Topics in Machine Learning (CS729)

Instructor: Saketh

Contents

Chapter 1

Introduction

This is a specialized course on machine learning that focuses on statistical learning theory and kernel methods. The syllabus is as follows[1]:

I. Background Introduction to

- Statistical Learning Theory and Support Vector Machines (25%)
- Kernel Methods (15%)

II. Advanced Topics Learning theory, Formalization and Algorithms for:

- Semi-supervised Learning (25%)
- Learning with Structured-Data (20%)
- Handling Dataset-shift (15%)

We will begin by introducing the theory which answers the fundamental question "can we build systems that predict future well". The setting of "Supervised Inductive Learning" (SIL) is considered first (chapter 2). Section 2.1 presents the learning theory for this case and will enable us to formalize the learning problem (in this setting) as an optimization problem. We then study how the well-known Support Vector Machines implement this formalization in section 2.2. will be updated as and when required

[1] Numbers in brackets *roughly* indicate the number of lectures spent on the corresponding topic

Chapter 2

Supervised Inductive Learning

Humans are amazingly good at many cognitive tasks. For instance they recognize people from a distance and perhaps even when they are in odd postures. The question then comes whether we can build systems that perform similar cognitive tasks. However very less is known regarding how this cognition happens in humans.

Motivated by the process by which humans tend to learn, for instance to recognize people, we consider the simplest learning setting called the Supervised Inductive Learning (SIL). Here a *training set* consisting of input-output (x, y) pairs are assumed to be available. Training dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Each pair (x_i, y_i) is called a training instance; while x_i is called the training example/training data-point and y_i denotes its label. For eg., the input x could be a picture and the output could be whether it contains a human or not. The task in this example is to build a model which can predict whether ***any*** picture shown contains a human or not. Such a system perhaps could be used to improve google's image search. In general, given $\mathcal{D}$, the goal in SIL is to build a function f such that $f(x) = y$ for any new data-point x.

The special case where y takes only two distinct values, such as the example given above, is known as the setting of Binary Classification. Case where y takes on a set of finite values, for example we need to predict whether the given image is of a place in India or US or Japan etc., is known as Multi-class Classification. Multi-label Classification is the case similar to multi-class classification but data-points are allowed to be labeled with multiple values from a finite set, for eg. predict whether a image contains humans and/or animals and/or trees etc. In Ordinal Regression, y takes on finite number of numeric values (which makes labels comparable); for eg. one needs to predict whether a picture is highly-relevant or moderately-relevant or neutral or irrelevant to a particular topic/subject like say,

politics. The case of Regression is with y taking on real values, for eg. indicating the degree of relevance of the picture to politics. As one can see there are many real-world applications in which an SIL system is desirable.

Statistical Learning Theory (SLT) is the theory which focuses on the question whether such learning systems can be built. If so, what are the kind of guarantees we have on their performance etc. We introduce this theory in the SIL setting in the subsequent section.

2.1 Statistical Learning Theory (for SIL case)

Here we assume that the unknown concept modeling the input-output relation is some joint distribution $F_{XY}(x, y)$, where $X \in \mathcal{X}, Y \in \mathcal{Y}$ are the random variables denoting the input and output respectively. To simplify notation we use $P(x, y)$ for $F_{XY}(x, y)$. We further assume that the training dataset is a set of m iid samples from $P(x, y)$.

The ideal goal is to construct a function f such that the prediction error is low. One way of saying this is: "find an f from a function-class $\mathcal{F}$ such that $\mathbb{E}[1_{f(X)\neq Y}]$ is least", where $1_{f(X)\neq Y} = \begin{cases} 1 & \text{if } f(X) \neq Y, \\ 0 & \text{otherwise} \end{cases}$. In other words $f = \text{argmin}_{f\in\mathcal{F}}\mathbb{E}[1_{f(X)\neq Y}] = \text{argmin}_{f\in\mathcal{F}}P[f(X) \neq Y]$.

Its not necessary that we always penalize an f for mislabeling and moreover equally penalize for all mislabelings. For example, in case of regression, one might want to penalize less for small deviations from the true label and more for large deviations. It is hence typical to urge the application to provide with a loss function: $l : \mathcal{X} \times \mathcal{Y} \times \mathcal{F} \mapsto \mathbb{R}^+$. Typical loss functions used are listed and discussed in section 3.1 in Schölkopf and Smola [2002]. The simplest loss-function, discussed above, $l(X, Y, f) = 1_{f(X)\neq Y}$ is called the zero-one loss.

Lets also take a quick look at the possible function classes $\mathcal{F}$. The most interesting and widely used (because of its simplicity) is the set of linear functions: $\mathcal{F}^l_W = \left\{ f \mid f(x) = w^\top x,\ \|w\| \leq W \right\}$. For regression problems and binary classification problems with loss other than 0-1, one uses this function class frequently. However if one wishes to employ the 0-1 loss in the binary classification case, then one usually considers the composition of the $\mathcal{F}^l$ class with sign function, leading to the class of linear discriminators: $\mathcal{F}^{ld} = \left\{ f \mid f(x) = \text{sign}(w^\top x) \right\}$. One can easily think about counterparts of these classes for the affine, quadratic, cubic, etc. cases.

The expected loss with a function f is known as the risk with that f: $R[f] =$

$\mathbb{E}[l(X,Y,f)]$. R is called the risk functional which takes a f and outputs a number indicating the risk in employing the function as the predictor. With this notation, the ideal goal is to solve:

$$f^* = \text{argmin}_{f \in \mathcal{F}} R[f]. \tag{2.1}$$

Obviously this goal is note achievable as $R[f]$ is unknown because $P(x,y)$ is unknown[1]. Learning theory helps us realize what kind of goals can be reached starting from $\mathcal{D}, \mathcal{F}$ and also helps to formalize the learning problem with the (perhaps) relaxed goal.

