(\varphi(x),y)\ :\ \varphi \in \mathcal{J}\}.$$

By assumption, any function in $\mathcal{Q}$ maps into $[0,1]$. Applying Theorem 2.4.1, we have for any $\varphi \in \mathcal{J}$, with probability at least $1-\delta$ over the labeled sample $(X_i, Y_i)_{i=1}^{\ell}$, that

$$\mathbb{E}V(\varphi(X), Y) \leq \frac{1}{\ell}\sum_{i=1}^{\ell} V(\varphi(X_i), Y_i) + \hat{R}_\ell(\mathcal{Q}) + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}.$$

The following lemma completes the proof:

Lemma 2.6.1. $\hat{R}_\ell(\mathcal{Q}) \leq 2B\hat{R}_\ell(\mathcal{J}) + \frac{2}{\sqrt{\ell}}$.

Proof. Define the functions $g_y = V(0,y)$ and $h_y(\hat{y}) = V(\hat{y},y) - V(0,y)$. Then $V(\varphi(x),y) = g_y + h_y(\varphi(x))$, and

$$\mathcal{Q} = g_y + h_y \circ \mathcal{J}.$$

Since $|g_y| \leq 1$ for all y, we have

$$\hat{R}_\ell(\mathcal{Q}) \leq \hat{R}_\ell(h_y \circ \mathcal{J}) + \frac{2}{\sqrt{\ell}},$$

by a property of Rademacher complexity (see e.g. [26, Thm 4.15(v), p. 101]). For all y, $h_y(\cdot)$ is Lipschitz with constant B, and $h_y(0) = 0$. Thus by the Ledoux-Talagrand contraction inequality [20, Thm 4.12, p.112] we have

$$\hat{R}_\ell(h_y \circ \mathcal{J}) \leq 2B\hat{R}_\ell(\mathcal{J})$$

□

The bound in Lemma 2.6.1 is sometimes of the right order of magnitude and sometimes quite loose. Consider the space $\mathcal{X} \times \mathcal{Y} = \mathbf{R} \times \{-1,0,1\}$ and the loss $V(\hat{y},y) = (\tau(\hat{y}) - y)^2/4$, where $\tau(y) = \min(1, \max(-1,y))$. Note that V satisfies the uniform Lipschitz condition of Theorem 2.4.2. Assume $P(y=0) = 1$. Then for any class $\mathcal{J}$ of prediction functions that map into $\{-1,1\}$, with probability 1 we have $V(\varphi(x),y) = 1$. Thus with probability 1 over the labeled sample $(x_1,y_1), \ldots, (x_\ell, y_\ell)$, the empirical Rademacher complexity of the loss class $\mathcal{Q} = \{(x,y) \mapsto V(\varphi(x),y) \ :$

$\varphi \in \mathcal{J}\}$ is given by

$$\begin{aligned}
\hat{R}_\ell(\mathcal{Q}) &= \mathbb{E}^\sigma\left[\sup_{f\in\mathcal{Q}}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i f(x_i,y_i)\right|\right] \\
&= \mathbb{E}^\sigma\left[\sup_{\varphi\in\mathcal{J}}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i V(\varphi(x_i),y_i)\right|\right] \\
&= \frac{2}{\ell}\mathbb{E}^\sigma\left[\sup_{\varphi\in\mathcal{J}}\left|\sum_{i=1}^{\ell}\sigma_i\right|\right] = \frac{2}{\ell}\mathbb{E}^\sigma\left|\sum_{i=1}^{\ell}\sigma_i\right|.
\end{aligned}$$

By the Kahane-Khintchine inequality (Lemma B.2.1), we conclude that $\hat{R}_\ell(\mathcal{Q}) = \Theta(\ell^{-1/2})$. Note that this expression holds for the loss class $\mathcal{Q}$ corresponding to *any* class of prediction functions $\mathcal{J} : \mathcal{X} \to \{-1,1\}$. If we choose $\mathcal{J}$ small, say $\mathcal{J} = \{x \mapsto 1\}$, then $\hat{R}_\ell(\mathcal{J}) = \frac{2}{\ell}\mathbb{E}^{\boldsymbol{\sigma}}\left|\sum_{i=1}^{\ell}\sigma_i\right| = \Theta(\ell^{-1/2})$, and the upper bound on $\hat{R}_\ell(\mathcal{Q})$ given by Lemma 2.6.1 is $2B\hat{R}_\ell(\mathcal{J}) + 2/\sqrt{\ell} = O(\ell^{-1/2})$, which is tight in order of magnitude. On the other hand, let us take $\mathcal{J}$ to be the set of all measurable functions mapping $\mathcal{X} \to \{-1,1\}$, and assume that x has a continuous distribution so that with probability 1 the sample points $x_1,\ldots,x_\ell$ are all distinct. Then for any sample of Rademacher variables $\sigma_1,\ldots,\sigma_\ell$, there is a $\varphi \in \mathcal{J}$ such that $\varphi(x_i) = \sigma_i$ for $i = 1,\ldots,\ell$. Thus with probability 1,

$$\hat{R}_\ell(\mathcal{J}) = \mathbb{E}^\sigma\left[\sup_{\varphi\in\mathcal{J}}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i\varphi(x_i)\right|\right] = \frac{2}{\ell}\mathbb{E}^\sigma\left|\sum_{i=1}^{\ell}\sigma_i\sigma_i\right| = \frac{2}{\ell}\mathbb{E}^\sigma\left|\sum_{i=1}^{\ell}1\right| = 2.$$

Thus for this choice of $\mathcal{J}$, the bound given by Lemma 2.6.1 is $O(1)$, which is quite loose, since $\hat{R}_\ell(\mathcal{Q}) = O(\ell^{-1/2})$.

2.6.2 Proof of Theorem 2.4.3

We prove this theorem in several steps, starting from the definition

$$\hat{R}_\ell(\mathcal{J}) = \mathbb{E}^{\boldsymbol{\sigma}}\left[\sup_{(f,g)\in\mathcal{H}}\left|\frac{1}{\ell}\sum_{i=1}^{\ell}\sigma_i(f(x_i)+g(x_i))\right|\right], \tag{2.8}$$

where as usual the expectation is with respect to the Rademacher random variables $\boldsymbol{\sigma} = (\sigma_1,\ldots,\sigma_\ell)$. We first convert from a supremum over the function space $\mathcal{H}$ to a supremum over a finite-dimensional Euclidean space that we can solve directly. Next, we use the Kahane-Khintchine inequality to bound the expectation over $\boldsymbol{\sigma}$ above and below by expectations that we can compute explicitly. Finally, with some matrix algebra we can write our bounds on $\hat{R}_\ell(\mathcal{J})$ in terms of blocks of the original kernel matrices.

Converting to Euclidean Space

Since $(f,g) \in \mathcal{H}$ implies $(-f,-g) \in \mathcal{H}$, we can drop the absolute value in Equation (2.8). Next, note that the expression inside the supremum depends only on the values of f and g at the sample points. Recall our notation for the "span of the data" in each space: $\mathcal{L}^1 = \operatorname{span}\{k^1(x_i,\cdot)\}_{i=1}^{\ell+u} \subset \mathcal{H}^1$ and $\mathcal{L}^2 = \operatorname{span}\{k^2(x_i,\cdot)\}_{i=1}^{\ell+u} \subset \mathcal{H}^2$. By the reproducing kernel property, it is easy to show (c.f. Lemma A.2.2) that $f(x_j) = (\operatorname{Proj}_{\mathcal{L}^1} f)(x_j)$ and $g(x_j) = (\operatorname{Proj}_{\mathcal{L}^2} g)(x_j)$ for any input point in the training set x_j. Thus the supremum in the expression for $\hat{R}_\ell(\mathcal{J})$ is unchanged if we restrict the domain of the supremum to $(f,g) \in (\mathcal{L}^1 \times \mathcal{L}^2) \cap \mathcal{H}$. Applying these observations, we get

$$\hat{R}_\ell(\mathcal{J}) = \frac{1}{\ell}\mathbb{E}^{\boldsymbol{\sigma}} \sup\left\{\sum_{i=1}^{\ell} \sigma_i(f(x_i)+g(x_i)) \ : \ (f,g) \in (\mathcal{L}^1 \times \mathcal{L}^2) \cap \mathcal{H}\right\}.$$

Finally, we can write the set of functions $(\mathcal{L}^1 \times \mathcal{L}^2) \cap \mathcal{H}$ as

$$\begin{aligned}(\mathcal{L}^1 \times \mathcal{L}^2) \cap \mathcal{H} &= \Bigg\{(f_{\boldsymbol{\alpha}}, g_{\boldsymbol{\beta}}) \ : \ \gamma_1 \boldsymbol{\alpha}' K^1 \boldsymbol{\alpha} + \gamma_2 \boldsymbol{\beta}' K^2 \boldsymbol{\beta} \\ &\qquad + \lambda \sum_{i=\ell+1}^{\ell+u} \left(f_{\boldsymbol{\alpha}}(x_i) - g_{\boldsymbol{\beta}}(x_i)\right)^2 \le 1\Bigg\} \\ &= \left\{(f_{\boldsymbol{\alpha}}, g_{\boldsymbol{\beta}}) \ : \ (\boldsymbol{\alpha}' \ \boldsymbol{\beta}') N \begin{pmatrix}\boldsymbol{\alpha} \\ \boldsymbol{\beta}\end{pmatrix} \le 1\right\}\end{aligned}$$

where

$$N := \begin{pmatrix}\gamma_1 K^1 & 0 \\ 0 & \gamma_2 K^2\end{pmatrix} + \lambda \begin{pmatrix} K^1_{UU} \\ K^1_{LU} \\ \text{-}K^2_{UU} \\ \text{-}K^2_{LU}\end{pmatrix} \left(K^1_{UU} \ K^1_{UL} \ \text{-}K^2_{UU} \ \text{-}K^2_{UL}\right),$$

Now we can write

$$\hat{R}_\ell(\mathcal{J}) = \frac{1}{\ell}\mathbb{E}^{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbf{R}^{u+\ell}}\left\{\boldsymbol{\sigma}'\left(K^1_{LU}\ K^1_{LL}\right)\boldsymbol{\alpha} + \boldsymbol{\sigma}'\left(K^2_{LU}\ K^2_{LL}\right)\boldsymbol{\beta}\right.\right.$$

$$\left.\left.\text{s.t. } (\boldsymbol{\alpha}'\ \boldsymbol{\beta}')N\begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{pmatrix} \leq 1\right\}\right]$$

Evaluating the Supremum

For a symmetric positive definite (spd) matrix M, it is easy to show that

$$\sup_{\boldsymbol{\alpha}:\boldsymbol{\alpha}'M\boldsymbol{\alpha}\leq 1} \boldsymbol{v}'\boldsymbol{\alpha} = \left\|M^{-1/2}\boldsymbol{v}\right\|.$$

However, our matrix N may not have full rank. Note that each entry of the column vector $\left(K^1_{LU}\ K^1_{LL}\right)\boldsymbol{\alpha}$ is an inner product between $\boldsymbol{\alpha}$ and a row (or column, by symmetry) of K^1. Similarly, each entry of $\left(K^2_{LU}\ K^2_{LL}\right)\boldsymbol{\beta}$ is an inner product between $\boldsymbol{\beta}$ and a row or column of K^2. Thus if we write the projections of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ onto the column spaces of K^1 and K^2 as

$$\begin{aligned}\boldsymbol{\alpha}_\| &= \mathrm{Proj}_{\mathrm{ColSpace}(K^1)}\boldsymbol{\alpha}\\ \boldsymbol{\beta}_\| &= \mathrm{Proj}_{\mathrm{ColSpace}(K^2)}\boldsymbol{\beta}\end{aligned}$$

then we have

$$\begin{aligned}\left(K^1_{LU}\ K^2_{LL}\right)\boldsymbol{\alpha} &= \left(K^1_{LU}\ K^2_{LL}\right)\boldsymbol{\alpha}_\|\\ \left(K^2_{LU}\ K^2_{LL}\right)\boldsymbol{\beta} &= \left(K^2_{LU}\ K^2_{LL}\right)\boldsymbol{\beta}_\|\end{aligned}$$

and by similar reasoning we can show that

$$(\boldsymbol{\alpha}'\ \boldsymbol{\beta}')N\begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{pmatrix} = (\boldsymbol{\alpha}'_\|\ \boldsymbol{\beta}'_\|)N\begin{pmatrix}\boldsymbol{\alpha}_\|\\ \boldsymbol{\beta}_\|\end{pmatrix}.$$

Thus the supremum can be rewritten as

$$\sup_{\substack{\boldsymbol{\alpha}_\|\in\mathrm{ColSpace}(K^1)\\ \boldsymbol{\beta}_\|\in\mathrm{ColSpace}(K^2)}}\left\{\boldsymbol{\sigma}'\left(K^1_{LU}\ K^2_{LL}\right)\boldsymbol{\alpha}_\| + \boldsymbol{\sigma}'\left(K^2_{LU}\ K^2_{LL}\right)\boldsymbol{\beta}_\|\right.$$

$$\left.\text{s.t. } (\boldsymbol{\alpha}'_\|\ \boldsymbol{\beta}'_\|)N\begin{pmatrix}\boldsymbol{\alpha}_\|\\ \boldsymbol{\beta}_\|\end{pmatrix} \leq 1\right\}$$

Changing to eigenbases cleans up this expression and clears the way for substantial simplifications in later sections. Diagonalize the psd kernel matrices to get orthonormal bases for the column spaces of K^1 and K^2:

$$V'K^1V = \Sigma_{\mathcal{F}} \qquad W'K^2W = \Sigma_{\mathcal{G}}$$

where $\Sigma_{\mathcal{F}}$ and $\Sigma_{\mathcal{G}}$ are diagonal matrices containing the nonzero eigenvalues, and the columns of V and W are bases for the column spaces of K^1 and K^2, respectively. Now introduce column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ such that

$$\boldsymbol{\alpha}_\| = V\boldsymbol{a} \qquad \text{and} \qquad \boldsymbol{\beta}_\| = W\boldsymbol{b}$$

Applying this change of variables to the quadratic form, we get

$$(\boldsymbol{\alpha}'_\| \; \boldsymbol{\beta}'_\|)N \begin{pmatrix} \boldsymbol{\alpha}_\| \\ \boldsymbol{\beta}_\| \end{pmatrix} \;\; = \;\; (\boldsymbol{a}' \; \boldsymbol{b}')T \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix}$$

where

$$T = \Sigma + \lambda RR'$$

with

$$\Sigma := \begin{pmatrix} \gamma_1 \Sigma_{\mathcal{F}} & 0 \\ 0 & \gamma_2 \Sigma_{\mathcal{G}} \end{pmatrix} \qquad R := \begin{pmatrix} V' & 0 \\ 0 & W' \end{pmatrix} \begin{pmatrix} K^1_{UU} \\ K^1_{LU} \\ \text{-}K^2_{UU} \\ \text{-}K^2_{LU} \end{pmatrix}$$

The matrix T is spd, since it is the sum of the spd diagonal matrix Σ and the psd matrix $\lambda RR'$. For compactness, define

$$\mathcal{W} = \left(K^1_{LU} \; K^2_{LL} \; K^2_{LU} \; K^2_{LL} \right) \begin{pmatrix} V & 0 \\ 0 & W \end{pmatrix}.$$