We realized that a (random) quantity computable from $\mathcal{D}$, which is the average loss over the training set — denoted by $\hat{R}_m[f] = \frac{1}{m}\sum_{i=1}^{m} l(X_i, Y_i, f)$ and known in Machine Learning (ML) community as empirical risk of f, has an interesting property: the sequence of random variables $\hat{R}_1[f], \hat{R}_2[f], \ldots, \hat{R}_m[f], \ldots$ obtained by including a new sample from $P(x,y)$ into the training set at each stage and computing the average loss converges in probability to the (true) risk. i.e., $\left\{\hat{R}_m[f]\right\} \xrightarrow{p} R[f]$. This is from (weak) Law of Large Numbers (LLN) in probability theory (refer lectures 22-24 in Nath [2009]). This motivates the first induction principle:

Empirical Risk Minimization (ERM) [Vapnik, 1998]: Solve

$$f_m^{ERM} = \text{argmin}_{f \in \mathcal{F}} \hat{R}_m[f]. \tag{2.2}$$

Note that unlike (2.1), solving this problem may not be impossible. Though this makes ERM attractive, it is still a question how far will the true risk with f_m^{ERM} be from that with f^*. Given the results like LLN from probability theory we will be happy if: $\left\{R[f_m^{ERM}]\right\} \xrightarrow{p} R[f^*]$. If this convergence happens then we say ERM is consistent. Note that with such goals we are relaxing our initial goal (2.1) and saying that we are happy as long as we are Probably Approximately Correct (PAC) i.e., for finite m with high probability the risk with ERM candidate is close to risk with true candidate (in other words, ERM candidate is approximate). Now either when cardinality of $\mathcal{F}$ denoted by $|\mathcal{F}|$ is unity or when $\mathcal{F}$ includes a f which incurs zero loss on every sample of $P(x,y)$, then it is easy to see that ERM is consistent.

We gave an example where ERM is not (non-trivially) consistent: consider the case of binary classification with $\mathcal{F}$ containing all possible functions. Suppose we construct a f which simply remembers all training instances correctly (i.e., $f(x_i) = y_i$) and then outputs 1 (indicating positive class, say) for all other unseen data-points. Clearly the empirical risk with f is zero and the ERM picks it. With

[1] Note that $\mathbb{E}[l(X,Y,f)] = \int l(x,y,f)dP(x,y)$. And it is not possible to recover the mean from finite number of samples.

whatever m this is true; while the true risk could be arbitrary[2]. We then began the exploration "when is ERM consistent?". We realized that the condition for consistency is rather hard to verify because it involves true risk R (and not the $\hat{R}$). Hence we thought of writing down a sufficiency condition (which was proved to be a necessary condition for non-trivial consistency by Vapnik and Chervonenkis [1991]) for ERM consistency:

$$\lim_{m\to\infty} P\left[\max_{f\in\mathcal{F}}\left(R[f]-\hat{R}_m[f]\right)>\epsilon\right]=0,\ \forall\ \epsilon>0. \tag{2.3}$$

Refer sec. 5.4 in Schölkopf and Smola [2002] for the derivation of these conditions.
In some sense this says that the ERM is (non-trivially) consistent iff the deviation in the true and empirical risks in the worst-case f goes to zero. We will refer to this condition as the uniform convergence condition for ERM consistency[3]. In the subsequent section we analyze the case of finite function classes for ERM consistency.

2.1.1 ERM Consistency — Finite $\mathcal{F}$ case

Lets assume $\mathcal{F}$ has finite no. functions. Using Boole's inequality we have: $P\left[\max_{f\in\mathcal{F}}\left(R[f]-\hat{R}_m[f]\right)>\right.$ $\sum_{f\in\mathcal{F}} P\left[R[f]-\hat{R}_m[f]>\epsilon\right]$. Now we require to bound probabilities involving deviations of average of iid random variables from its mean. Chernoff bounding technique [Chernoff, 1952], is a general technique which provides a bound for probability of a linear function of independent random variables deviating from its true mean. The key steps in this technique are[4]:

- $P\left[R[f]-\hat{R}_m[f]>\epsilon\right]=P\left[e^{s\left(R[f]-\hat{R}_m[f]\right)}>e^{s\epsilon}\right]$ for some $s>0$.
- Applying Markov inequality gives LHS$\leq e^{-s\epsilon}\mathbb{E}[e^{s\left(R[f]-\hat{R}_m[f]\right)}]$

[2]Provided the space $\mathcal{X}$ is not finite.

[3]Because it resembles that of uniform convergence criteria in case of sequence of real-valued functions on $\mathbb{R}$. The difference being the present condition is "one-sided".

[4]Note that the technique is generic and when applied with different partial information about the involving random variables and the function combining them, one gets different bounds. We will shortly see another bound called McDiarmid's inequality which follows most of these basic steps. You can also refer sec.5.2 in Schölkopf and Smola [2002] for detailed derivation (for case $|cal F| = 1$). Here we provide the version with the relevant random variables for the present context.

- Use the fact that the random variables[5] $L_1(f), L_2(f), \ldots, L_m(f)$ are independent (infact iid): LHS$\leq e^{-s\epsilon}\Pi_{i=1}^{m} E[e^{\frac{s}{m}(\mathbb{E}[L_i(f)]-L_i(f))}]$
- Use the Hoeffding bound (refer http://en.wikipedia.org/wiki/Hoeffding%27s_lemma for proof) to bound the moment generating function (mgf) of the mean zero and finitely supported random variable $\mathbb{E}[L_i(f)] - L_i(f)$ (finite support is true whenever the loss function is bounded, which in particular is true with zero-one loss): LHS$\leq |\mathcal{F}|e^{-s\epsilon}e^{\frac{s^2}{8m}}$.
- Finally, choose the best s (by minimizing the bound on RHS): LHS$\leq |\mathcal{F}|e^{-2m\epsilon^2}$

This bounding first of all shows that the probability term in question which is sandwiched between zero and $|\mathcal{F}|e^{-2m\epsilon^2}$ goes to zero as $m \to \infty$ — confirming that ERM is consistent in finite $|\mathcal{F}|$ case[6]. In other words, PAC learning is possible with ERM in the finite $|\mathcal{F}|$ case. Secondly, re-writing the bound by denoting $\delta = |\mathcal{F}|e^{-2m\epsilon^2}$ gives:

with probability $1-\delta$,

$$R[f] \leq \hat{R}_m[f] + \sqrt{\frac{1}{2m}\log\left(\frac{|\mathcal{F}|}{\delta}\right)} \;\forall\; f \in \mathcal{F}. \tag{2.4}$$

Inequalities of such type are called as VC-type inequalities[7]. Interestingly this gives an upper-bound on the risk (the quantity we want to minimize) that involves terms that can be computed based on $\mathcal{D}$ and $\mathcal{F}$. Hence such bounds provide computable (upper) bounds on the performance (risk) of f obtained with an induction principle like ERM[8]. Moreover, such bounds motivate a new induction principle that suggests minimizing the bound itself:

Structural Risk Minimization (SRM) [Vapnik, 1998]: Given a $\mathcal{F}$ construct the sets $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}$. This is like giving structure to $\mathcal{F}$, based on increasing size/complexity/richness[9]. Solve: $i^* = \text{argmin}_i \min_{f\in\mathcal{F}_i} \hat{R}_m[f] + \sqrt{\frac{1}{2m}\log\left(\frac{|\mathcal{F}_i|}{\delta}\right)}$. The candidate for SRM is $f_m^{SRM} = \text{argmin}_{f\in\mathcal{F}_{i^*}} \hat{R}_m[f]$.

[5] We denote the random variable (X_i, Y_i) by Z_i and the random variable $l(X_i, Y_i, f) = l(Z_i, f)$ by $L_i(f)$.

[6] Note that the analysis is very similar in the countable case. It is the uncountable case which calls for a different analysis. Nevertheless at a later stage we will clarify why countable case is similar to the finite case.

[7] As they were popularized by Vapnik and Chervonenkis.

[8] We commented on the play between $|\mathcal{F}|, m, \delta$ and the tightness of the bound.

[9] Application specific domain knowledge can perhaps motivate preferring a particular structure over the others.

The story seems to good in the finite/countable $\mathcal{F}$ case. However for real-world applications, such function classes are rather useless. Hence we turned our attention to the case of arbitrary (possibly uncountable) function classes. Refer theorem 5 in Bousquet et al. [2004] for the details of the derivation in this case[10]. In the following section we provided a rough sketch of the same.

2.1.2 ERM Consistency — General $\mathcal{F}$ case

In arbitrary function class case one cannot resort to the Boole's inequality and one needs to focus on the random variable $g(Z_1, \ldots, Z_m) = \max_{f \in \mathcal{F}} R[f] - \hat{R}_m[f]$. We noted that g is a function of iid random variables and moreover satisfies the bounded difference property. Hence one can employ the McDiarmid's inequality [McDiarmid, 1989] to bound probability of high deviations of g from its mean. Refer www.cs.berkeley.edu/~bartlett/courses/281b-sp06/bdddiff.pdf for an easy proof of the McDiarmid inequality and the definition of bounded difference property. With this we have that with probability $1 - \delta$,

$$R[f] \leq \hat{R}_m[f] + \mathbb{E}\left[\max_{f \in \mathcal{F}} R[f] - \hat{R}_m[f]\right] + \sqrt{\frac{1}{2m} \log\left(\frac{1}{\delta}\right)}, \ \forall \ f \in \mathcal{F} \tag{2.5}$$

The equation holds for losses which vary between 0 and 1 (like 0-1 loss or truncated hinge-loss). Needless to say, a similar statement can be written for any bounded loss function.

We noted that the expectation in the RHS above represents how big a function class is and hence the VC-type inequality in the general $\mathcal{F}$ case is very similar to that in the finite case (2.4). In order that the bound is useful we wanted to further bound the expectation term (which is unknown):

Ghost Samples: $\mathbb{E}\left[\max_{f \in \mathcal{F}} R[f] - \hat{R}_m[f]\right] = \mathbb{E}\left[\max_{f \in \mathcal{F}} \mathbb{E}\left[\hat{R}'_m[f]\right] - \hat{R}_m[f]\right]$. Here $\hat{R}'_m[f] = \frac{1}{m}\sum_{i=1}^{m} l(Z'_i, f)$ represents the empirical risk with f evaluated on a set of m iid samples $Z'_1, \ldots, Z'_m$ (called ghost samples) which are independent of the given training set.

Max. and Expectation interchange: Since maximum of sum/integral is less than or equal to sum/integral of maxima, we have[11]: $\mathbb{E}\left[\max_{f \in \mathcal{F}} \mathbb{E}\left[\hat{R}'_m[f]\right] - \hat{R}_m[f]\right] \leq \mathbb{E}\left[\max_{f \in \mathcal{F}} \hat{R}'_m[f] - \hat{R}_m[f]\right] = \mathbb{E}\left[\max_{f \in \mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\left(l(Z'_i, f) - l(Z_i, f)\right)\right]$. Note that the final expectation is wrt. both Z_i and Z'_i forall i.

[10] Refer Koltchinskii [2001] for the original paper.

[11] This explanation is perhaps more apt than the contrived Jensen's inequality argument presented in lecture.

Rademacher variables: With motivation from studies of empirical processes [Ledoux and Talagrand, 1991] and the fact that we want to elevate the difficulty in computing the expectation (which is unknown as distribution P itself is unknown) by using ideas of conditioning on expectation, we introduce new random variables $\sigma_1, \ldots, \sigma_m$, called Rademacher variables, which are iid with distribution: $P[\sigma_i = 1] = 0.5, P[\sigma_i = -1] = 0.5$. We have, $\mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\left(l(Z_i', f) - l(Z_i, f)\right)\right] = \mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i\left(l(Z_i', f) - l(Z_i, f)\right)\right]$. This equality is true because the distribution of $l(Z_i', f) - l(Z_i, f)$ is symmetrical. Note that the expectation in the last expression is wrt. all random variables i.e., $Z_i, Z_i', \sigma_i, \ \forall\ i$.

Again, max. and sum inequality: $\mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i\left(l(Z_i', f) - l(Z_i, f)\right)\right] = \mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i l(Z_i', f)\right] + \mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} -\sigma_i l(Z_i, f)\right] = 2\mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i l(Z_i, f)\right]$ This expectation has a name: Rademacher average of a function class $\mathcal{G}$ is defined as $\mathcal{R}(\mathcal{G}) = \mathbb{E}\left[\max_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(Z_i)\right]$, where the expectation is over the random variables $Z_i, \sigma_i, \ \forall\ i$. With this notation the expectation in the final expression above can be called as Rademacher average[12] of the class $\mathcal{L} = l \circ \mathcal{F} = \{l(\cdot, \cdot, f) \mid f \in \mathcal{F}\}$. The Rademacher average conditioned on the training examples is called the conditional Rademacher average: $\hat{\mathcal{R}}(\mathcal{G}) = \mathbb{E}\left[\max_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(Z_i) \mid Z_1, \ldots, Z_m\right]$. Note that unlike $\mathcal{R}$, the quantity $\hat{\mathcal{R}}$ can be computed (given the training set). Hence we would like to have a bound in terms of $\hat{\mathcal{R}}$ rather than $\mathcal{R}$.

McDiarmid Inequality: It is easy to see that the function $h(Z_1, \ldots, Z_m) = \hat{\mathcal{R}}(\mathcal{L})$ satisfies bounded difference property and hence application of McDiarmid's inequality[13] gives with probability $1 - \delta$:

$$\mathcal{R}(\mathcal{L}) = \mathbb{E}\left[\hat{\mathcal{R}}(\mathcal{G})\right] \leq \hat{\mathcal{R}}(\mathcal{L}) + \sqrt{\frac{1}{2m}\log\left(\frac{1}{\delta}\right)} \tag{2.6}$$

Union bound: Combining equations (2.5) and (2.6) with a union bound (Boole's inequality) we have with probability $1 - \delta$:

$$R[f] \leq \hat{R}_m[f] + 2\hat{\mathcal{R}}(\mathcal{L}) + 3\sqrt{\frac{1}{2m}\log\left(\frac{2}{\delta}\right)},\ \forall\ f \in \mathcal{F} \tag{2.7}$$

[12] In lecture we gave intuition of why Rademacher average measures complexity of a function class.