We can now write

$$\hat{R}_\ell(\mathcal{J}) = \frac{1}{\ell}\mathbb{E}^{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{a},\boldsymbol{b}} \left\{ \boldsymbol{\sigma}'\mathcal{W} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} \text{ s.t. } (\boldsymbol{a}' \; \boldsymbol{b}')T \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} \leq 1 \right\} \right]$$

Since T is spd, we can evaluate the supremum as described above to get

$$\hat{R}_\ell(\mathcal{J}) \;\; = \;\; \frac{1}{\ell}\mathbb{E}^{\boldsymbol{\sigma}} \left\| T^{-1/2}\mathcal{W}'\boldsymbol{\sigma} \right\|$$

Bounding $\hat{R}_\ell(\mathcal{J})$ above and below

We make use of the Kahane-Khintchine inequality, which we state here for convenience:

Lemma 2.6.2. *[Kahane-Khintchine inequality] For any vectors $a_1, \ldots, a_n$ in a Hilbert space and independent Rademacher random variables $\sigma_1, \ldots, \sigma_n$, we have*

$$\frac{1}{2}\mathbb{E}\left\|\sum_{i=1}^n \sigma_i a_i\right\|^2 \leq \left(\mathbb{E}\left\|\sum_{i=1}^n \sigma_i a_i\right\|\right)^2 \leq \mathbb{E}\left\|\sum_{i=1}^n \sigma_i a_i\right\|^2$$

Taking the columns of $T^{-1/2}\mathcal{W}'$ to be the a_i's, we can apply this lemma to our expression for $\hat{R}_\ell(\mathcal{J})$ to get

$$\frac{1}{\sqrt[4]{2}}\frac{U}{\ell} \leq \hat{R}_\ell(\mathcal{J}) \leq \frac{U}{\ell},$$

where

$$\begin{aligned} U^2 &:= \mathbb{E}^{\boldsymbol{\sigma}}\left\|T^{-1/2}\mathcal{W}'\boldsymbol{\sigma}\right\|^2 \\ &= \mathbb{E}^{\boldsymbol{\sigma}} \operatorname{tr}\left[\mathcal{W}T^{-1}\mathcal{W}'\boldsymbol{\sigma}\boldsymbol{\sigma}'\right] \\ &= \operatorname{tr}\left[\mathcal{W}T^{-1}\mathcal{W}'\right] \end{aligned}$$

To get the second line we expanded the squared norm, took the trace of the scalar quantity inside the expectation, and rotated the factors inside the trace. To get the last equality we interchanged the trace and the expectation and noted that $\mathbb{E}\boldsymbol{\sigma}\boldsymbol{\sigma}'$ is the identity matrix.

Writing our Expression in terms of the Original Kernel Matrices

It will be helpful to divide V and W into labeled and unlabeled parts. We note the dimensions of V and W are $(\ell+u)\times r_{\mathcal{F}}$ and $(\ell+u)\times r_{\mathcal{G}}$, where $r_{\mathcal{F}}$ and $r_{\mathcal{G}}$ are the ranks of K^1 and K^2, respectively. So we have

$$K^1 = \begin{pmatrix} K^1_{UU} & K^1_{UL} \\ K^1_{LU} & K^1_{LL} \end{pmatrix} = \begin{pmatrix} V_u \\ V_\ell \end{pmatrix} \Sigma_{\mathcal{F}}(V_u' \; V_\ell') \tag{2.9}$$

$$K^2 = \begin{pmatrix} K^2_{UU} & K^2_{UL} \\ K^2_{LU} & K^2_{LL} \end{pmatrix} = \begin{pmatrix} W_u \\ W_\ell \end{pmatrix} \Sigma_{\mathcal{G}}(W_u' \; W_\ell') \tag{2.10}$$

Rearranging the diagonalization, we also have

$$V' \begin{pmatrix} K^1_{UU} & K^1_{UL} \\ K^1_{LU} & K^1_{LL} \end{pmatrix} = \Sigma_{\mathcal{F}}(V'_u \; V'_\ell) \tag{2.11}$$

$$W' \begin{pmatrix} K^2_{UU} & K^2_{UL} \\ K^2_{LU} & K^2_{LL} \end{pmatrix} = \Sigma_{\mathcal{G}}(W'_u \; W'_\ell) \tag{2.12}$$

By equating blocks in these four matrix equations, we attain all the substitutions we need to write U^2 in terms of the original kernel submatrices: $K^1_{UU}, K^1_{UL}, K^1_{LL}, K^2_{UU}, K^2_{UL}, K^2_{LL}$. For example, by equating the top left submatrices in Equation 2.9, we get $K^1_{UU} = V_u \Sigma_{\mathcal{F}} V'_u$. Using these substitutions, we can write:

$$\mathcal{W}' = \begin{pmatrix} V' & 0 \\ 0 & W' \end{pmatrix} \begin{pmatrix} K^1_{UL} \\ K^1_{LL} \\ K^2_{UL} \\ K^2_{LL} \end{pmatrix} = \begin{pmatrix} \Sigma_{\mathcal{F}} & 0 \\ 0 & \Sigma_{\mathcal{G}} \end{pmatrix} \begin{pmatrix} V'_\ell \\ W'_\ell \end{pmatrix}$$

$$R = \begin{pmatrix} V' & 0 \\ 0 & W' \end{pmatrix} \begin{pmatrix} K^1_{UU} \\ K^1_{LU} \\ \text{-}K^2_{UU} \\ \text{-}K^2_{LU} \end{pmatrix} = \begin{pmatrix} \Sigma_{\mathcal{F}} & 0 \\ 0 & \text{-}\Sigma_{\mathcal{G}} \end{pmatrix} \begin{pmatrix} V'_u \\ W'_u \end{pmatrix}$$

We now work on the T^{-1} factor in our expression $U^2 = \mathrm{tr}(\mathcal{W}T^{-1}\mathcal{W})$. Using the Sherman-Morrison-Woodbury formula[2], we expand $T^{-1} = (\Sigma + \lambda RR')^{-1}$ as

$$T^{-1} = \Sigma^{-1} - \lambda\Sigma^{-1}R\left(I + \lambda R'\Sigma^{-1}R\right)^{-1} R'\Sigma^{-1}$$

Since Σ and $I + \lambda R'\Sigma^{-1}R$ are spd, our inverses exist and the expansion is justified. Substituting this expansion into our expression for U^2, we get

$$U^2 = \mathrm{tr}\left(\mathcal{W}\Sigma^{-1}\mathcal{W}'\right) - \lambda\,\mathrm{tr}\left(\mathcal{W}\Sigma^{-1}R\left(I + \lambda R'\Sigma^{-1}R\right)^{-1} R'\Sigma^{-1}\mathcal{W}\right)$$

[2] $(A + UU')^{-1} = A^{-1} - A^{-1}U(I + U^TA^{-1}U)^{-1}U^TA^{-1}$, provided the inverses exist [15, p. 50].

We'll have our final form once we can express $\mathcal{W}\Sigma^{-1}\mathcal{W}'$, $R'\Sigma^{-1}R$, and $R'\Sigma^{-1}\mathcal{W}$ in terms of the original kernel matrix blocks. We have

$$\begin{aligned}
\mathcal{W}\Sigma^{-1}\mathcal{W}' &= (V_\ell \; W_\ell) \begin{pmatrix} \Sigma_{\mathcal{F}} & 0 \\ 0 & \Sigma_{\mathcal{G}} \end{pmatrix} \begin{pmatrix} \gamma_1^{-1}\Sigma_{\mathcal{F}}^{-1} & 0 \\ 0 & \gamma_2^{-1}\Sigma_{\mathcal{G}}^{-1} \end{pmatrix} \begin{pmatrix} \Sigma_{\mathcal{F}} & 0 \\ 0 & \Sigma_{\mathcal{G}} \end{pmatrix} \begin{pmatrix} V_\ell' \\ W_\ell' \end{pmatrix} \\
&= (V_\ell \; W_\ell) \begin{pmatrix} \gamma_1^{-1}\Sigma_{\mathcal{F}} & 0 \\ 0 & \gamma_2^{-1}\Sigma_{\mathcal{G}} \end{pmatrix} \begin{pmatrix} V_\ell' \\ W_\ell' \end{pmatrix} \\
&= \gamma_1^{-1} V_\ell \Sigma_{\mathcal{F}} V_\ell' + \gamma_2^{-1} W_\ell \Sigma_{\mathcal{G}} W_\ell' \\
&= \gamma_1^{-1} K_{LL}^1 + \gamma_2^{-1} K_{LL}^2
\end{aligned}$$

The last equality follows by equating submatrices in Equations 2.9 and 2.10. Using very similar steps, but with different substitutions read from Equations 2.9 and 2.10, we also get

$$\begin{aligned}
R'\Sigma^{-1}R &= \gamma_1^{-1} K_{UU}^1 + \gamma_2^{-1} K_{UU}^2 = S \\
R'\Sigma^{-1}\mathcal{W}' &= \gamma_1^{-1} K_{UL}^1 - \gamma_2^{-1} K_{UL}^2 = J
\end{aligned}$$

Putting things together, we get

$$U^2 = \operatorname{tr}\left(\gamma_1^{-1} K_{LL}^1 + \gamma_2^{-1} K_{LL}^2\right) - \lambda \operatorname{tr}\left(J' \left(I + \lambda S\right)^{-1} J\right)$$

□

2.7 Experiments

The objective of our experiments was to investigate whether the reduction in hypothesis space complexity due to co-regularization correlates with an improvement in test performance. We closely followed the experimental setup used in [11] on the UCI repository data sets [9]. We selected those data sets with continuous target values, between 5 and 500 examples, and at least 5 features. For each of these 29 data sets, we generated two views by randomly splitting the features into two sets of equal size. To get our performance numbers, we averaged over 10 randomly chosen feature splits. To evaluate the performance of each split, we performed 10-fold 'inverse' cross validation, in which one fold is used as labeled data, and the other nine folds are used as unlabeled data.

For each data set, we used the CoRLS algorithm with loss functional

$$\hat{L}(f,g) = \frac{1}{2\ell}\sum_{i=1}^{\ell}\Big(\big[f(x_i) - y_i\big]^2 + \big[g(x_i) - y_i\big]^2\Big),$$

as in [28, 11]. In [11], CoRLS is compared to RLS. Here, we compare CoRLS with co-regularization parameter $\lambda = 1/10$ to the performance with $\lambda = 0$. In Figure 2.1, for each data set we plot the percent improvement in RMS error when going from $\lambda = 0$ to $\lambda = 1/10$ against the size of the decrease in the Rademacher complexity. The correlation between these two quantities is $r = .67$. The error bars extend two standard errors from the mean.

2.8 Conclusions

In this chapter, we have given tight bounds for the Rademacher complexity of the co-regularized hypothesis space arising from two RKHS views, as well as a generalization bound for the CoRLS algorithm. While our theorems bound the gap between training and test performance, it says nothing about the absolute performance: if neither view has good prediction functions, then we will have poor performance, regardless of the generalization bound. Nevertheless, experimentally we found a correlation between improved generalization bounds and improved test performance. This may suggest that for typical parameter settings, or at least for those used in [11], reduction in Rademacher complexity is a good predictor of improved performance.

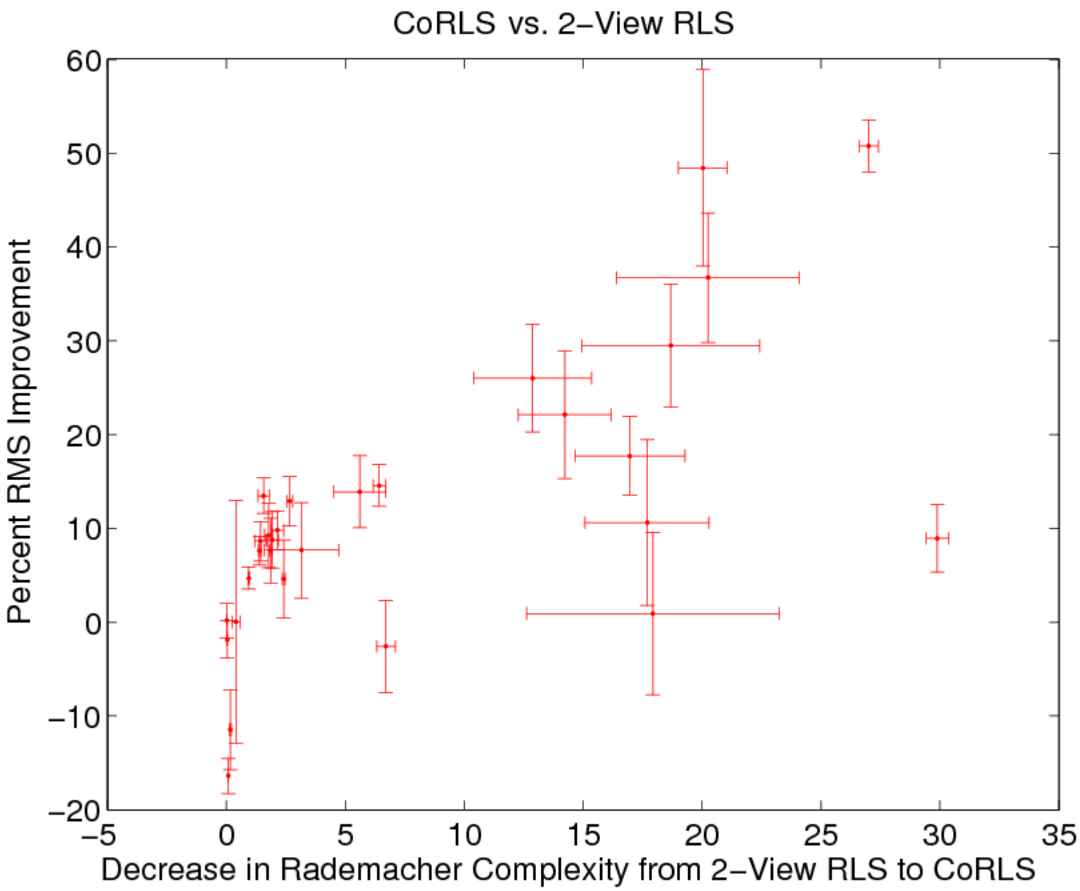


Figure 2.1: The percent improvement in RMS error of the CoRLS algorithm ($\lambda = 1/10$) over the 2-view RLS algorithm ($\lambda = 0$) vs. the decrease in Rademacher complexity.

Chapter 3

Multi-View Reproducing Kernel Hilbert Spaces

3.1 Setup

We now generalize the RKHS co-regularization setting of Chapter 2 in several directions. The new framework is called *multi-view point cloud regularization.* The generalizations are the following:

1. We explicitly allow m views, rather than just 2.
2. We make predictions with an arbitrary, but fixed in advance, linear combination of the prediction functions in each view, rather than with a pointwise average.
3. We replace the co-regularization term with a more general "multi-view point cloud regularization" term, which allows the incorporation of other approaches, such as manifold regularization, into the same framework as co-regularization.

The main result of this chapter, Theorem 3.2.1, does not put any conditions on the labeled and unlabeled data, beyond that they be points in $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{X}$, respectively. To emphasize this point, we now call the points from $\mathcal{X}$ the *point cloud*, and enumerate them as $x_1, \ldots, x_n$. From a practical perspective, in addition to points sampled from

$P_{\mathcal{X}}$, the point cloud may include points from our labeled training set, or even hand-chosen design points. To make any statistical claims, however, we will require that the labeled data be independent of the point cloud.