[13] Again, the inequality is written with 0-1 loss of truncated hinge-loss in mind. Similar expression for any bounded loss can be written.

Now one sufficiency condition for ERM being consistent is ofcourse $\hat{\mathcal{R}}(\mathcal{L}) \to 0$ as $m \to \infty$. This is evident from (2.7) by re-writing it as upper bound on probability of the complementary event. Clearly this does not happen with $\mathcal{F}$ being the set of all (measurable) functions as in that case $\hat{\mathcal{R}} = 0.5$ (assuming 0-1 loss). This establishes the statement that PAC learning may not be possible unless the function class is restricted in its complexity (as measured by Rademacher averages). In the subsequent section we look at linear-discriminant function class $\left\{ f \mid f(x) = \text{sign}(w^\top x) \right\}$, which is shown to be "good" for text categorization tasks, and look at what restrictions lead to ERM consistency.

2.1.3 Example of function/loss class with ERM consistency — Linear functions

We took the case of binary classification. We noted that the 0-1 loss is not attractive for two reasons: i) in binary classification problems one may want a hold on confidence of the label prediction. Hence one may want to use hinge-loss of its variants (which basically says more the value of $w^\top x$, more the confidence that x belongs to the positive class and vice-versa). ii) the ERM problem with 0-1 loss itself is computationally hard (a hard combinatorial optimization problem)[14].

The following discussion hence assumes truncated hinge-loss with which also (2.7) holds. We focus on the class of linear functions $\mathcal{F}_W^l$ in n-dimensional Euclidean space[15]. Notation: let $l(x, y, f) = \phi(y f(x))$, where $\phi(z) = \min(\max(0, 1 - z), 1)$ (representing the truncated hinge loss). We came up with an upper bound on the conditional Rademacher average in this case[16] (we assume things as and when necessary):

Contraction Lemma:

$$\hat{\mathcal{R}}(\mathcal{L}) = \mathbb{E}\left[\max_{\|w\| \leq W} \frac{1}{m} \sum_{i=1}^{m} \sigma_i \phi\left(y_i w^\top x_i \right) \right] \leq \mathbb{E}\left[\max_{\|w\| \leq W} \frac{1}{m} \sum_{i=1}^{m} \sigma_i y_i w^\top x_i \right].$$

This follows from the contraction lemma [Ledoux and Talagrand, 1991] (refer Lemma5 in Meir and Zhang [2003] for a simple proof) as ϕ is a Lipschitz continuous function[17] with Lipschitz constant as unity.

[14] Infact a more comprehensive statement can be made: refer Feldman et al. [2009] for details.

[15] We noted that in real-world text categorization applications promising results were obtained using $\mathcal{F}_l$ and hinge-loss (for which the truncated hinge loss forms a lower bound) — making this example a non-trivial and infact interesting one.

[16] The derivation presented here is based on the proof of theorem 24 in Lanckriet et al. [2004]

[17] A function f is said to be Lipschitz continuous with Lipschitz constant L iff $|f(x) - f(y)| \leq L\|x - y\| \ \forall \ x, y \in dom(f)$.

Cauchy-Schwartz Inequality:

$$\mathbb{E}\left[\max_{\|w\|\leq W}\frac{1}{m}\sum_{i=1}^{m}\sigma_i y_i w^\top x_i\right] \leq \frac{W}{m}\mathbb{E}\left[\|\sum_{i=1}^{m}\sigma_i y_i x_i\|\right] = \frac{W}{m}\mathbb{E}\left[\sqrt{\hat{\sigma}^\top K\hat{\sigma}}\right],$$

where $\hat{\sigma}$ is the vector with entries as $\sigma_i y_i$ and K is the matrix of all possible dot products: $(i,j)^{th}$ entry in K is $K_{ij} = x_i^\top x_j$. Such a matrix is called a gram matrix. So K is the gram matrix of the training datapoints.

Jensen's Inequality: $\frac{W}{m}\mathbb{E}\left[\sqrt{\hat{\sigma}^\top K\hat{\sigma}}\right] \leq \frac{W}{m}\sqrt{\mathbb{E}\left[\hat{\sigma}^\top K\hat{\sigma}\right]}$ and this is equal to $\frac{W}{m}\sqrt{trace(K)}$, as σ_i are iid with mean zero and variance unity[18].

Radius bound: Now one can easily come up with cases where the above bound may not go to zero (for $m \to \infty$) as the trace term in the numerator may itself blow. One way of restricting this is to say that the input space $\mathcal{X}$ is bounded i.e., there exists an r such that $\|x\| \leq r \ \forall \ x \in \mathcal{X}$. With this assumption one obtains the following radius-margin bound[19]:

$$\hat{\mathcal{R}}(\mathcal{L}) \leq \frac{Wr}{\sqrt{m}}, \tag{2.8}$$

which indeed goes to zero as $m \to \infty$.

Hence ERM should be consistent in this case. Using similar learning theory bounds Vapnik [Vapnik, 1998] proposed a optimization formalism that implements the ERM principle. This is the well celebrated formulation of SVMs (Support Vector Machines), which is the subject of discussion in the subsequent section.

2.2 Support Vector Machines (SVMs)

Motivated by the result that ERM is consistent, one can look for a linear function which solves the following problem:

$$\begin{aligned} \min_{w\in\mathbb{R}^n} \quad & \textstyle\sum_{i=1}^{m} l(x_i, y_i, w), \\ \text{s.t.} \quad & \|w\| \leq W \end{aligned} \tag{2.9}$$

One may use the truncated hinge loss or any upper bound of it. For eg. hinge loss. The advantage with hinge-loss is it is convex[20], whereas the truncated hinge-loss

[18] Trace of matrix M is sum of its diagonal entries

[19] We noted in the lecture why the bound is intuitive in the binary classification case.

[20] One may also re-derive the bounds for hinge-loss case, which would lead to similar expressions and results.

is not. With hinge loss (2.9) can be written as:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \textstyle\sum_{i=1}^m \max(0, 1 - y_i w^\top x_i), \\ \text{s.t.} \quad & \|w\| \leq W \end{aligned} \tag{2.10}$$

The above problem is convex (and hence can be solved efficiently). Infact it can be posed as a Second-Order Cone Program (SOCP)[21], once the objective is turned linear: we used a standard trick of introducing additional variables ξ_i such that $\xi_i \geq \max(0, 1 - y_i w^\top x_i)$. This gives:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \textstyle\sum_{i=1}^m \xi_i, \\ \text{s.t.} \quad & \|w\| \leq W,\ \xi_i \geq 0,\ y_i w^\top x_i \geq 1 - \xi_i. \end{aligned} \tag{2.11}$$

Infact problems of the form (2.9) have been studied in optimization theory. Most common example is with the case of square-loss (regression problem). The term in the objective measures the fit of the model to the data, while the constraint "regularizes" the model. Such a regularization is known as Ivanov regularization. Moreover, regularization problems can be written in two more equivalent forms:

Tikhonov regularization:

$$\min_{w \in \mathbb{R}^n} \|w\| + C \sum_{i=1}^m l(x_i, y_i, w), \tag{2.12}$$

where C is a parameter (plays a role similar to W). Here the interpretation is fit the model to the data while regularizing it. C controls the trade-off between data fit and regularization. Some also refer to such a form as "Regularized risk minimization" (which we have shown is equivalent to ERM). Here regularized risk refers to the weighted sum of the regularizer and empirical risk.

Morozov regularization:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \|w\|, \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m l(x_i, y_i, w) \leq A, \end{aligned} \tag{2.13}$$

where A is a parameter similar to C and W. Here the interpretation maximally regularize the model while data fit is under certain tolerance. A is a bound on the (empirical) error of data fit.

The Tikhonov regularized version with hinge-loss was used by Cortes and Vapnik [1995] and published as SVMs (only difference being $0.5\|w\|^2$ is used instead of $\|w\|$ as the regularizer):

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \tfrac{1}{2}\|w\|^2 + C \textstyle\sum_{i=1}^m \xi_i, \\ \text{s.t.} \quad & \xi_i \geq 0,\ y_i w^\top x_i \geq 1 - \xi_i. \end{aligned} \tag{2.14}$$

[21] refer `http://stanford.edu/~boyd/papers/socp.html`.

The squared version of the regularizer was used to obtain a nice convex Quadratic Program (as above), for which highly efficient off-the-shelf solvers exist.

The Morozov regularized version (with squared-regularizer, hinge-loss and $A = 0$ i.e., no empirical error) was used in a preliminary paper before SVM [Boser et al., 1992] and leads to what usually is known as the hard-margin SVM:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \tfrac{1}{2}\|w\|^2, \\ \text{s.t.} \quad & y_i w^\top x_i \geq 1. \end{aligned} \tag{2.15}$$

Please read Burges [1998], which is an excellent tutorial on SVMs. Here we tried to cover things not covered there (including learning theory results). We next provide an insight into the specialty of the solution with the SVM problem that will be helpful in our analysis later on.

Note that the geometric interpretation of (2.15) is that of maximally separating two set of points. It is well known that this problem is equivalent to minimizing distance between convex hulls of the two sets of points[22]. Infact, the normal to the maximally separating hyperplane (i.e., w) will be in the direction of line joining the two minimum distant points in the convex hulls. From this it is immediate that $w = \sum_{i=1}^m \alpha_i x_i$. Infact, later on we will (rigorously) prove a more generic statement under the name "Representer theorem" — which says (loosely) any "SVM-kind" of problem (i.e., norm-regularized linear fit problem) has a solution of the form $w = \sum_{i=1}^m \alpha_i x_i$ i.e., the solution is a linear combination of the training datapoints. Moreover, the name "Support Vector" is also motivated from this duality result: from the above argument it is also clear that many αs can be zero at optimality and hence the solution is a linear combination of few important examples called "support vectors". Will fill-in more details as and when required.

With this discussion we are clear about ERM. Though ERM is consistent, the function class $\mathcal{F}$ itself may be too big (in which case we may overfit) or too small (in which case we may underfit). The problem of which $\mathcal{F}$ to choose is hence crucial and is discussed in the subsequent section.

2.3 Model Selection Problem

Here we deal with the question which $\mathcal{F}$ to choose? Ideally we want $\mathcal{F}$ to be as big a set as possible so that $R[f^*]$ is as close as possible to $R[f^{**}]$, where $f^{**} = \text{argmin}_f R[f]$ i.e., the minimizer of true risk among all (measurable) functions. f^{**}

[22] Infact, this equivalence drives all duality principles in optimization. Refer notes at `http://www.cse.iitb.ac.in/saketh/teaching/cs709.html` for details.

is called the Bayes (optimal) function[23]. The risk with f^{**} is called the Bayesian (optimal) risk. However we at a very early stage of our analysis realized that one may not be consistent if $\mathcal{F}$ is very big (say all functions).

So the obvious idea is to try several $\mathcal{F}_i$ and choose the "best". Now the problem of choosing the "best" $\mathcal{F}_i$ is called the model selection problem. Analogously, the problem of finding the "best" f_i given $\mathcal{F}_i$ may be called the model-parameter selection problem (hence ERM is a principle for model-parameter selection). On passing, we introduce some more terminology: given an induction principle (like ERM), let the candidate selected by it in a function class $\mathcal{F}$ be f_m^*. The difference between risks of $f_m^* \in \mathcal{F}$ and $f^* \in \mathcal{F}$ (which is the true minimizer of risk in $\mathcal{F}$) is called the Estimation error: $EstErr = R[f_m^*] - R[f^*]$. This indicates the error introduced in finding risk minimizer because of finite data and it usually decreases with m (atleast we know that in probability it goes to zero as $m \to \infty$ for f_m^* returned by ERM). The difference between the risks of $f^* \in \mathcal{F}$ and the Bayesian risk is called the approximation error: $AprErr = R[f^*] - R[f^{**}]$. This indicates the error in approximating the set of all functions with $\mathcal{F}$. The related quantity that measures difference in risks with the induced f_m^* and the Bayes function is called the generalization error: $GenErr = R[f_m^*] - R[f^{**}]$. Needlessly to say, generalization error is of atmost interest to us. One says that an induction principle is Bayes consistent iff $\{R[f_m^*]\} \xrightarrow{p} R[f^{**}]$. We still need to do quite a bit of analysis to answer questions about Bayes consistency. For the time being we will be happy with (statistical) consistency i.e., $\{R[f_m^*]\} \xrightarrow{p} R[f^*]$, which was our subject of discussion from the beginning.