The Prediction Functions

Instead of two RKHS views, we now consider m views. Let $\mathcal{H}^1, \ldots, \mathcal{H}^m$ be reproducing kernel Hilbert spaces of real-valued functions on $\mathcal{X}$, with kernels $k^1, \ldots, k^m : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$, respectively. For convenience, we introduce the product space

$$\mathcal{F} = \mathcal{H}^1 \times \cdots \times \mathcal{H}^m.$$

Rather than restricting to predictions of the form $(f^1 + f^2)/2$, as in Chapter 2, we now allow predictions with any fixed linear combination of functions in each view: For any $f = (f^1, \ldots, f^m) \in \mathcal{F}$, the final prediction function is given by $u(f)$, where

$$u(f) = u\left((f^1, \ldots, f^m)\right) = a_1 f^1 + \cdots + a_m f^m,$$

for any fixed $a_1, \ldots, a_m \in \mathbf{R}$. In particular, we allow $a_i = 0$, which we make use of in Section 4.3 below. Finally, we define the space of prediction functions $\tilde{\mathcal{H}}$ as the image of $\mathcal{F}$ under u. That is,

$$\tilde{\mathcal{H}} = u(\mathcal{F}) = \{u(f) \mid f \in \mathcal{F}\}.$$

The Loss Functional

Let $L : \tilde{\mathcal{H}} \to \mathbf{R}$ denote an arbitrary loss functional on the space of prediction functions. In regression, for example, a typical loss functional would be $L(h) = \frac{1}{\ell}\sum_{i=1}^{\ell}\left[h(x_i) - y_i\right]^2$, where $(x_1, y_1), \ldots, (x_\ell, y_\ell) \in \mathcal{X} \times \mathcal{Y}$ is the set of labeled training points. We note that formulating the loss functional in this way is slightly less general than the formulation given in Chapter 2. There, we allowed the loss functional to have a separate dependence on the prediction function from each view, as in Equation (2.2). For Theorem 3.2.1 below, we require that the loss functional depend only on the combined prediction functions, as in Equation (2.3). Since we typically make our final predictions with $u\left((f^1,\dots,f^m)\right)$, it seems like a fairly mild restriction to not allow direct dependence on the component prediction functions $f^1,\dots,f^m$.

The Point Cloud Penalty

In this chapter, instead of unlabeled data, we consider a "point cloud," which we denote by $x_1,\dots,x_n \in \mathcal{X}$. This terminology follows that used in [27], and in fact the setting here includes their semi-supervised learning framework as a special case.

While in CoRLS we consider only an L^2-disagreement penalty, here we allow a much broader class of regularization penalties. For any element $f = (f^1,\dots,f^m) \in \mathcal{F}$, we define the following column vector of function evaluations on the point cloud $x_1,\dots,x_n$:

$$\underline{\boldsymbol{f}} = \left(f^1(x_1),\dots,f^1(x_n),\dots,f^m(x_1),\dots,f^m(x_n)\right)^T.$$

Throughout this chapter, we use bold face to indicate a finite dimensional column vector and an underline to indicate that a vector is the concatenation of a column vector associated with each of the m views. In multi-view point cloud regularization, we replace the L^2-disagreement penalty $\sum_i \left(f^1(x_i) - f^2(x_i)\right)^2$ with the more general $\underline{\boldsymbol{f}}^T M \underline{\boldsymbol{f}}$, where $M \in \mathbf{R}^{mn\times mn}$ is a positive semidefinite matrix. This penalty naturally allows for more than two views and allows a much broader range of multi-view penalties.

The Objective Function

The objective function for multi-view point cloud regularization is the following:

$$\operatorname*{arg\,min}_{\varphi\in\tilde{\mathcal{H}}} \min_{(f^1,\dots,f^m)\in u^{-1}(\varphi)} L(\varphi) + \sum_{i=1}^{m} \gamma_i \|f^i\|^2_{\mathcal{H}^i} + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{f}}, \tag{3.1}$$

where $\gamma_1,\dots,\gamma_m > 0$ are RKHS norm regularization parameters, and $\lambda \geq 0$ is the point cloud norm regularization parameter. Just as for CoRLS, we can use arguments similar to the proof of the representer theorem to express this objective as a finite dimensional optimizaton problem. As discussed below, this framework subsumes the

the manifold regularization framework described in [6] as well as the multi-view co-regularization framework described in [11], except for our restriction that the loss functional be of the form $L(a_1 f^1 + \cdots + a_m f^m)$.

In Section 3.2, we express this objective function as a standard Tikhonov regularization problem over a new data-dependent RKHS, which we will call the *multi-view RKHS*. We also find an explicit, closed form expression for the corresponding *multi-view kernel*. This multi-view kernel can be plugged into any standard kernel method, giving convenient and immediate access to multi-view point cloud regularization techniques for a wide variety of learning problems. As a particular application, in Section 3.4 we present the *co-regularization kernel*, first discussed in [29], by which we can convert any supervised learning method into a multi-view semi-supervised approach. In Section 4.3, we propose a co-regularization based alternative to manifold regularization [6, 27] that leads to major empirical improvements on semi-supervised tasks. Finally, in Section 3.3, we use the multi-view kernel to give much simpler proofs of the complexity and generalization results of Chapter 2. We also give more refined generalization bounds using known results about the localized Rademacher complexity of kernel classes.

3.2 The Multi-View RKHS

We would like to put a norm on $\tilde{\mathcal{H}}$ such that the multi-view point cloud optimization problem in Equation (3.1) can be written simply as

$$\arg\min_{\varphi \in \tilde{\mathcal{H}}} L(\varphi) + \|\varphi\|_{\tilde{\mathcal{H}}}^2,$$

which is a standard RKHS regularization problem. By comparison to Equation (3.1), it is clear that when such a norm $\|\cdot\|_{\tilde{\mathcal{H}}}$ exists, it can be written as

$$\|\varphi\|_{\tilde{\mathcal{H}}} = \sqrt{\min_{(f^1,\ldots,f^m) \in u^{-1}(\varphi)} \sum_{i=1}^{m} \gamma_i \|f^i\|_{\mathcal{H}^i}^2 + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{f}}} \tag{3.2}$$

We will show that this norm indeed exists and, furthermore, corresponds to an inner product for which $\tilde{\mathcal{H}}$ is a reproducing kernel Hilbert space. Finally, we give an explicit form for the reproducing kernel of $\tilde{\mathcal{H}}$ in terms of the matrix M, the kernels $k^1, \dots, k^m$, and their evaluations on the fixed set of points $x_1, \dots, x_n$. The closed form for this kernel is quite important for actual implementation, and it also allows us to give explicit expressions for complexity and generalization bounds.

We now introduce some additional notation. Denote the point-cloud kernel matrix for the jth view by $K^j = (k^j(x_i, x_k))_{i,k=1}^n$, and define the block diagonal matrix $\mathcal{K} = \operatorname{diag}(K^1, \dots, K^m) \in \mathbf{R}^{nm \times nm}$. Recall that $a_1, \dots, a_m$ are the coefficients used in the linear combination of views $u(f)$, and $\gamma_1, \dots, \gamma_m > 0$ are scale parameters in front of the norm penalties in the objective function. Define diagonal matrices of these parameters

$$\underline{A} = \operatorname{diag}\left(\underbrace{a_1, \dots, a_1}_{n \text{ times}}, \dots, \underbrace{a_m, \dots, a_m}_{n \text{ times}}\right) \qquad \underline{G} = \operatorname{diag}\left(\underbrace{\gamma_1, \dots, \gamma_1}_{n \text{ times}}, \dots, \underbrace{\gamma_m, \dots, \gamma_m}_{n \text{ times}}\right),$$

and denote the column vector of kernel evaluations between the point cloud and an arbitrary point $x \in \mathcal{X}$, for each kernel, by

$$\underline{\boldsymbol{k}}_x = \left(k^1(x_1, x), \dots, k^1(x_n, x), \dots, k^m(x_1, x), \dots, k^m(x_n, x)\right)^T.$$

Our main theorem is the following:

Theorem 3.2.1. *For any linear mapping $u : (f^1, \dots, f^m) \mapsto a_1 f^1 + \dots + a_m f^m$, for any symmetric positive semidefinite $M \in \mathbf{R}^{nm \times nm}$, for any norm regularization parameters $\gamma_1, \dots, \gamma_m > 0$ and for any point-cloud norm regularization parameter $\lambda \geq 0$, there exists a reproducing kernel Hilbert space $\tilde{\mathcal{H}} = u(\mathcal{F})$ with norm*

$$\|\varphi\|_{\tilde{\mathcal{H}}} = \sqrt{\min_{(f^1, \dots, f^m) \in u^{-1}(\varphi)} \sum_{i=1}^m \gamma_i \|f^i\|_{\mathcal{H}^i}^2 + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{f}}}, \tag{3.3}$$

and kernel function

$$\tilde{k}(z, x) = \sum_{j=1}^m \frac{a_j^2}{\gamma_j} k^j(z, x) - \lambda \underline{\boldsymbol{k}}_x^T \underline{A}\,\underline{G}^{-1} \left(I + \lambda M \underline{G}^{-1} \mathcal{K}\right)^{-1} M \underline{G}^{-1} \underline{A}\,\underline{\boldsymbol{k}}_z$$

Proof

The Product Hilbert Space

Recall that $\mathcal{F} = \mathcal{H}^1 \times \cdots \times \mathcal{H}^m$. For any $f = (f^1, \ldots, f^m) \in \mathcal{F}$ and $g = (g^1, \ldots, g^m) \in \mathcal{F}$, we propose the following as their inner product in $\mathcal{F}$:

$$\langle f, g \rangle_{\mathcal{F}} = \sum_{i=1}^{m} \gamma_i \left\langle f^i, g^i \right\rangle_{\mathcal{H}^i} + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{g}}$$

It is straightforward to check that $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is a valid inner product. Moreover, we have the following:

Lemma 3.2.2. *$\mathcal{F}$ is a Hilbert space.*

Proof. We need to show that $\mathcal{F}$ is complete. We define the Hilbert space norm in the standard way: $\|f\|_{\mathcal{F}}^2 = \langle f, f \rangle_{\mathcal{F}}$, and let $f_1, f_2, \ldots$ be any Cauchy sequence in $\mathcal{F}$. Then for any $a, b \in \{1, 2, 3, \ldots\}$, we have

$$\|f_a - f_b\|_{\mathcal{F}}^2 = \sum_{i=1}^{m} \gamma_i \|f_a^i - f_b^i\|_{\mathcal{H}^i}^2 + \lambda \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}_b\right)^T M \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}_b\right)$$

Note that both the left hand side (LHS) and all terms on the right hand side (RHS) are nonnegative. Thus if the LHS goes to zero, then all terms on the RHS go to 0. Therefore, $f_1^i, f_2^i, \ldots$ is a Cauchy sequence in $\mathcal{H}^i$, for each $i = 1, \ldots, m$. Since the $\mathcal{H}^i$ are RKHS's, they are complete, and thus $\lim_{a \to \infty} f_a^i = f^i$, for some $f^i \in \mathcal{H}^i$, for each $i = 1, \ldots, m$. We now show that $f_a = (f_a^1, \ldots, f_a^m) \to f = (f^1, \ldots, f^m) \in \mathcal{F}$. We have

$$\|f_a - f\|_{\mathcal{F}}^2 = \sum_{i=1}^{m} \gamma_i \|f_a^i - f^i\|_{\mathcal{H}^i}^2 + \lambda \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right)^T M \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right)$$

If we write $\sigma_1(M)$ for the largest eigenvalue of M, then by the variational characterization of eigenvalues we have

$$\begin{aligned} 0 \leq \lambda \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right)^T M \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right) &\leq \lambda \sigma_1(M) \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right)^T \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right) \\ &= \lambda \sigma_1(M) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(f_a^i(x_j) - f^i(x_j)\right)^2 . \end{aligned}$$

Since $\mathcal{H}^i$'s are RKHS's, their evaluation functionals are continuous. That is, for any fixed $x \in \mathcal{X}$, if f_a^i and f^i are close in the $\|\cdot\|_{\mathcal{H}^i}$ norm, then $|f_a^i(x) - f^i(x)|$ is small. Thus we can choose N so large that $a > N$ implies $|f_a^i(x_j) - f^i(x_j)| \leq \varepsilon$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Thus $\lambda \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right)^T M \left(\underline{\boldsymbol{f}}_a - \underline{\boldsymbol{f}}\right) \to 0$ as $a \to \infty$. The summands in $\sum_{i=1}^m \gamma_i \|f_a^i - f^i\|_{\mathcal{H}^i}$ each go to zero by convergence of f_a^i to f^i in $\mathcal{H}^i$. □

$\tilde{\mathcal{H}}$ is a Hilbert Space

In Theorem 5 of [7], they prove a special case of our theorem with $m = 2$ (two views) and $\lambda = 0$ (i.e. without any point-cloud regularization term in the norm). To show that $\tilde{\mathcal{H}}$ is a Hilbert space, we follow the approach of [7] and push the Hilbert space structure from $\mathcal{F}$ onto $\tilde{\mathcal{H}}$: Denote the nullspace of u by $N := u^{-1}(0)$. N is a closed subspace of $\mathcal{F}$, and thus its orthogonal complement $N^\perp$ is also a closed subspace. We can consider $N^\perp$ as a Hilbert space with the inner product that is the natural restriction of $\langle\cdot,\cdot\rangle_{\mathcal{F}}$ to $N^\perp$. Define $v : N^\perp \to \tilde{\mathcal{H}}$ as the restriction of u to $N^\perp$. Then v is a bijection, and we define an inner product on $\tilde{\mathcal{H}}$ by

$$\langle f, g\rangle_{\tilde{\mathcal{H}}} = \langle v^{-1}(f), v^{-1}(g)\rangle_{\mathcal{F}}.$$

We conclude that $\left(\tilde{\mathcal{H}}, \langle\cdot,\cdot\rangle_{\tilde{\mathcal{H}}}\right)$ is a Hilbert space isomorphic to $N^\perp$.