What ever is the terminology, the important question is which $\mathcal{F}$ to choose? A hint towards this goal is given by (2.7) itself! For example, one may look for the $f_i \in \mathcal{F}_i$ which minimizes this bound. Then the hope is that the true risk is minimized by minimizing its upper bound. This ofcourse is the idea behind SRM discussed earlier:

One chooses a hierarchy of function classes: $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_n \subset \ldots$, each of which have decaying Rademacher average (i.e., ERM consistency is guaranteed), and then picks $i^* = \text{argmin}_i \min_{f \in \mathcal{F}_i} \tilde{R}[f]$, where $\tilde{R}[f]$ is called the guaranteed risk with f which is the vc-type bound on the true risk (one may use RHS of (2.4) or (2.7) as the case may be[24]). The candidate for SRM is $f_m^{SRM} = \text{argmin}_{f \in \mathcal{F}_{i^*}} \hat{R}_m[f]$.

---

[23] In case of binary classification, this optimal is given by $f^{**}(x) = \begin{cases} 1 & \text{if } P[Y=1/X=x] \geq P[Y=-1/X=x] \\ -1 & \text{if } P[Y=-1/X=x] > P[Y=1/X=x] \end{cases}$. Refer Duda et al. [2000] or any other classical pattern recognition/machine learning book for an in depth discussion. Note that the Bayes optimal function cannot be realized as $P(x,y)$ is unknown.

[24] Infact, researchers have come up with various bounds which sometimes involve notions about function-class complexity other than Rademacher averages. Please refer the following for de-

It is easy to see that such a principle, provided we prove its consistency, is indeed useful for model selection. Infact, a closer look convinces us that with such a principle we can perhaps get close to Bayes consistency. This is because SRM kind of searches in $\cup_{i=1}^{\infty}\mathcal{F}_i$, which itself need not be a class where ERM is consistent. For eg. one may choose $\mathcal{F}_1^l, \mathcal{F}_2^l, \ldots, \mathcal{F}_n^l, \ldots$ whose union is all possible linear functions. We will prove that SRM is (statistically) consistent in the subsequent section.

On passing, we note that there are alternative principles for model selection. The most frequently used is the validation-set method and its variants. Here one divides the given dataset into two parts: i) the training set ii) the validation set. Using the training set alone, $f_m^{*i} \in \mathcal{F}_i,\ i = 1, \ldots, k$ are constructed by implementing some induction principle (say, ERM). Now the problem of model selection is equivalent choosing among $\mathcal{F} = \left\{f_m^{*1}, f_m^{*2}, \ldots, f_m^{*k}\right\}$. While in case of SRM this choice is made by further looking at guaranteed risk, here one evaluates each f_m^i on the validation set and computes validation risk (which is same as empirical risk but evaluated with validation set samples rather than training set samples). Again since LLN gives that validation risk is a good (asymptotic) estimate of the true risk, we pick the f_m^{*i} which gives least validation error. While this is fine because we have a relation similar to (2.4), the bound also says one should not take too high k and then look for a validation risk minimizer because like with ERM, this might lead to over-fitting (to the validation set); while taking small k may lead to under-fitting (to the validation set). One may resort to something like SRM again to decide what k. Nevertheless in practice one just fixes a "reasonable" $k = 5$, say and looks for validation risk minimizer. This is called the validation-set method. Please refer Chapelle et al. [2002] for other variants.

2.3.1 SRM consistency

In this section we show that SRM is consistent in the specific case as that in section 2.1.3. Refer appendix-1 for the details and a proof[25] of SRM consistency that is based on the derivations in Lugosi and Zeger [1996].

We commented that this is a remarkable result as it gives us a way of being (statistically) consistent in potentially large function classes (i.e., $\cup_{i=1}^{\infty}\mathcal{F}_i$; whose Rademacher average may not decay with m) while performing a principled search (SRM) among function classes ($\mathcal{F}_i$) with restricted capacity. This will lead us to Bayes consistency provided we consider functions class ($\cup_{i=1}^{\infty}\mathcal{F}_i$) which can well approximate or contain the Bayes optimal function. Since the Bayes optimal function can be any "measurable" function and need not be linear, we first generalize

tails: Bousquet et al. [2004], Bartlett and Mendelson [2002], Vapnik [1998]

[25] All appendix sections appear towards the end of this notes.

our analysis to non-linear function classes. This analysis is presented in the next section (which is an abridged version of the explanation in section 2.1 in Schölkopf and Smola [2002]).

2.4 Non-linear Function-classes

Through examples of affine and quadratic functions, we noted that non-linear functions in input space $\mathcal{X}$ are nothing but linear functions in a suitable (non-linearly) transformed space $\phi(\mathcal{X})$. e.g. $f(x) = ax_1^2 + bx_2^2 + \sqrt{2}cx_1x_2 = [a\ b\ c]^\top \phi(x),\ \phi(x) = [x_1^2\ x_2^2\ \sqrt{2}x_1x_2]^\top$ (here $x = [x_1\ x_2]^\top \in \mathbb{R}^2$). We also noted this is the case with all polynomial functions. This observation motivates the following methodology for handling non-linear function classes: given a polynomial function class (say all polynomials upto degree d) we first create the space $\phi(\mathcal{X})$ that contains in each dimension a monomial involving the input dimensions. Then we consider linear function classes over this new feature space $\phi(\mathcal{X})$. And one can repeat the entire analysis in previous sections. The only constraint is ϕ should be such that $\|x\| \leq r \Rightarrow \|\phi(x)\| \leq r'$ for some r' and this holds for the polynomials case atleast.

For a moment we might think the problem is solved, but as Lokesh pointed out creation of the feature space might require astronomical time: if the input dimensionality is n and degree of polynomials under consideration is d, then the size of the feature vector is $n+d+1$ choose d. This number could be unmanageable with even reasonable n, d. So though our methodology is flawless theoretically, when it comes to implementation it looks like it may take a beating.

The obvious question is do we really need to compute $\phi(x)$? A re-look at the nature of SVM solution hinted towards the end of section 2.2 suggests that it is enough to know the dot-products of examples in order to solve the SVM (i.e., ERM) problem. This is because, using $w = \sum_{i=1}^m \alpha_i x_i$, (2.14) can be re-written as:

$$\begin{array}{ll} \min\limits_{\alpha \in \mathbb{R}^m} & \sum_{i=1}^m \max\left(0, 1 - y_i \sum_{j=1}^m \alpha_j x_j^\top x_i\right), \\ \text{s.t.} & \sqrt{\alpha K \alpha} \leq W, \end{array} \tag{2.16}$$

here K is the gram matrix with the training datapoints. Moreover, the evaluation of the SVM/ERM candidate function can be done using dot-products alone: $f(x) = \sum_{i=1}^m \alpha_i x_i^\top x$. This raises the question can we (atleast in some cases) efficiently compute the dot products in feature spaces using the input space vectors? If so, then we can solve the SVM in the feature space without explicitly going into the feature space.