The Norm

The norm of any $h \in \tilde{\mathcal{H}}$ is given by

$$\begin{aligned}
\|h\|^2_{\tilde{\mathcal{H}}} &= \|v^{-1}(h)\|^2_{\mathcal{F}} && \text{by definition of } \tilde{\mathcal{H}} \\
&= \min_{n \in N} \left(\|v^{-1}(h)\|^2_{\mathcal{F}} + \|n\|^2_{\mathcal{F}}\right) && \text{since } 0 \in N \\
&= \min_{n \in N} \left(\|v^{-1}(h) + n\|^2_{\mathcal{F}}\right) && \text{since } v^{-1}(h) \perp N \text{ in } \mathcal{F} \\
&= \min_{f \in u^{-1}(h)} \left(\|f\|^2_{\mathcal{F}}\right) && \text{since } u^{-1}(h) = \left\{v^{-1}(h) + n \mid n \in N\right\} \\
&= \min_{(f^1, \ldots, f^m) \in u^{-1}(h)} \sum_{i=1}^m \gamma_i \|f^i\|_{\mathcal{H}^i} + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{f}}
\end{aligned}$$

We see that the inner product on $\tilde{\mathcal{H}}$ induces the norm claimed in the theorem statement. We note in passing that, by a similar argument, we can show

$$v^{-1}(h) = \operatorname*{arg\,min}_{f \in u^{-1}(h)} \|f\|_{\mathcal{F}}^2$$

Reproducing Property of the Kernel

Overview We now construct a reproducing kernel $\tilde{k}(\cdot,\cdot)$ for the Hilbert space $\tilde{\mathcal{H}}$. Our approach is to find, for each $z \in \mathcal{X}$, a tuple $h_z = (h_z^1, \dots, h_z^m) \in \mathcal{F}$ such that the function $u(h_z) \in \tilde{\mathcal{H}}$ has the reproducing property:

$$\forall \varphi \in \tilde{\mathcal{H}}, \langle \varphi, u(h_z) \rangle_{\tilde{\mathcal{H}}} = \varphi(z)$$

We then define $\tilde{k}(z,\cdot) = u(h_z)$. If we can do this for each $z \in \mathcal{X}$, then by the definition of RKHS (see Section ...), $\tilde{k}(\cdot,\cdot)$ is indeed a reproducing kernel for $\tilde{\mathcal{H}}$.

Construction Fix $z \in \mathcal{X}$, and choose $h_z = (h_z^1, \dots, h_z^m) \in \mathcal{F}$ in a way to be specified later. Define $\tilde{k}(z,\cdot) = u(h_z)$. By definition of the maps u and v, we can write

$$v^{-1}(\tilde{k}(z,\cdot)) = h_z + n_z,$$

for some $n_z \in N$.

Fix any $\varphi \in \tilde{\mathcal{H}}$, and let $f = (f^1, \dots, f^m) = v^{-1}(\varphi)$. Then

$$\begin{aligned}
\left\langle \varphi, \tilde{k}(z,\cdot) \right\rangle_{\tilde{\mathcal{H}}} &= \left\langle v^{-1}(\varphi), v^{-1}\left(\tilde{k}(z,\cdot)\right) \right\rangle_{\mathcal{F}} \\
&= \left\langle v^{-1}(\varphi), n_z + h_z \right\rangle_{\mathcal{F}} \\
&= \left\langle \left(f^1, \dots, f^m\right), h_z \right\rangle_{\mathcal{F}} \text{ since } v^{-1}(\varphi) \perp n_z \\
&= \sum_{j=1}^m \gamma_j \left\langle f^j, h_z^j \right\rangle_{\mathcal{H}^j} + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{h}}_z,
\end{aligned}$$

where

$$\underline{\boldsymbol{h}}_z = \left(h_z^1(x_1), \dots, h_z^1(x_n), \dots, h_z^m(x_1), \dots, h_z^m(x_n)\right)^T$$

We now guess a form for the functions $h_z^1, \dots, h_z^m : \mathcal{X} \to \mathbf{R}$. Suppose that for each $z \in \mathcal{X}$ there are scalars $\beta_i^j(z)$, for $i = 1, \dots, n$ and $j = 1, \dots, m$, for which

$$h_z^j(\cdot) = \frac{1}{\gamma_j}\left[a_j k^j(z,\cdot) + \lambda \sum_{i=1}^n \beta_i^j(z) k^j(x_i,\cdot)\right]$$

Then cleary $h_z^j \in \mathcal{H}^j$, and

$$\begin{aligned}\sum_{j=1}^{m} \gamma_j \left\langle f^j, h_z^j \right\rangle_{\mathcal{H}^j} &= \sum_{j=1}^{m} \left\langle f^j, a_j k^j(z,\cdot) + \lambda \sum_{i=1}^{n} \beta_i^j(z) k^j(x_i,\cdot) \right\rangle \\ &= \sum_{j=1}^{m} a_j \left\langle f^j, k^j(z,\cdot)\right\rangle + \lambda \sum_{j=1}^{m}\sum_{i=1}^{n} \beta_i^j(z) \left\langle f^j, k^j(x_i,\cdot)\right\rangle \\ & \sum_{j=1}^{m} a_j f^j(z) + \lambda \sum_{j=1}^{m}\sum_{i=1}^{n} \beta_i^j(z) f^j(x_i).\end{aligned}$$

For convenience, we define the following vector for each $x \in \mathcal{X}$:

$$\underline{\boldsymbol{\beta}}_x = \left(\beta_1^1(x), \dots, \beta_n^1(x), \dots, \beta_1^m(x), \dots, \beta_n^m(x)\right)^T.$$

With this notation, we have

$$\underline{\boldsymbol{h}}_z = \underline{G}^{-1}\left(\underline{A}\underline{\boldsymbol{k}}_z + \lambda \mathcal{K} \underline{\boldsymbol{\beta}}_z\right),$$

and so

$$\begin{aligned}\left\langle \varphi, \tilde{k}(z,\cdot)\right\rangle_{\tilde{\mathcal{H}}} &= \sum_{j=1}^{m} a_j f^j(z) + \lambda \sum_{j=1}^{m}\sum_{i=1}^{n} \beta_i^j(z) f^j(x_i) + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{h}}_z \\ &= \varphi(z) + \lambda \underline{\boldsymbol{f}}^T \underline{\boldsymbol{\beta}}_z + \lambda \underline{\boldsymbol{f}}^T M \underline{G}^{-1}\left(\underline{A}\underline{\boldsymbol{k}}_z + \lambda \mathcal{K}\underline{\boldsymbol{\beta}}_z\right) \\ &= \varphi(z) + \lambda \underline{\boldsymbol{f}}^T \left[\underline{\boldsymbol{\beta}}_z + M\underline{G}^{-1}\left(\underline{A}\underline{\boldsymbol{k}}_z + \lambda\mathcal{K}\underline{\boldsymbol{\beta}}_z\right)\right] \\ &= \varphi(z) + \lambda \underline{\boldsymbol{f}}^T \left[\left(I + \lambda M \underline{G}^{-1}\mathcal{K}\right)\underline{\boldsymbol{\beta}}_z + M\underline{G}^{-1}\underline{A}\underline{\boldsymbol{k}}_z\right]\end{aligned}$$

Note that if $(I + \lambda M\underline{G}^{-1}\mathcal{K})\,\underline{\boldsymbol{\beta}}_z + M\underline{G}^{-1}\underline{A}\underline{\boldsymbol{k}}_z = 0$, then we will have the reproducing property for z. If $I + \lambda M\underline{G}^{-1}\mathcal{K}$ has full rank, then the solution to this equation is

$$\underline{\boldsymbol{\beta}}_z \;=\; -\left(I + \lambda M\underline{G}^{-1}\mathcal{K}\right)^{-1} M\underline{G}^{-1}\underline{A}\underline{\boldsymbol{k}}_z.$$

Note that $I + \lambda M\underline{G}^{-1}\mathcal{K}$ is not symmetric, in general, even for symmetric M and $\mathcal{K}$. Nevertheless, we can show[1] that $I + \lambda M\underline{G}^{-1}\mathcal{K}$ does indeed have full rank. Thus we

[1] Here we show that $I + \lambda M\mathcal{K}$ has full rank, for any $\lambda \geq 0$ and PSD M and $\mathcal{K}$: Write $M^{1/2}$ for the PSD square root of M. For arbitrary square matrices A and B, it is well known that the eigenvalues of AB and BA are the same (see e.g. [8]). Thus $M\mathcal{K} = M^{1/2}(M^{1/2}\mathcal{K})$ and $(M^{1/2}\mathcal{K})M^{1/2}$ have the same eigenvalues, as do the matrices $I + \lambda M\mathcal{K}$ and $I + \lambda M^{1/2}\mathcal{K}M^{1/2}$. In the latter matrix, the term $\lambda M^{1/2}\mathcal{K}M^{1/2}$ is clearly PSD since $\mathcal{K}$ is PSD and $\lambda \geq 0$. Thus all eigenvalues of $I + \lambda M\mathcal{K}$ are positive, and so has full rank.

can put things together as follows:

$$\begin{aligned}
\tilde{k}(z,x) &= u\left((h_z^1,\ldots,h_z^m)\right) \\
&= \sum_{j=1}^{m} \frac{a_j}{\gamma_j}\left[a_j k^j(z,x) + \lambda \sum_{i=1}^{n} \beta_i^j(z) k^j(x_i,x)\right] \\
&= \sum_{j=1}^{m} \frac{a_j^2}{\gamma_j} k^j(z,x) + \lambda \sum_{j=1}^{m} \frac{a_j}{\gamma_j} \sum_{i=1}^{n} \beta_i^j(z) k^j(x_i,x) \\
&= \sum_{j=1}^{m} \frac{a_j^2}{\gamma_j} k^j(z,x) + \lambda \underline{\boldsymbol{k}}_x^T \underline{A}\,\underline{G}^{-1} \underline{\boldsymbol{\beta}}_z \\
&= \sum_{j=1}^{m} \frac{a_j^2}{\gamma_j} k^j(z,x) - \lambda \underline{\boldsymbol{k}}_x^T \underline{A}\,\underline{G}^{-1} \left(I + \lambda M \underline{G}^{-1} \mathcal{K}\right)^{-1} M \underline{G}^{-1} \underline{A}\,\underline{\boldsymbol{k}}_z
\end{aligned}$$

Thus we have found a kernel function $\tilde{k} \in \tilde{\mathcal{H}}$ that has the reproducing property. Thus $\tilde{\mathcal{H}}$ is an RKHS with kernel $\tilde{k}$. □

3.3 Rademacher Complexity and Generalization Bounds

We now use Theorem 3.2.1 together with standard techniques and results from RKHS theory to derive Rademacher complexity and generalization bounds for the multi-view algorithm presented above.

3.3.1 Rademacher Complexity Bound

To get the complexity bound, we first use Theorem 3.2.1 to show that the final prediction function for multi-view point cloud regularization comes from a norm ball of a particular radius in the RKHS $\tilde{\mathcal{H}}$. We denote the norm ball of radius r by

$$\tilde{\mathcal{H}}_r = \left\{\varphi \in \tilde{\mathcal{H}} \mid \|\varphi\|_{\tilde{\mathcal{H}}} \le r\right\}.$$

We then apply a standard result to get tight upper and lower bounds on the empirical Rademacher complexity of $\tilde{\mathcal{H}}_r$ in terms of the empirical kernel matrix. Using Theorem 3.2.1 again, we can write down an explicit form for this kernel matrix in terms of the original view kernel matrices $K^1,\ldots,K^m$. These results are collected in the following theorem:

Theorem 3.3.1. *The prediction function for multi-view point cloud regularization, defined as*

$$\varphi^* = \arg\min_{\varphi \in \tilde{\mathcal{H}}} \min_{(f^1,\ldots,f^m) \in u^{-1}(\varphi)} L(\varphi) + \sum_{i=1}^{m} \gamma_i \|f^i\|^2_{\mathcal{H}^i} + \lambda \underline{\boldsymbol{f}}^T M \underline{\boldsymbol{f}}$$

always has norm bounded as

$$\|\varphi^*\|^2_{\tilde{\mathcal{H}}} \leq r^2 := L(0), \tag{3.4}$$

where the 0 *denotes the prediction function that always predicts* 0*. That is,* $\varphi^* \in \tilde{\mathcal{H}}_r$*. The empirical Rademacher complexity of* $\tilde{\mathcal{H}}_r$ *for a sample* $x_1^\ell, \ldots, x_\ell^\ell \in \mathcal{X}$ *is bounded as*

$$\frac{1}{\sqrt[4]{2}} \frac{2r}{\ell} \sqrt{\operatorname{tr} \tilde{K}} \leq \hat{R}_\ell(\tilde{\mathcal{H}}_r) \leq \frac{2r}{\ell} \sqrt{\operatorname{tr} \tilde{K}}, \tag{3.5}$$

where $\tilde{K}$ *is the labeled data kernel matrix, which can be written as*

$$\tilde{K} = \sum_{j=1}^{m} \frac{a_j^2}{\gamma_j} K^j - \lambda \underline{K}_\ell^T \underline{A} \underline{G}^{-1} \left(I + \lambda M \underline{G}^{-1} \mathcal{K} \right)^{-1} M \underline{G}^{-1} \underline{A} \underline{K}_\ell \tag{3.6}$$

where

$$\underline{K}_\ell = \left(\underline{\boldsymbol{k}}_{x_1^\ell} \ldots \underline{\boldsymbol{k}}_{x_\ell^\ell} \right) = \begin{pmatrix} k^1(x_1, x_1^\ell) & \cdots & k^1(x_1, x_\ell^\ell) \\ \vdots & & \vdots \\ k^1(x_n, x_1^\ell) & \cdots & k^1(x_n, x_\ell^\ell) \\ \vdots & & \vdots \\ k^m(x_1, x_1^\ell) & \cdots & k^m(x_1, x_\ell^\ell) \\ \vdots & & \vdots \\ k^m(x_n, x_1^\ell) & \cdots & k^m(x_n, x_\ell^\ell) \end{pmatrix}.$$

Proof. By Theorem 3.2.1, we know that

$$\varphi^* = \arg\min_{\varphi \in \tilde{\mathcal{H}}} L(\varphi) + \|\varphi\|^2_{\tilde{\mathcal{H}}}$$

It then follows by a simple argument given in Section A that $\|\varphi^*\|^2_{\tilde{\mathcal{H}}} \leq L(0)$. Thus we can restrict our search for φ^* to the norm ball in $\tilde{\mathcal{H}}$ of radius $\sqrt{L(0)}$. The bounds in Equation (3.5) are a standard result, which we state and prove in Lemma B.2.2. Finally, the expression for the multi-view kernel matrix $\tilde{K}$ is immediate from the explicit form of the kernel function given in Theorem 3.2.1. □

3.3.2 Generalization and Regret Bounds

We can get a basic generalization bound for φ^* by plugging the Rademacher complexity bound given in Theorem 3.3.1 into the generalization bound of Theorem 2.4.2. Another approach is to use the result of Theorem 3.2.1, where we expressed the multi-view point cloud regularization problem as a standard supervised learning problem in a new RKHS. This allows us to apply the large body of theory on RKHS learning methods to our multi-view learning algorithm. As a particular example, we present a bound from the theory of localized Rademacher complexity and discuss how this refines the bound in Theorem 2.4.2.

As in earlier chapters, we consider a labeled training set $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ sampled i.i.d. from $P_{\mathcal{X}\times\mathcal{Y}}$. We now define the loss functional $L : \tilde{\mathcal{H}} \to \mathbf{R}$ as the empirical loss for a loss function $V : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. That is,

$$\hat{L}(\varphi) = \frac{1}{\ell} \sum_{i=1}^{\ell} V(\varphi(x_i), y_i)$$

We will write the empirical risk minimizer as $\hat{\varphi}_* \in \arg\min_{\varphi\in\tilde{\mathcal{H}}_1} \hat{L}(\varphi)$ and the true risk minimizer as $\varphi_* \in \arg\min_{\varphi\in\tilde{\mathcal{H}}_1} L(\varphi)$, where $L(\varphi) := \mathbb{E}\hat{L}(\varphi) = \mathbb{E}V(\varphi(X), Y)$. In the theorem below, we bound the gap between the risk of $\hat{\varphi}_*$ and the risk of φ_*. Since $\hat{\varphi}_*$ depends on the training set, the bound holds with high probability over the training set. We first state some conditions on the loss function V:

Condition 1. The loss $V(\cdot, \cdot)$ is Lipschitz in its first argument, i.e. there exists a constant A such that $\forall y, \hat{y}_1, \hat{y}_2$: $|V(\hat{y}_1, y) - V(\hat{y}_2, y)| \leq A\,|\hat{y}_1 - \hat{y}_2|$.