We realized that this again can be done in the polynomial function class case as above: e.g. for homogeneous quadratic in $\mathbb{R}^2$ case $\phi(x)^\top \phi(z) = x_1^2 z_1^2 + x_2^2 z_2^2 +$

$2x_1x_2z_1z_2 = (x^\top z)^2$. Similarly, in case of non-homogeneous d degree polynomials we can compute the dot product in the feature space using $(1 + x^\top z)^d$.

So till now the story is excellent... we can handle polynomial function classes on Euclidean spaces using the analysis of linear function classes and computation-wise also there are no challenges. Now this makes us greedy and ask the question can we do this for non-linear functions over arbitrary input spaces $\mathcal{X}$ that are not Euclidean (such a situation arises for example in a task of classifying images/videos etc. — which are hard to describe using Euclidean vectors). Secondly, since our primary goal is Bayes consistency the key question is do we get large enough function classes with polynomials? Intuitively atleast the answer seems no as it is sounding too restrictive to say that Bayes optimal is a polynomial function. However what might be more believable is that perhaps $e^{x^\top z}$ (we write this function by looking at $(x^\top z)^d$) is the function which might represent a dot product in the feature space that have all monomials without any degree restriction. Even if this were true, ofcourse such a feature space wont be a Euclidean space rather a Hilbert space[26], which generalizes the notion of Euclidean spaces. In summary, we are looking at results in mathematics that kind of say which class of functions (we name them as positive kernels later) represent inner-products (generalization of dot product notion) in some Hilbert space? Infact such results are well-known, even at the beginning of the previous century, in the field of operator theory. In the subsequent section we will discuss such a key result that will help us solve both our problems (handling generic input spaces and feature maps which lead to "big" function classes such as with $e^{x^\top z}$) in one shot.

2.4.1 Kernels and Kernel-trick

With the motivation in the previous section we begin with the following definition: Given an input space $\mathcal{X}$ (need not be Euclidean; infact need not be a vector space), a positive kernel is any function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying i) symmetry: $x, z \in \mathcal{X} \Rightarrow k(x, z) = k(z, x)$ and ii) Positivity: $x_1, \ldots, x_m \in \mathcal{X} \Rightarrow G_k(x_1, \ldots, x_m) \succeq 0$, where $G_k(x_1, \ldots, x_m)$ is the matrix with ij^{th} entry as $k(x_i, x_j)$ i.e., it is the matrix of all possible kernel evaluations on the given set of m points. The symbol $M \succeq 0$ means that the matrix M is positive semi-definite (psd)[27].

[26] Refer lecture-notes 1-4 in Saketh [2009] for refreshing the idea of Hilbert spaces. We also noted two non-Euclidean Hilbert-spaces: space of square-summable sequences (l_2) http://en.wikipedia.org/wiki/Sequence_space and space of square integrable functions (L_2) http://en.wikipedia.org/wiki/Lp_space. Infact, all infinite-dimensional (separable) Hilbert spaces are "equivalent" to the l_2 space, which is an intuitive generalization of Euclidean space.

[27] $M \succeq 0 \Leftrightarrow x^\top Mx \geq 0 \ \forall \ x$. Some textbooks may prefer to define psd matrices as symmetric ones satisfying this condition — leading to a definition of positive kernels in Schölkopf and Smola

One can now prove the following crucial theorem [Schölkopf and Smola, 2002]:

Theorem 2.4.1. *Consider an input space $\mathcal{X}$ and a positive kernel k over it. Then there exists a Hilbert space $\mathcal{H}_k$ and a feature map $\phi_k : \mathcal{X} \to \mathcal{H}_k$ such that the kernel evaluation of any two datapoints in the input space, i.e., $k(x, z)$, is equal to the inner product of those two datapoints in the feature space, i.e., $\langle \phi_k(x), \phi_k(z) \rangle_{\mathcal{H}_k}$. In other words, $k(x, z) = \langle \phi_k(x), \phi_k(z) \rangle_{\mathcal{H}_k}$.*

Refer section 2.2.2 in Schölkopf and Smola [2002] for a proof of the same[28].

Note that this theorem shows existence of a Hilbert space. Obviously there may be several space and mappings satisfying this criteria. Refer to theorem 2.10 and proposition 2.12 in Schölkopf and Smola [2002] for an alternate Hilbert space, actually an l_2 space, construction.. However, from the proof it is clear that the theorem points out a special Hilbert space that satisfies the following condition: $f \in \mathcal{H}_k \Rightarrow f(x) = \langle f(\cdot), k(x, \cdot) \rangle_{\mathcal{H}_k}$. Note that this condition may not be satisfied by other Hilbert spaces that satisfy the criteria. This special Hilbert space pointed out in theorem 2.4.1 above is called a Reproducing Kernel Hilbert Space (RKHS).

Now all this development is useful, only if we show some examples of positive kernels. Before giving examples lets look at some operations that preserve positivity of kernels, which come in handy to prove positiveness of a given function. i) conic combination of positive kernels is positive ii) product of positive kernels is positive iii) limit of a sequence of positive kernels (if exists) is positive. Refer section 13.1 in Schölkopf and Smola [2002] for details. Though these results are simple to prove we argued that from application perspective they are far reaching: consider an application involving multi-modal data (say, video,audio, text modes) and suppose kernels for video, audio and text data are given. By linearly combining products of such kernels, one can obtain (non-trivial) feature representations for the multi-modal data!

We then showed that the functions $(x^\top z)^d$, $(1 + x^\top z)^d$ for $d \in \mathbb{N}$ are positive kernels (on the Euclidean space). Here is the sketch of the proof: we first showed that dot-product $x^\top z$ is a kernel[29]. This is because a gram matrix can be written

[2002] (refer definition 2.5).

[28] Justification of (2.31) in Schölkopf and Smola [2002] needs to be done as we did in lecture rather than as done in Schölkopf and Smola [2002]. Basically we need Cauchy-Schwartz inequality to hold for any two functions in Hilbert space rather than for kernels alone. In lecture we showed that this is indeed the case. Also in the lecture we gave a nice justification for the choice of the feature map, which is at the heart of the proof. We said that representing an object by its similarities with all other objects is the most obvious representation (and infact the richest representation).