Condition 2. For any probability distribution $P_{\mathcal{X}\times\mathcal{Y}}$, there exists $\varphi_* \in \tilde{\mathcal{H}}_1$ satisfying $\mathbb{E}V(\varphi_*(X), Y) = \inf_{\varphi\in\tilde{\mathcal{H}}_1} \mathbb{E}V(\varphi(X), Y)$, where $(X, Y) \sim P_{\mathcal{X}\times\mathcal{Y}}$.

Condition 3. There is a constant $B \geq 1$ such that for every probability distribution $P_{\mathcal{X}\times\mathcal{Y}}$ and every $\varphi \in \tilde{\mathcal{H}}_1$ we have,

$$\mathbb{E}(\varphi(X) - \varphi_*(X))^2 \leq B\,\mathbb{E}\left[V[\varphi(X), Y] - V[\varphi_*(X), Y)]\right]$$

In the following theorem, let P_ℓ denote the empirical probability measure for the labeled sample of size ℓ. In the theorem below, we assume that $\sup_{x\in\mathcal{X}} k(x, x) \leq 1$.

By Lemma A.1.4, this implies that the functions in $\tilde{\mathcal{H}}_1$ are uniformly bounded by 1. That is,

$$\sup_{x\in\mathcal{X}} \sup_{\varphi\in\tilde{\mathcal{H}}_1} |\varphi(x)| \leq 1$$

.

Theorem 3.3.2. *[Cor. 6.7 from [3]] Assume that* $\sup_{x\in\mathcal{X}} k(x,x) \leq 1$ *and that* V *satisfies the 3 conditions above. Let* $\hat{\varphi}_*$ *be an empirical risk minimizer, as defined above. There exists a constant* c *depending only on* A *and* B *s.t. with probability at least* $1 - 6e^{-\nu}$*,*

$$\mathbb{E}V[\hat{\varphi}_*(X), Y] - \mathbb{E}V[\varphi_*(X), Y] \leq c\left(\hat{r}^* + \frac{\nu}{\ell}\right),$$

where $\hat{r}^* \leq \min_{0\leq h\leq\ell}\left(\frac{h}{\ell} + \frac{1}{\ell}\sqrt{\sum_{i>h}\lambda_i}\right)$ *and where* $\lambda_1, \ldots, \lambda_\ell$ *are the eigenvalues of the labeled-data kernel matrix* $\tilde{K}$ *in decreasing order.*

Since the localized bound only needs to account for the capacity of the function class in the neighborhood of f_*, the bounds are potentially tighter. Indeed, while the bound in Theorem 2.4.2 is in terms of the trace of the kernel matrix, the bound in Theorem 3.3.2 involves the tail sum of kernel eigenvalues. If the eigenvalues decay very quickly, the latter is potentially much smaller.

3.4 Co-Regularized Semi-Supervised Learning

Here we consider a multi-view generalization of the semi-supervised learning approach presented in Chapter 2. One version of this generalization, presented in [11], extends CoRLS to more than two views by penalizing the disagreement between all pairs of views. They solve the following minimization problem:

$$\underset{(f^1,\ldots,f^m)\in\mathcal{H}^1\times\cdots\times\mathcal{H}^m}{\arg\min} \hat{L}\left(f^1,\ldots,f^m\right) + \sum_{i=1}^{m}\gamma_i\|f^i\|^2_{\mathcal{H}^i} + \lambda\sum_{i,j=1}^{m}\sum_{k=\ell+1}^{\ell+u}[f^i(x_k) - f^j(x_k)]^2. \quad (3.7)$$

Even when using multiple views, we are often ultimately interested in having a single prediction function, rather than a separate function from each view. In [11, 14, 25], for example, the final prediction function is taken to be $\varphi^* = \frac{1}{m}\left(\hat{f}^1_\ell + \cdots \hat{f}^m_\ell\right)$.

In this case, it seems most reasonable to have the loss functional directly measure the performance of φ^*. With this restriction on $\hat{L}$, we can write the objective function in Equation (3.7) in the form of Equation (3.3) of Theorem 3.2.1. Starting with the co-regularization term, we have:

$$\begin{aligned}
\lambda \sum_{i,j=1}^{m} \sum_{k=\ell+1}^{\ell+u} [f^i(x_k) - f^j(x_k)]^2 &= \lambda \sum_{i,j=1}^{m} \sum_{k=\ell+1}^{\ell+u} \left(f^i(x_k)\right)^2 + \left(f^j(x_k)\right)^2 - 2f^i(x_k)f^j(x_k) \\
&= \lambda \underline{\boldsymbol{f}}^T M^{\text{coreg}} \underline{\boldsymbol{f}}
\end{aligned}$$

where M^{coreg} is an $mn \times mn$ matrix defined by

$$M^{\text{coreg}} = \begin{cases} 1 & i = j \\ \text{-1} & |i-j| \in \{n, 2n, \ldots, (m-1)n\} \\ 0 & \text{otherwise} \end{cases}$$

For example, if $m = 3$ and $n = 3$, then we would have

$$M^{\text{coreg}} = \begin{pmatrix}
1 & & & -1 & & & -1 & & \\
& 1 & & & -1 & & & -1 & \\
& & 1 & & & -1 & & & -1 \\
-1 & & & 1 & & & -1 & & \\
& -1 & & & 1 & & & -1 & \\
& & -1 & & & 1 & & & -1 \\
-1 & & & -1 & & & 1 & & \\
& -1 & & & -1 & & & 1 & \\
& & -1 & & & -1 & & & 1
\end{pmatrix} = \begin{pmatrix} I & -I & -I \\ -I & I & -I \\ -I & -I & I \end{pmatrix}$$

In general, M^{coreg} is an $m \times m$ block matrix where the diagonal blocks are $n \times n$ identity matrices, and the off-diagonal blocks are $n \times n$ negative identity matrices.

Thus if we take the unlabeled points $x_{\ell+1}, \ldots, x_{\ell+u}$ to be the "point cloud", and our final prediction function to be $\hat{=}_{\ell} a_1 \hat{f}_\ell^1 + \cdots + a_m \hat{f}_\ell^m$, for some fixed $a_1, \ldots, a_m \in \mathbf{R}$, then we can write the multi-view co-regularization problem as

$$\operatorname*{arg\,min}_{\varphi \in \tilde{\mathcal{H}}} \min_{(f^1,\ldots,f^m) \in u^{-1}(\varphi)} L(\varphi) + \sum_{i=1}^{m} \gamma_i \|f^i\|_{\mathcal{H}^i}^2 + \lambda \underline{\boldsymbol{f}}^T M^{\text{coreg}} \underline{\boldsymbol{f}}$$

Now Theorem 3.2.1 will give the RKHS and corresponding kernel with which this problem can be written as a traditional RKHS regularization problem. For the special case of 2 views, with $a_1 = a_2 = \frac{1}{2}$, this RKHS and kernel were derived by us in [29].

Having expressed this problem in the framework of the theorem, we can apply the Rademacher complexity and generalization bounds of Section 3.3. The Rademacher complexity bound generalizes Theorem 2.4.3 to the case of multiple views, with an arbitrary linear combination of views, rather than a uniform average. With Theorem 3.2.1, the proof of these theorems is straightforward compared to the lengthy "bare-hands" approach used for the proof of Theorem 2.4.3.

Chapter 4

Manifold Co-Regularization

In this chapter, we present the manifold co-regularization (CoMR) algorithm. This algorithm was initially conceived as a "multi-view" version of the manifold regularization (MR) algorithm introduced by Belkin et al. in [6]. We show below that MR is, in fact, a limiting case of CoMR, in a way we make precise below. In Section 4.1, we present the graph transduction algorithm, in which a manifold smoothness constraint is directly enforced on the prediction functions. However, graph transduction is not a true semi-supervised learning algorithm, since it only gives predictions on the unlabeled data points that are part of the training set, and not on other unseen points of the input space. In Section 4.2, we review the MR algorithm, which is a natural generalization of graph transduction to the fully semi-supervised setting. In Section 4.3 we present our CoMR algorithm, and in Section 4.4 we present experimental results suggesting that CoMR often offers a significant performance improvement over MR.

4.1 Graph Transduction

In transductive learning, we are given a training set of labeled and unlabeled points, and the goal is to make label predictions for the unlabeled points. We will denote the *point cloud* of labeled and unlabeled points by $\mathcal{M} = \{x_1, \ldots, x_\ell, x_{\ell+1}, x_{\ell+u}\}$. In graph transduction, we look for a prediction function $f : \mathcal{M} \to \mathbf{R}$ that performs well on the labeled training data and is "smooth" with respect to the point cloud. As discussed in Section 1.5, we can write such a smoothness measure as a quadratic form $\boldsymbol{f}^T M^{\text{intrinsic}} \boldsymbol{f}$, where $M^{\text{intrinsic}} \in \mathbf{R}^{(\ell+u)\times(\ell+u)}$ is a positive semidefinite matrix, such as the graph Laplacian of the data adjacency matrix, and $\boldsymbol{f} \in \mathbf{R}^{\ell+u}$ is the column vector of evaluations of f on $\mathcal{M}$. We write "intrinsic" in the superscript of $M^{\text{intrinsic}}$ for ease of comparison with other methods later in this chapter. The graph transduction objective function is the following:

$$G(f) = \hat{L}(f) + \gamma \boldsymbol{f}^T M^{\text{intrinsic}} \boldsymbol{f}, \tag{4.1}$$

where, as usual, $\hat{L}$ is a loss functional that only depends on its argument f via its evaluations on the labeled data points, and the scalar $\gamma > 0$ is a regularization parameter. The term $\boldsymbol{f}^T M^{\text{intrinsic}} \boldsymbol{f}$ has been called the "graph regularization" term (e.g. [34]) as well as the "intrinsic regularization" term (e.g. [6]).

The problem with this transductive approach is that there is no clear "out of sample extension" for a prediction function $f : \mathcal{M} \to \mathbf{R}$. That is, if we want to make predictions on new points, not contained in $\mathcal{M}$, it is not clear how to avoid adding the points to $\mathcal{M}$, generating a new matrix $M^{\text{intrinsic}}$, and solving a new minimization problem. We would prefer an "inductive" solution, in which the semi-supervised learning algorithm produces a function $f : \mathcal{X} \to \mathbf{R}$ defined on the whole input space $\mathcal{X}$. One of the cleanest methods to get an out-of-sample extension for graph transduction is the manifold regularization approach, which was introduced in [27] and [6], and which we review in the next section.

4.2 Manifold Regularization

Let $\mathcal{H}$ be an RKHS of functions from $\mathcal{X} \to \mathbf{R}$. Following [5], we call $\mathcal{X}$ the *ambient space* and the set of training points $\mathcal{M}$ the *intrinsic space*. Recall that in the graph regularization objective function, we have a labeled loss functional, which measures the fit to the labeled data, and the intrinsic regularization term, which measures the smoothness of the function in the intrinsic space $\mathcal{M}$. In manifold regularization, we add an additional term that ensures that the chosen f has a smooth extension from $\mathcal{M}$ to the ambient space $\mathcal{X}$. This term is just the squared RKHS norm of f in the ambient space $\mathcal{H}$. Thus in manifold regularization we search for a function $\hat{f}_\ell \in \mathcal{H}$ solving

$$\arg\min_{f \in \mathcal{H}} \hat{L}(f) + \gamma_{\mathcal{A}} \|f\|_{\mathcal{H}}^2 + \gamma_{\mathcal{I}} \boldsymbol{f}^T M^{\text{intrinsic}} \boldsymbol{f}, \tag{4.2}$$

for regularization parameters $\gamma_{\mathcal{A}} > 0$ and $\gamma_{\mathcal{I}} \geq 0$. Essentially, the size of the "ambient norm" $\|\cdot\|_{\mathcal{H}}$ is used to choose among all functions that perform well in the graph transduction sense, i.e. for which $G(f)$ is small. With nice loss functionals, such as the hinge loss and square loss, solving this optimization is quite easy (c.f. Appendix A.4).

We note that manifold regularization is a special case of the multi-view point cloud regularization problem, presented in Chapter 3, in which we have a single view. Let us write K for the kernel matrix on the point cloud and $\boldsymbol{k}_x$ for the column vector of kernel evaluations of the form $k(x,p)$, where p ranges over the points in the point cloud. Then by Theorem 3.2.1, the *manifold regularization kernel* is given by

$$\tilde{k}(z,x) = \frac{1}{\gamma_{\mathcal{A}}} k(z,x) - \frac{1}{\gamma_{\mathcal{A}}^2} \gamma_{\mathcal{I}} \boldsymbol{k}_x^T \left(I + \frac{\gamma_{\mathcal{I}}}{\gamma_{\mathcal{A}}} M^{\text{intrinsic}} K \right)^{-1} M^{\text{intrinsic}} \boldsymbol{k}_z,$$

which recovers the result of Proposition 2.2 from [27].

4.3 Manifold Co-Regularization

In this section, we present the *manifold co-regularization* (CoMR) algorithm. CoMR is a multi-view approach to manifold regularization. Roughly speaking, manifold co-regularization is defined as co-regularization with a particular choice of views, called the ambient view and the intrinsic view. We take the *ambient view*, denoted by $\mathcal{H}^{\mathcal{A}}$, to be an RKHS of functions defined on the input space $\mathcal{X}$. We measure the "smoothness" of a function $f \in \mathcal{H}^{\mathcal{A}}$ using the RKHS norm $\|f\|_{\mathcal{H}^{\mathcal{A}}}$. The *intrinsic view*, denoted by $\mathcal{H}^{\mathcal{I}}$, comprises functions whose domain is restricted to the data manifold $\mathcal{M} = \{x_1, \ldots, x_\ell, x_{\ell+1}, x_{\ell+u}\}$. The measure of smoothness for a function $f \in \mathcal{H}^{\mathcal{I}}$ is taken to be $\sqrt{S(f)}$, where $S(f)$ is an incompatibility function measuring manifold smoothness, such as one of those presented in Section 1.5 above. In particular, we assume that $S(f)$ has the form $S(f) = \boldsymbol{f}^T M^{\text{intrinsic}} \boldsymbol{f}$, where $M^{\text{intrinsic}}$ is a positive semidefinite matrix.

4.3.1 Transductive CoMR

In the version of co-regularization discussed in Chapter 2, the final prediction function is taken to be the average of the prediction functions chosen from each view. We can write this as $(f^{\mathrm{a}} + f^{\mathrm{i}})/2$, where $f^{\mathrm{a}} \in \mathcal{H}^{\mathcal{A}}$ and $f^{\mathrm{i}} \in \mathcal{H}^{\mathcal{I}}$. However, since the domain of f^{i} is restricted to the data manifold $\mathcal{M}$, the final prediction function will not be defined on "out-of-sample" points of $\mathcal{X}$. While $(f^{\mathrm{a}} + f^{\mathrm{i}})/2$ is a very reasonable prediction function for the *transductive* setting, where we need only provide predictions on $\mathcal{M}$, it is not a fully semi-supervised solution. In our first description of the CoMR algorithm in [29], we used f^{a} alone to make out-of-sample predictions. However, to fit the algorithm into the multi-view RKHS framework of that paper, our objective function was minimizing the loss of $(f^{\mathrm{a}} + f^{\mathrm{i}})/2$ rather than the loss of f^{a}, which would be more consistent with the final prediction function. Part of the work in Chapter 3 was motivated by the desire to build a multi-view framework that would allow the loss to depend on more general combinations of f^{a} and f^{i}. Below, we attain a more satisfying version of semi-supervised CoMR using the more general framework of Chapter 3.

4.3.2 General CoMR

We first present the most general form of the CoMR objective function. It accommodates both the transductive and semi-supervised formulations. Specializing the two-view co-regularization formulation of Section 3.4, we define the view combination function as $u(f^{\mathrm{a}}, f^{\mathrm{i}}) = a^{\mathrm{a}} f^{\mathrm{a}} + a^{\mathrm{i}} f^{\mathrm{i}}$, for any $a^{\mathrm{a}}, a^{\mathrm{i}} \in \mathbf{R}$, and we define the space of final prediction functions as $\tilde{\mathcal{H}} = \left\{ u(f^{\mathrm{a}}, f^{\mathrm{i}}) \mid f^{\mathrm{a}} \in \mathcal{H}^{\mathcal{A}}, f^{\mathrm{i}} \in \mathcal{H}^{\mathcal{I}}) \right\}$. Then the CoMR optimization problem is written as:

$$\operatorname*{arg\,min}_{\varphi \in \tilde{\mathcal{H}}} \min_{(f^{\mathrm{a}}, f^{\mathrm{i}}) \in u^{-1}(\varphi)} \hat{L}(\varphi) + \gamma_{\mathcal{A}} \|f^{\mathrm{a}}\|^2_{\mathcal{H}^{\mathcal{A}}} + \gamma_{\mathcal{I}} \left(\boldsymbol{f}^{\mathrm{i}}\right)^T M^{\mathrm{intrinsic}} \boldsymbol{f}^{\mathrm{i}} + \lambda \underline{\boldsymbol{f}}^T M^{\mathrm{coreg}} \underline{\boldsymbol{f}}$$

where

$$\underline{\boldsymbol{f}} = \begin{pmatrix} \boldsymbol{f}^{\mathrm{a}} \\ \boldsymbol{f}^{\mathrm{i}} \end{pmatrix} = \left(f^{\mathrm{a}}(x_1), \ldots, f^{\mathrm{a}}(x_n), f^{\mathrm{i}}(x_1), \ldots, f^{\mathrm{i}}(x_n)\right)^T$$

and where M^{coreg} is the co-regularization matrix for two-views and a point cloud of size n, as defined in Section 3.4.

By adding a small multiple of the identity matrix to $M^{\text{intrinsic}}$, if necessary, we may assume that $M^{\text{intrinsic}}$ is invertible. In this case, $\mathcal{H}^{\mathcal{I}}$ is a finite-dimensional RKHS with norm $\sqrt{S(f)}$ and kernel function $k : \mathcal{M} \times \mathcal{M} \to \mathbf{R}$ given by the matrix $(M^{\text{intrinsic}})^{-1}$. Thus our two views are proper RKHSs, and we may directly apply the theory from Chapter 3. In particular, Theorem 3.3.1 gives a CoMR kernel and associated RKHS, in which we can compute the CoMR solution by optimizing in a single RKHS.

4.3.3 Semi-Supervised CoMR

In semi-supervised CoMR, we take $a^{\text{a}} = 1$ and $a^{\text{i}} = 0$ so that the domain of the final prediction function is the entire input space $\mathcal{X}$. In this case, we can reduce the objective function to involve only a single view without assuming that $M^{\text{intrinsic}}$ is invertible. We can also show that manifold regularization is a limiting case of semi-supervised CoMR. Noting that $\underline{\boldsymbol{f}}^T M^{\text{coreg}} \underline{\boldsymbol{f}} = 2(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}})^T(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}})$, we can write the semi-supervised CoMR objective function as follows:

$$\operatorname*{arg\,min}_{f^{\text{a}} \in \mathcal{H}^{\mathcal{A}}} \min_{f^{\text{i}} \in \mathcal{H}^{\mathcal{I}}} \hat{L}(f^{\text{a}}) + \gamma_{\mathcal{A}} \|f^{\text{a}}\|^2_{\mathcal{H}^{\mathcal{A}}} + \gamma_{\mathcal{I}} \left(\boldsymbol{f}^{\text{i}}\right)^T M^{\text{intrinsic}} \boldsymbol{f}^{\text{i}} + 2\lambda(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}})^T(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}}).$$

Since the terms depending on f^{i} in the manifold co-regularization objective function are quadratic, we can work out the minimum in closed form. The terms involving f^{i} are:

$$\begin{aligned} & \gamma_{\mathcal{I}} \left(\boldsymbol{f}^{\text{i}}\right)^T M^{\text{intrinsic}} \boldsymbol{f}^{\text{i}} + 2\lambda(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}})^T(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}}) \\ = \; & \left(\boldsymbol{f}^{\text{i}}\right)^T \left(2\lambda I + \gamma_{\mathcal{I}} M^{\text{intrinsic}}\right) \boldsymbol{f}^{\text{i}} + 2\lambda \left(\boldsymbol{f}^{\text{a}}\right)^T \boldsymbol{f}^{\text{a}} - 4\lambda \left(\boldsymbol{f}^{\text{a}}\right)^T \boldsymbol{f}^{\text{i}} \end{aligned}$$

By differentiation or completing the quadratic form, it is easy to show that a minimum of this quadratic function is attained at:

$$\begin{aligned} \boldsymbol{f}^{\text{i}}_* &= \operatorname*{arg\,min}_{f^{\text{i}} \in \mathcal{H}^{\mathcal{I}}} \gamma_{\mathcal{I}} \left(\boldsymbol{f}^{\text{i}}\right)^T M^{\text{intrinsic}} \boldsymbol{f}^{\text{i}} + 2\lambda(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}})^T(\boldsymbol{f}^{\text{i}} - \boldsymbol{f}^{\text{a}}) \\ &= \left(I + \frac{\gamma_{\mathcal{I}}}{2\lambda} M^{\text{intrinsic}}\right)^{-1} \boldsymbol{f}^{\text{a}}, \end{aligned}$$

where we have assumed that $\lambda > 0$ to ensure invertibility. Note that

$$\lim_{\lambda\to\infty} \boldsymbol{f}^{\mathrm{i}}_* = \boldsymbol{f}^{\mathrm{a}}$$

This is as expected, since larger λ puts more weight on the L^2 disagreement penalty between the two views.

Plugging in the optimal $\boldsymbol{f}^{\mathrm{i}}_*$, we can now find the optimal value of those terms in the objective function involving $\boldsymbol{f}^{\mathrm{i}}$. To shorten expressions, let $R = \left(I + \frac{\gamma_{\mathcal{I}}}{2\lambda} M^{\mathrm{intrinsic}}\right)^{-1}$. Then $\boldsymbol{f}^{\mathrm{i}}_* = \left(I + \frac{\gamma_{\mathcal{I}}}{2\lambda} M^{\mathrm{intrinsic}}\right)^{-1} \boldsymbol{f}^{\mathrm{a}} = R\boldsymbol{f}^{\mathrm{a}}$. For any fixed value of $\boldsymbol{f}^{\mathrm{a}}$, we have

$$\begin{aligned}
& \min_{f^{\mathrm{i}}\in\mathcal{H}^{\mathcal{I}}} \gamma_{\mathcal{I}} \left(\boldsymbol{f}^{\mathrm{i}}\right)^T M^{\mathrm{intrinsic}} \boldsymbol{f}^{\mathrm{i}} + 2\lambda(\boldsymbol{f}^{\mathrm{i}} - \boldsymbol{f}^{\mathrm{a}})^T(\boldsymbol{f}^{\mathrm{i}} - \boldsymbol{f}^{\mathrm{a}}) \\
= & \min_{f^{\mathrm{i}}\in\mathcal{H}^{\mathcal{I}}} \left(\boldsymbol{f}^{\mathrm{i}}\right)^T \left(2\lambda I + \gamma_{\mathcal{I}} M^{\mathrm{intrinsic}}\right) \boldsymbol{f}^{\mathrm{i}} + 2\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T \boldsymbol{f}^{\mathrm{a}} - 4\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T \boldsymbol{f}^{\mathrm{i}} \\
= & \left(R\boldsymbol{f}^{\mathrm{a}}\right)^T \left(2\lambda R^{-1}\right) R\boldsymbol{f}^{\mathrm{a}} + 2\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T \boldsymbol{f}^{\mathrm{a}} - 4\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T R\boldsymbol{f}^{\mathrm{a}} \\
= & -2\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T R\boldsymbol{f}^{\mathrm{a}} + 2\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T \boldsymbol{f}^{\mathrm{a}} \\
= & 2\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T \left(I - R\right) \boldsymbol{f}^{\mathrm{a}} \\
= & 2\lambda \left(\boldsymbol{f}^{\mathrm{a}}\right)^T \left(R^{-1} - I\right) R\boldsymbol{f}^{\mathrm{a}} \\
= & \left(\boldsymbol{f}^{\mathrm{a}}\right)^T M^{\mathrm{intrinsic}} \left(\frac{1}{\gamma_{\mathcal{I}}} I + \frac{1}{2\lambda} M^{\mathrm{intrinsic}}\right)^{-1} \boldsymbol{f}^{\mathrm{a}}
\end{aligned}$$

Thus an equivalent expression for the semi-supervised CoMR optimization problem is:

$$\arg\min_{f^{\mathrm{a}}\in\mathcal{H}^{\mathcal{A}}} \hat{L}(f^{\mathrm{a}}) + \gamma_{\mathcal{A}} \|f^{\mathrm{a}}\|^2_{\mathcal{H}^{\mathcal{A}}} + \gamma_{\mathcal{I}} \left(\boldsymbol{f}^{\mathrm{a}}\right)^T M^{\mathrm{intrinsic}} \left(I + \frac{\gamma_{\mathcal{I}}}{2\lambda} M^{\mathrm{intrinsic}}\right)^{-1} \boldsymbol{f}^{\mathrm{a}}. \tag{4.3}$$

When we take $\lambda \to \infty$, the objective function reduces to the manifold regularization problem, as given in Equation (4.2). As $\lambda \to 0$, the dependence on the unlabeled data vanishes as the last term in the objective function goes to 0. Thus λ mediates between purely supervised learning at one limit, and manifold regularization at the other limit. Note also that since the matrix in the last term of Equation (4.3) is positive semidefinite, we may view the last term as an intrinsic regularization term. Thus semi-supervised CoMR may also be viewed as MR with a modified smoothness measure. The ratio $\gamma_{\mathcal{I}}/\lambda$ governs whether the intrinsic regularization term is more like the original MR term (when $\gamma_{\mathcal{I}}/\lambda$ is small) or when it is more like a Euclidean norm penalty[1] on $\boldsymbol{f}^{\mathrm{a}}$ (when $\gamma_{\mathcal{I}}/\lambda$ is large).

4.4 Experiments

In this section, we compare the empirical performance of MR and CoMR on several data sets. These experiments build on the earlier work of Sindhwani et al. in [27]. In particular, we use their general experimental design and also treat as fixed certain parameter values that they found worked well for MR. The experiments described in this section were originally presented in [29].

We performed semi-supervised classification experiments on the five data sets described in Table 4.1. The LINES data set is a variant of the two-dimensional problem shown in Figure 4.1, where we added random noise around the two perpendicular lines. The G50C, USPST, COIL20, and PCMAC data sets are well-known and have frequently been used for empirical studies in semi-supervised learning literature. They were used for benchmarking manifold regularization in [27] against a number of competing methods. G50C is an artificial data set generated from two unit covariance normal distributions with equal probabilities. The class means are adjusted so that the Bayes error is 5%. COIL20 consists of 32 × 32 gray scale images of 20 objects viewed from varying angles. USPST is taken from the test subset of the USPS data set of images containing 10 classes of handwritten digits. PCMAC is a binary text categorization problems drawn from the 20-newsgroups data set.

4.4.1 Protocol

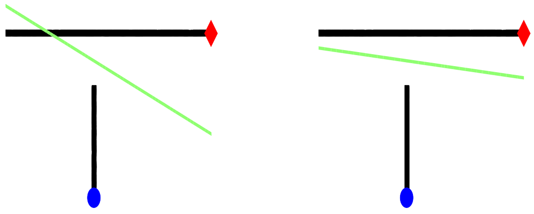
For each of the 5 data sets, we constructed 10 random splits of the data into labeled, unlabeled, test, and validation sets. The sizes of these sets are given in Table 4.1. For all data sets except LINES, the ambient view $\mathcal{H}^{\mathcal{A}}$ was taken to be an RKHS of functions mapping from the full input space (i.e. we do not do any feature

[1] More precisely, this holds when $M^{\mathrm{intrinsic}}$ has full rank. Otherwise, when $\gamma_{\mathcal{I}}/\lambda$ is small we get a Euclidean norm penalty on the projection of f^{a} onto the column space of $M^{\mathrm{intrinsic}}$ and a manifold regularization-type penalty on the projection into the null space.

Table 4.1: Data sets with d features and c classes. 10 random data splits were created with ℓ labeled, u unlabeled, t test, and v validation examples.

Data set	d	c	l	u	t	v
LINES	2	2	2	500	250	250
G50C	50	2	50	338	112	50
USPST	256	10	50	1430	477	50
COIL20	241	20	40	1320	40	40
PCMAC	7511	2	50	1385	461	50

Figure 4.1: Decision boundaries of MR and CoMR on the LINES data set.



(a) MR

(b) CoMR

splitting) into $\mathbf{R}$. We take the RKHS to have a Gaussian kernel with fixed covariance. For the LINES data set, took $\mathcal{H}^{\mathcal{A}}$ to be an RKHS with the standard Euclidean inner product as the kernel function.

For the graph regularization term, we follow the approach described in Section 1.5.1: We first define the point-cloud similarity matrix on the $n = \ell + u$ training points as follows:

$$W_{ij} = \begin{cases} \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2\right) & \text{when } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are } k\text{-nearest neighbors, and} \\ 0 & \text{otherwise,} \end{cases}$$

where we took the bandwidth parameter to be $\sigma^2 = 1$. We then define the normalized Laplacian, $L = I - D^{-1/2} W D^{-1/2}$, where D is the diagonal matrix whose i'th entry is $D_{ii} = \sum_{j=1}^{n} W_{ij}$. Next we "iterate" the normalized Laplacian by raising it to some positive integer power p. Finally, we add a small diagonal matrix for numerical stability. We write the resulting matrix as

$$M^{\text{intrinsic}} = \left(10^{-6} I + L^p\right).$$

The parameters k and p used for each data set are tabulated in Table 4.2 for reproducibility. The choice of these parameters was based on the work of [27], where the same data sets were used in a similar experimental setup.

For both MR and CoMR, we took the loss function to be squared loss:

$$V(y, \hat{y}) = (y - \hat{y})^2.$$

Manifold regularization with this choice of loss is also referred to as Laplacian regularized least squares (LapRLS), which was introduced by Belkin et al. in [5]. For multi-class problems, we used the one-versus-rest strategy.

Parameter Selection For CoMR, the co-regularization parameter λ was fixed for all experiments to the value 1. With this fixed, both MR and CoMR have two parameters to set: $\gamma_{\mathcal{A}}$ and $\gamma_{\mathcal{I}}$. These parameters were varied on a grid of values, $10^{-6}, 10^{-4}, 10^{-2}, 1, 10, 100$, and were chosen based on their performance on the validation sets, averaged across the 10 random splits. The chosen parameters are reported

Table 4.2: Parameters used for MR and CoMR. Note $\lambda = 1$ for CoMR. Linear kernel was used for the LINES data set.

Data Set	nn	σ	p	MR	CoMR
				γ_1, γ_2	γ_1, γ_2
LINES	10	–	1	$0.01, 10^{-6}$	$10^{-4}, 100$
G50C	50	17.5	5	$1, 100$	$10, 10$
USPST	10	9.4	2	$0.01, 0.01$	$10^{-6}, 10^{-4}$
COIL20	2	0.6	1	10^{-4},10^{-6}	$10^{-6}, 10^{-6}$
PCMAC	50	2.7	5	$10, 100$	$1, 10$

Table 4.3: MR and CoMR error rates (in percentage) on test data.

Data Set	MR	CoMR
LINES	7.7 (1.2)	**1.0** (1.5)
G50C	5.8 (2.8)	5.5 (2.3)
USPST	18.2 (1.5)	**14.1** (1.6)
COIL20	23.8 (11.1)	**14.8** (8.8)
PCMAC	11.9 (3.4)	**8.9** (2.6)

in Table 4.2. With the chosen parameter values, we evaluated the MR and CoMR solutions on both the unlabeled training data and the unseen test data, for each split.

4.4.2 Results

We found that CoMR gives significant empirical improvements over MR on all data sets except G50C, where both methods approach the Bayes error rate. In Tables 4.3 and 4.4, we report the mean and standard deviation of error rates on unlabeled and test examples across the 10 random splits. We performed a paired t-test to compare the relative performance of MR and CoMR across the 10 splits. Results for the data sets showing a significant difference at the 5% level are shown in bold.

Table 4.4: MR and CoMR error rates (in percentage) on unlabeled data.

Data Set	MR	CoMR
LINES	7.5 (1.0)	**1.3** (2.0)
G50C	6.6 (0.8)	6.9 (1.0)
USPST	18.6 (1.4)	**13.3** (1.0)
COIL20	37.5 (6.0)	**14.8** (3.3)
PCMAC	11.0 (2.4)	**9.4** (1.9)

Chapter 5

Conclusions and Future Directions

5.1 Summary of Contributions

This thesis makes several contributions. In Chapter 2, we considered the popular CoRLS algorithm and gave the first precise statement of how much co-regularization reduces the complexity of a function class. We showed that the amount of the reduction in Rademacher complexity is related to how different the two views are in their representations of the labeled training points. Using this result, we also provided generalization bounds for CoRLS prediction functions. Our experimental results suggested a correlation between the reduction in complexity and a performance improvement, as one introduces co-regularization.

In Chapter 3, we introduced a learning framework called multi-view point cloud regularization. This framework has CoRLS and manifold regularization as special cases. The main theorem of this chapter shows that the objective function over multiple views can be rewritten as an objective function over a single RKHS, with a new "multi-view kernel". This result generalizes an earlier result of Sindhwani et al. for the case of manifold regularization [27] as well as our result for the case of two-view co-regularization [29]. With this reformulation of the multi-view learning problem in terms of a standard RKHS regularization problem, we were easily able to generalize our Rademacher complexity bounds of Chapter 2 to the more general setting of Chapter 3. We were also able to give more refined generalization bounds by applying the theory of localized Rademacher complexity.

In Chapter 4, we presented manifold co-regularization (CoMR), which fits nicely into the multi-view point cloud setting of Chapter 4. We showed that the CoMR algorithm performed significantly better than the standard MR algorithm on several data sets. While our motivation for developing CoMR was to give a multi-view version of manifold regularization, we showed that CoMR can itself be seen as a manifold regularization algorithm with a modified graph regularization term. In future work, it will be interesting to make a more careful comparison of the new "CoMR graph regularization" term to the more traditional graph penalty terms, such as the normalized Laplacian and the Hessian, as discussed in Section 1.5.1.

5.2 Future Work

5.2.1 Model Selection in Semi-Supervised Learning

The multi-view algorithms presented in this thesis have several regularization parameters. While this is fairly common for machine learning algorithms, it is of special concern in semi-supervised learning contexts. A typical approach to choosing parameters is simply to try a wide range of parameter values and estimate their performance using a held out validation set or cross-validation. However, when we only have a limited number of labeled data points to work with, one needs to take care not to overfit the validation set. For future work, it would seem more reasonable to consider "parameter-free" algorithms, in which all parameter selection methods are internal to the algorithm. This would highlight the important practical issue of how to best use a fixed amount of labeled data to both train a model and tune parameters. In our experiments from Section 4.4, for all but one of the data sets we took our validation set to be the same size as our labeled training set. Splitting a fixed amount of labeled training data evenly into training and validation sets is unlikely to be the most efficient use of the data.

5.2.2 Choosing Multiple Views

A typical concern in applying co-training and co-regularization methods is how to choose the views. If the input features do not split naturally into multiple views, one common approach is to choose a random split of the features, as we did in the experiments of Section 2.7. However, this randomization may lead to a high level of variance in performance results. Another approach, using the multi-view co-regularization setup of Section 3.4, would be to choose a large number of random subsets of features and make a view corresponding to each subset. The co-regularization then encourages all chosen prediction functions to agree. This should reduce the variance of the final prediction function, though some experiments would be required to determine the effect on performance. From a theoretical standpoint, one could consider the limiting case in which we have d features and we construct $\binom{d}{k}$ views, one for each subset of k features.

5.2.3 Generative Models for Multiple Views

It is often the case that the solution to a regularization problem is also the MAP estimate for a parameter of a Bayesian model, where the regularization term corresponds to the prior distribution on the parameter. One interesting direction for future work is to find a Bayesian model that corresponds in the same way to our multi-view point cloud regularization framework. One advantage of having a fully generative model for the data is that it is very natural to update one's parameter estimates in an online fashion, as new data come in. Some work in this direction was done by Yu et al. in [33]. However, their model was entirely conditional on the observed data, and did not maintain a generative model for new data points. Thus one cannot naturally use their model in an online setting as described above.

Bibliography

[1] Maria F. 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.

[2] Maria-Florina Balcan and Avrim Blum. A PAC-style model for learning from labeled and unlabeled data. In *COLT*, volume 3559, pages 111–126, June 2005.

[3] Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[4] Suzanna Becker and Geoffrey E. Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, January 1992.

[5] M. Belkin, P. Niyogi, and V. Sindhwani. On Manifold Regularization. In *AISTAT*, 2005.

[6] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[7] A. Bertinet and Thomas C. Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

[8] Rajendra Bhatia. Eigenvalues of AB and BA. *Resonance*, 7(1):88–93, January 2002.

[9] Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[10] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[11] Ulf Brefeld, Thomas Gärtner, Tobias Scheffer, and Stefan Wrobel. Efficient co-regularised least squares regression. In *International Conference on Machine Learning*, volume 23, 2006.

[12] Sanjoy Dasgupta, Michael L. Littman, and David A. Mcallester. PAC generalization bounds for co-training. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *NIPS*, pages 375–382. MIT Press, 2001.

[13] Virginia R. de Sa. Learning classification with unlabeled data. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Proc. NIPS'93, Neural Information Processing Systems*, pages 112–119, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[14] Jason D. R. Farquhar, David R. Hardoon, Hongying Meng, John S. Taylor, and Sándor Szedmák. Two view learning: SVM-2K, theory and practice. In *NIPS*, 2005.

[15] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, October 1996.

[16] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[17] Rie Johnson and Tong Zhang. On the effectiveness of laplacian normalization for graph semi-supervised learning. *Journal of Machine Learning Research*, 8:1489–1517, July 2007.

[18] 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.

[19] Rafal Latala and Krzysztof Oleszkiewicz. On the best constant in the Khintchine-Kahane inequality. *Studia Mathematica*, 109(1):101–104, 1994.

[20] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 1991.

[21] David G. Luenberger. *Optimization by Vector Space Methods (Series in Decision and Control)*. Wiley-Interscience, January 1997.

[22] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *J. Mach. Learn. Res.*, 7:2651–2667, 2006.

[23] Tom Mitchell. Machine learning and extracting information from the web. NAACL Invited Talk, 2001.

[24] David S. Rosenberg. Kernel co-training: An algorithm for semi-supervised learning. Statistics Department Student Seminar Talk, University of California at Berkeley, January 2006.

[25] David S. Rosenberg and Peter L. Bartlett. The Rademacher complexity of co-regularized kernel classes. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, March 2007.

[26] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, June 2004.

[27] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pages 824–831. ACM, 2005.

[28] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *International Conference on Machine Learning*, 2005.

[29] Vikas Sindhwani and David S. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. In *International Conference on Machine Learning*, July 2008.

[30] Karthik Sridharan and Sham M. Kakade. An information theoretic framework for multi-view learning. In Rocco A. Servedio, Tong Zhang, Rocco A. Servedio, and Tong Zhang, editors, *COLT*, pages 403–414. Omnipress, 2008.

[31] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, November 2001.

[32] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[33] Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, and Bharat R. Rao. Bayesian co-training. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1665–1672. MIT Press, Cambridge, MA, 2008.

[34] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Appendix A

Reproducing Kernel Hilbert Spaces

A.1 Definition and Basic Properties

Definition A.1.1 (RKHS). *A reproducing kernel Hilbert space (RKHS) of functions from $\mathcal{X}$ to $\mathbf{R}$ is a Hilbert Space $\mathcal{H}$ that possesses a reproducing kernel, i.e. a function $k : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ for which the following properties hold:*

1. $k(x, .) \in \mathcal{H}$ *for all* $x \in \mathcal{X}$*, and*
2. $\langle f, k(x, .)\rangle_{\mathcal{H}} = f(x)$*, for all* $x \in \mathcal{X}$ *and* $f \in \mathcal{H}$*, where* $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ *denotes the inner product in* $\mathcal{H}$*.*

An *evaluation functional* is a functional that maps a function $f \in \mathcal{H}$ to its evaluation at some fixed point, such as $f \mapsto f(x)$. An RKHS is sometimes defined as a Hilbert space for which all evaluation functionals are continuous. Here we show that this is a consequence of our Definition A.1.1:

Proposition A.1.2. *Any evaluation functional* $f \mapsto f(x)$ *on an RKHS* $\mathcal{H}$ *is uniformly continuous.*

Proof. For any $f, g \in \mathcal{H}$, we have

$$\begin{aligned} |f(x) - g(x)| &= |\langle f, k(x,\cdot)\rangle_{\mathcal{H}} - \langle g, k(x,\cdot)\rangle_{\mathcal{H}}| \\ &= |\langle f - g, k(x,\cdot)\rangle_{\mathcal{H}}| \\ &\leq \|f - g\|_{\mathcal{H}}\|k(x,\cdot)\|_{\mathcal{H}} \text{ by Cauchy Schwartz} \\ &= \|f - g\|_{\mathcal{H}}\sqrt{\langle k(x,\cdot), k(x,\cdot)\rangle_{\mathcal{H}}} \\ &= \|f - g\|_{\mathcal{H}}\sqrt{k(x,x)} \end{aligned}$$

Thus for all $f, g \in \mathcal{H}$ for which $\|f-g\|_{\mathcal{H}} \leq \varepsilon/\sqrt{k(x,x)}$, we have $|f(x)-g(x)| \leq \varepsilon$. □

Proposition A.1.3. *The reproducing kernel $k(\cdot,\cdot)$ for an RKHS $\mathcal{H}$ is positive definite, which means that for any $n = 1, 2, 3, \ldots$, and any choice of points $x_1, \ldots, x_n \in \mathcal{X}$, the kernel matrix $K = (k(x_i, x_j))_{i,j=1}^n$ is positive semidefinite.*

Proof. Fix any $n = 1, 2, 3, \ldots$ and any choice of points $x_1, \ldots, x_n \in \mathcal{X}$. For any $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbf{R}^n$, we have

$$0 \leq \left|\left|\sum \alpha_i k(x_i,\cdot)\right|\right|_{\mathcal{H}}^2 = \sum_{i,j=1}^n \alpha_i\alpha_j k(x_i, x_j) = \boldsymbol{\alpha}' K \boldsymbol{\alpha}$$

Thus K is positive semidefinite. □

Lemma A.1.4. *Let $\mathcal{H}$ be an RKHS of functions mapping $\mathcal{X} \to \mathbf{R}$, having reproducing kernel $k(\cdot,\cdot)$. Fix any $z \in \mathcal{X}$, and let $k_z = k(z,\cdot) \in \mathcal{H}$. Then*

$$\begin{aligned} \sup_{f:\|f\|_{\mathcal{H}}\leq 1} |f(z)| &= \sup_{\|f\|_{\mathcal{H}}\leq 1} |\langle f, k_z\rangle| \\ &= \langle k_z, k_z/\|k_z\|_{\mathcal{H}}\rangle = \frac{1}{\|k_z\|_{\mathcal{H}}} k(z,z) \\ &= \sqrt{k(z,z)} \end{aligned}$$

A.2 Projections

The Hilbert projection theorem (see e.g. [21, p. 51]) states that for any f in a Hilbert space $\mathcal{H}$, and for any closed subspace $\mathcal{L} \subset \mathcal{H}$, there exists a unique element

$f_{\|} \in \mathcal{L}$, called the *projection* of f onto $\mathcal{L}$, such that $\|f - f_{\|}\|_{\mathcal{H}} \leq \|f - g\|_{\mathcal{H}}$ for every $g \in \mathcal{L}$. We will denote the projection of f onto $\mathcal{L}$ by $\text{Proj}_{\mathcal{L}} f$. A necessary and sufficient condition for $f_{\|} \in \mathcal{L}$ to be the projection of f onto $\mathcal{L}$ is that $\langle f - f_{\|}, g \rangle = 0$ for every $g \in \mathcal{L}$. A simple corollary of this characterization is that the norm of $f_{\|}$ never exceeds the norm of f:

Lemma A.2.1. *For any f in a Hilbert space $\mathcal{H}$, let $f_{\|} = Proj_{\mathcal{L}} f$, where $\mathcal{L}$ is a closed subspace. Then*

$$\|f_{\|}\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}}.$$

Proof. We have $f = f_{\|} + (f - f_{\|})$, and $\langle f_{\|}, f - f_{\|} \rangle = 0$. Thus

$$\|f\|_{\mathcal{H}}^2 = \|f_{\|}\|_{\mathcal{H}}^2 + \|f - f_{\|}\|_{\mathcal{H}}^2 + 2\langle f_{\|}, f - f_{\|} \rangle_{\mathcal{H}} \geq \|f_{\|}\|_{\mathcal{H}}^2.$$

□

When the Hilbert space $\mathcal{H}$ is an RKHS of functions mapping from $\mathcal{X} \to \mathbf{R}$, we have the following interesting result:

Lemma A.2.2. *Let $\mathcal{H}$ be an RKHS with kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$, and consider any point $x \in \mathcal{X}$. If $\mathcal{L} \subset \mathcal{H}$ is a closed subspace containing $k(x, \cdot)$, then the projection of f onto $\mathcal{L}$ has the same value at x as f does. That is,*

$$f(x) = \left(Proj_{\mathcal{L}} f\right)(x)$$

Proof. By the Hilbert projection theorem, there exists a unique element $f_{\|} = \text{Proj}_{\mathcal{L}} f \in \mathcal{L}$ such that $\left\langle f - f_{\|}, g \right\rangle_{\mathcal{H}} = 0$ for all $g \in \mathcal{L}$. By definition of RKHS, we have $f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$. Combining these facts, we get

$$\begin{aligned} f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}} &= \left\langle f_{\|} + f - f_{\|}, k(x, \cdot) \right\rangle = \left\langle f_{\|}, k(x, \cdot) \right\rangle + \left\langle f - f_{\|}, k(x, \cdot) \right\rangle \\ &= \left\langle f_{\|}, k(x, \cdot) \right\rangle = f_{\|}(x) \end{aligned}$$

□

A.3 The Representer Theorem and the "Span of the Data"

In practice, many RKHS optimization problems refer to a fixed set of points $x_1, \ldots, x_n \in \mathcal{X}$, and the solution to the optimization problem is contained in a special subspace of the RKHS, often referred to informally as the "span of the data." For an RKHS $\mathcal{H}$ with kernel $k(\cdot, \cdot)$, the *span of the data* is the linear subspace

$$\mathcal{L} = \operatorname{span}\left\{k(x_1, \cdot), \ldots, k(x_n, \cdot)\right\}.$$

We now state and prove the representer theorem, which gives some conditions under which the solution to an RKHS optimization problem is contained in the span of the data.

Theorem A.3.1 (Representer Theorem)**.** *Let $\mathcal{H}$ be an RKHS with kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$. Fix any function $V : \mathbf{R}^n \to \mathbf{R}$ and any nondecreasing function $\Omega : \mathbf{R} \to \mathbf{R}$. Define*

$$J(f) = V\left(f(x_1), \ldots, f(x_n)\right) + \Omega\left(\|f\|_{\mathcal{H}}^2\right)$$

Let $\mathcal{L} = span\{k(x_1, \cdot), \ldots, k(x_n, \cdot)\}$. Then for any $f \in \mathcal{H}$ we have

$$J(Proj_{\mathcal{L}} f) \leq J(f).$$

Thus if $J^* = \min_{f \in \mathcal{L}} J(f)$ exists, then this minimum is attained for some $f \in \mathcal{L}$. Furthermore, if Ω is strictly increasing, then each minimizer of $J(f)$ over $\mathcal{H}$ is also contained in $\mathcal{L}$.

Proof. Let $f_{\|} = \operatorname{Proj}_{\mathcal{L}} f$. By Lemma A.2.2, $f(x_i) = f_{\|}(x_i)$ for $i = 1, \ldots, n$. Thus

$$V\left(f(x_1), \ldots, f(x_n)\right) = V\left(f_{\|}(x_1), \ldots, f_{\|}(x_n)\right).$$

Since f and $f - f_{\|}$ are orthogonal, we have

$$\|f\|_{\mathcal{H}}^2 = \|f_{\|}\|_{\mathcal{H}}^2 + \|f - f_{\|}\|_{\mathcal{H}}^2 \geq \|f_{\|}\|_{\mathcal{H}}^2.$$

Since Ω is nondecreasing, $\Omega\left(\|f_{\|}\|_{\mathcal{H}}^2\right) \leq \Omega\left(\|f\|_{\mathcal{H}}^2\right)$. Combining these results, we get $J(f_{\|}) \leq J(f)$. If Ω is strictly increasing and $f \neq f_{\|}$ (i.e. $\|f - f_{\|}\|_{\mathcal{H}}^2 > 0$), then

$$\begin{aligned} \|f_{\|}\|_{\mathcal{H}}^2 &< \|f\|_{\mathcal{H}}^2 \\ \Longrightarrow \Omega\left(\|f_{\|}\|_{\mathcal{H}}^2\right) &< \Omega\left(\|f\|_{\mathcal{H}}^2\right) \\ \Longrightarrow J(f_{\|}) &< J(f). \end{aligned}$$

Thus every minimizer of $J(f)$ must be contained in $\mathcal{L}$. □

We define the *kernel matrix* for the data $x_1, \ldots, x_n \in \mathcal{X}$ by the matrix $K = (k(x_i, x_j))_{i,j=1}^n$. This matrix is very useful when dealing with functions that live in the span of the data. The following proposition gives two useful expressions involving the kernel matrix:

Proposition A.3.2. *For any function $f : \mathcal{X} \to \mathbf{R}$ of the form*

$$f(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot),$$

for some $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbf{R}^n$, we have

$$\|f\|_{\mathcal{H}}^2 = \left\langle \sum_{i=1}^n \alpha_i k(x_i, \cdot), \sum_{i=1}^n \alpha_i k(x_i, \cdot) \right\rangle_{\mathcal{H}} = \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j) = \boldsymbol{\alpha}^T K \boldsymbol{\alpha}.$$

Let $\boldsymbol{f} = (f(x_1), \ldots, f(x_n))^T$ be the column vector of evaluations of f on the data points. Then we also have

$$\boldsymbol{f} = (f(x_j))_{j=1}^n = \left(\sum_{i=1}^n \alpha_i k(x_i, x_j) \right)_{j=1}^n = K\boldsymbol{\alpha}$$

A.4 Kernel Ridge Regression and the SVM

In this section we present two specific algorithms: kernel ridge regression, also known as regularized least squares (RLS), and the soft-margin support vector machine (SVM). In their most natural forms, the optimization problems associated with these algorithms are optimizations over an RKHS. Below, we show how to use the

Representer Theorem to reduce the optimization over a function space to an optimization over a finite dimensional Euclidean space. The techniques used here for RLS and the SVM can also be used for the CoRLS problem of Chapter 2, the multi-view point cloud regularization problem of Chapter 3, and the MR and CoMR problems of Chapter 4.

Consider an RKHS $\mathcal{H}$ and a loss functional $L : \mathcal{H} \to \mathbf{R}$. We would like to find

$$\operatorname*{arg\,min}_{f \in \mathcal{H}} \left[L(f) + \lambda \|f\|_{\mathcal{H}}^2 \right], \tag{A.1}$$

for some $\lambda > 0$. Suppose, as is typical in practice, the loss functional takes the form

$$L(f) = \sum_{i=1}^{n} V(f(x_i), y_i),$$

where $(x_1, y_1), \ldots, (x_n, y_n)$ are labeled training examples. Then by the Representer Theorem (Theorem A.3.1), if a minimizer for Equation (A.1) exists, then the minimum is attained by some function of the form $f_{\boldsymbol{\alpha}} = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbf{R}^n$. Thus we have reduced the problem of finding the optimal $f \in \mathcal{H}$ to the finite dimensional optimization problem of finding the best $\boldsymbol{\alpha} \in \mathbf{R}^n$.

We now recall some of the notation and facts derived above that will facilitate writing this finite-dimensional optimization problem in a form that can be easily solved on a computer. For any function $f_{\boldsymbol{\alpha}}$, as defined above, define the column vector $\boldsymbol{f}_{\boldsymbol{\alpha}} = (f_{\boldsymbol{\alpha}}(x_1), \ldots, f_{\boldsymbol{\alpha}}(x_n))^T$. Let $K = (k(x_i, x_j))_{i,j=1}^{n}$ be the $n \times n$ kernel matrix on the data points. Then by Prop. A.3.2, $\boldsymbol{f}_{\boldsymbol{\alpha}} = K\boldsymbol{\alpha}$ and $\|f_{\boldsymbol{\alpha}}\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^T K \boldsymbol{\alpha}$.

For RLS, we take the loss function is to be $V(\hat{y}, y) = (y - \hat{y})^2$. Let $\boldsymbol{y} = (y_1, \ldots, y_n)^T$. Then we can write the minimization as

$$\begin{aligned}
\operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathbf{R}^n} \left[L(f_{\boldsymbol{\alpha}}) + \lambda \|f_{\boldsymbol{\alpha}}\|_{\mathcal{H}}^2 \right] &= \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathbf{R}^n} \left(\boldsymbol{y} - K\boldsymbol{\alpha} \right)^T \left(\boldsymbol{y} - K\boldsymbol{\alpha} \right) + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\
&= \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathbf{R}^n} \boldsymbol{\alpha} K^2 \boldsymbol{\alpha} - 2\boldsymbol{y}^T K \boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\
&= \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathbf{R}^n} \left[\boldsymbol{\alpha} \left(K^2 + \lambda K \right) \boldsymbol{\alpha} - 2\boldsymbol{y}^T K \boldsymbol{\alpha} \right]
\end{aligned}$$

Since we are minimizing over an open set, the minimum must occur at a critical point, so we solve the first order conditions:

$$\begin{aligned} 0 = \partial_{\boldsymbol{\alpha}} \boldsymbol{\alpha} \left(K^2 + \lambda K\right) \boldsymbol{\alpha} - 2\boldsymbol{y}^T K \boldsymbol{\alpha} &= 2\left(K^2 + \lambda K\right) \boldsymbol{\alpha} - 2K\boldsymbol{y} \\ \Longrightarrow \left(K^2 + \lambda K\right) \boldsymbol{\alpha} &= K\boldsymbol{y} \end{aligned}$$

It is clear that one solution to this equality is $\boldsymbol{\alpha}_* = \left(K + \lambda I\right)^{-1} \boldsymbol{y}$, which exists if we assume $\lambda > 0$. Thus the RLS prediction function is $f_{\boldsymbol{\alpha}_*}$.

For the SVM, we take the loss function to be the *hinge loss*, which is defined as

$$V(y, \hat{y}) = (1 - y\hat{y})_+ = \begin{cases} 1 - y\hat{y} & \text{for } 1 - y\hat{y} \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The SVM optimization problem can be written as

$$\min_{\boldsymbol{\alpha} \in \mathbf{R}^n} \sum_{i=1}^{n} \left(1 - y_i f_{\boldsymbol{\alpha}}(x_i)\right)_+ + \lambda \|f_{\boldsymbol{\alpha}}\|_{\mathcal{H}}^2.$$

Defining $Y = \operatorname{diag}(y_1, \ldots, y_n)$, we can rewrite this optimization problem as

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbf{R}^n} \quad & \sum_{i=1}^{n} \boldsymbol{\beta}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\beta} \succeq 0 \\ & \boldsymbol{\beta} \succeq 1 - YK\boldsymbol{\alpha}, \end{aligned}$$

where $\succeq$ means that the inequality holds component-wise for the vectors. This optimization problem is a quadratic program with linear constraints, which can be solved by many standard numerical computing packages.

Appendix B

Rademacher Complexity

B.1 Definition

Definition B.1.1. *The empirical Rademacher complexity of a function class* $\mathcal{A} \subset \mathbf{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbf{R}\}$ *on a sample* $x_1, \ldots, x_n \in \mathcal{X}$ *is defined as*

$$\hat{R}_n(\mathcal{A}) = \mathbb{E}^{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{A}} \left| \frac{2}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right],$$

where the expectation is with respect to $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$*, and the* σ_i *are i.i.d. Rademacher random variables, that is,* $P(\sigma_i = 1) = P(\sigma_i = -1) = \frac{1}{2}$*.*

B.2 Rademacher Complexity of an RKHS Ball

In this section we state and prove a well-known theorem about the empirical Rademacher complexity of a ball in an RKHS. We make use of the following lemma, known as the Kahane-Khintchine inequality. The upper bound of the inequality is just Jensen's inequality, and a proof of the lower bound can be found in [19].

Lemma B.2.1 (Kahane-Khintchine Inequality). *For any vectors* $a_1, \ldots, a_n$ *in a Hilbert space and Rademacher random variables* $\sigma_1, \ldots, \sigma_n$ *(i.e.* σ_i *are i.i.d. with*

$P(\sigma_i = 1) = P(\sigma_i = -1) = 1/2$), we have

$$\frac{1}{2}\mathbb{E}\left\|\sum_{i=1}^{n}\sigma_i a_i\right\|^2 \leq \left(\mathbb{E}\left\|\sum_{i=1}^{n}\sigma_i a_i\right\|\right)^2 \leq \mathbb{E}\left\|\sum_{i=1}^{n}\sigma_i a_i\right\|^2$$

Denote the ball about the origin in the RKHS $\mathcal{H}$ of radius r by $\mathcal{H}_r := \{f \in \mathcal{H} : \|f\| \leq r\}$. Then we have the following well-known result:

Lemma B.2.2.

$$\frac{1}{\sqrt{2}}\frac{2r}{n}\sqrt{\operatorname{tr} K} \leq \hat{R}_n(\mathcal{F}_r) \leq \frac{2r}{n}\sqrt{\operatorname{tr} K}$$

where K is the kernel matrix for the sample points $x_1, \ldots, x_n \in \mathcal{X}$.

Proof.

$$\begin{aligned}
\hat{R}_n(\mathcal{F}_r) &= \mathbb{E}^\sigma\left[\sup_{f\in\mathcal{F}_r}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right] \\
&= \mathbb{E}^\sigma\left[\sup_{f\in\mathcal{F}_r}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i \langle f, k(x_i,\cdot)\rangle\right|\right] \\
&= \frac{2}{n}\mathbb{E}^\sigma\left[\sup_{f\in\mathcal{F}_r}\left|\left\langle f, \sum_{i=1}^{n}\sigma_i k(x_i,\cdot)\right\rangle\right|\right] \\
&= \frac{2r}{n}\mathbb{E}^\sigma\left\|\sum_{i=1}^{n}\sigma_i k(x_i,\cdot)\right\|.
\end{aligned}$$

The last equality is attained by noting that the supremum is maximized (and non-negative) when f is in the same direction as $\sum_{i=1}^n \sigma_i k(x_i,\cdot)$ and of maximum length. That is, for $f^* = r\sum\sigma_i k(x_i,\cdot)/\|\sum\sigma_i k(x_i,\cdot)\|$. Thus the supremum evaluates to $r\|\sum\sigma_i k(x_i,\cdot)\|$. Applying the Kahane-Khintchine inequality (Lemma B.2.1), we get

$$\frac{1}{\sqrt{2}}b \leq \mathbb{E}^\sigma\left\|\sum_{i=1}^n \sigma_i k(x_i,\cdot)\right\| \leq b,$$

where $b := \sqrt{\mathbb{E}^\sigma\|\sum_{i=1}^n \sigma_i k(x_i,\cdot)\|^2}$. By the reproducing property of the kernel $k(\cdot,\cdot)$, and the fact that $\mathbb{E}\sigma_i\sigma_j = \delta_{ij}$, we have

$$\begin{aligned}
b^2 = \mathbb{E}^\sigma\left\|\sum_{i=1}^{n}\sigma_i k(x_i,\cdot)\right\|^2 &= \mathbb{E}^\sigma\left(\sum_{i,j=1}^{n}\sigma_i\sigma_j k(x_i,x_j)\right) \\
&= \sum_{i=1}^{n}k(x_i,x_i) = \operatorname{tr}(K)
\end{aligned}$$

□