[29] Infact, any inner-product is a kernel. Easiest proof of this is from equivalence of any finite-dimensional Hilbert space to Euclidean space and any infinite-dimensional (separable) Hilbert

as $X^\top X$ where X is the matrix containing the m datapoints in the columns. Now, $X^\top X$ is obviously symmetric and $z^\top X^\top X z = (Xz)^\top (Xz) \geq 0 \ \forall \ z$ and hence dot-products are kernels. Secondly we know that product of the two positive kernels $k_1(x,z) = (x^\top z)$ and $k_2(x,z) = (x^\top z)$ is again positive[30]. By induction, $(x^\top z)^d, d \in \mathbb{N}$ is a kernel. We gave a proof for the non-homogeneous case too.

Infact, usually one starts with $x^\top \Sigma y$, where $\Sigma \succeq 0$ and constructs kernels $k(x,z) = (x^\top \Sigma z)^d$ (known as the homogeneous polynomial kernel) and $k^{'}(x,z) = \left(1 + x^\top \Sigma z\right)^d$ (known as the non-homogeneous polynomial kernel). It is again an easy exercise to show that these are positive kernels (for a given $\Sigma \succeq 0$). By varying $d \in \mathbb{N}, \Sigma \succeq 0$ we obtain various kernels. Hence d, Σ are the parameters to a polynomial kernel.

After this, it was easy to show that $k(x,z) = e^{x^\top \Sigma z}$, is a positive kernel (by using the series expansion of e^x and the fact that polynomial kernels are positive and conic combinations of positive kernels is positive, which follows from simple linear algebra.). Usually one normalizes this kernel in the following way $k^{'}(x,z) = \frac{k(x,z)}{\sqrt{k(x,,x)k(z,z)}} = e^{-\frac{1}{2}(x-z)^\top \Sigma (x-z)}$. This is called the Gaussian kernel or the Radial Basis Function (RBF) kernel. Again, it is an easy exercise to show that normalized version of a positive kernel is positive.

Now that we have examples of kernels and the existence of Hilbert space theorem 2.4.1, the only thing left to be proved is the representer theorem, which says SVM-kind of problems require only inner-products rather than feature representations:

Theorem 2.4.2. *Let k be some positive kernel defined over an input space $\mathcal{X}$. Let $\mathcal{H}_k$ be the RKHS (or any other equivalent) and ϕ_k be the corresponding feature map. Suppose the model is all linear functions in that space i.e., $f(x) = \langle w, \phi_k(x) \rangle_{\mathcal{H}_k}$ with a (complexity) restriction $||w||_{\mathcal{H}_k} \leq W$. Now consider the problem of ERM:*

$$\begin{aligned} \min_{w \in \mathcal{H}_k} \quad & \textstyle\sum_{i=1}^m l(y_i \langle w, \phi_k(x_i) \rangle_{\mathcal{H}_k}), \\ \text{s.t.} \quad & ||w||_{\mathcal{H}_k} \leq W. \end{aligned} \tag{2.17}$$

Then an optimal solution of the ERM problem of the form: $w = \sum_{i=1}^m \alpha_i \phi_k(x_i)$ exists for some $\alpha_i \in \mathbb{R}$. Needless to say, the same statement holds for the Tikhonov and Morozov forms of the above Ivanov ERM problem.

Refer section 4.2 in Schölkopf and Smola [2002] for details.

space to l_2 space. In either case the gram matrix can be written as sum of gram-matrices obtained from each individual feature. And since sum of positive kernels is positive, we get the result.

[30] You may refer to any proof of Schur product theorem floating on the internet for this.

With this theorem, it is obvious that the problem (2.17) is equivalent to the following optimization problem in the Euclidean space:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^m} \quad & \textstyle\sum_{i=1}^m l\left(y_i \sum_{j=1}^m \alpha_j k(x_i, x_j)\right), \\ \text{s.t.} \quad & \sqrt{\alpha^\top G_k \alpha} \leq W. \end{aligned} \tag{2.18}$$

Here G_k is the matrix of all kernel evaluations on the training points and by theorem 2.4.1, it is the gram matrix of the training datapoints in $\mathcal{H}_k$. Moreover,

$$f(x) = \langle w, \phi_k(x) \rangle_{\mathcal{H}_k} = \sum_{i=1}^m \alpha_i k(x_i, x). \tag{2.19}$$

Hence both the ERM/SVM problem and the label prediction can be done using the kernel alone (and the feature representation ϕ_k is not required)! Infact, this "kernel trick" can be be used in any problem where dot-products are only involved. Refer section 14.2 in Schölkopf and Smola [2002] for example of such a problem.

Also, (2.19) clearly shows why non-linear functions will be induced by kernels like polynomial and Gaussian. The form of the learnt function will be some linear combination of the kernel functions with one argument fixed. In case of Gaussian kernels, we get that the function learnt is again a Gaussian function. On passing we also noted a specialty of the Gaussian kernel: theorem 2.18 in Schölkopf and Smola [2002]. This is special because for a linear kernel in n dimensions, the rank of the gram matrix (with any number of points) cannot be more than n i.e., the map of the input space is atmost an n-dimensional subspace in the feature space. However this result for a Gaussian kernel says that as the number of points increases the rank of gram-matrix increases and hence the map of the input space may be the entire feature space (which is possibly infinite dimensional)!

The examples till now are of kernels on Euclidean spaces. We now give an example of a kernel over distributions. Refer Jebara et al. [2004] for details. Such kernels are necessary in applications like Bioinformatics (refer section 8.2 in Jebara et al. [2004]) or in cases where the training datapoints are themselves noisy samples of the true inputs. In particular, one interesting result from the paper is: using a Gaussian kernel is like assuming there is a Normally distributed noise around the datapoints and we are classifying/regressing on these Normal distributions (refer section 3.1 in Jebara et al. [2004]). Hence using a Gaussian kernel would bring in some kind of robustness towards noise.

Now that one objective of this section is achieved (that of solving ERM in arbitrary spaces), lets move on to the second goal of whether some kernels lead to big enough function classes which well approximate the Bayes optimal? The answer is yes and such kernels are called as Universal kernels, which are the subject of study in the next section.

2.4.2 Universal Kernels

Lets begin with the question which is the "minimal" function class that approximates Bayes optimal well? The answer is provided by the Luzin's theorem [Folland, 1996], which gives that $\min_{f \in \mathcal{C}(\mathcal{X})} R[f] = R[f^{**}]$ i.e., the minimum risk in the set of all continuous functions ($\mathcal{C}(\mathcal{X})$) is equal to the Bayes optimal risk. Hence we would be happy if the function class induced by a kernel is $\mathcal{C}(\mathcal{X})$ or atleast dense in $\mathcal{C}(\mathcal{X})$, so that the minimum risk is close enough to the Bayes risk[31]. Hence we go with the following definition [Steinwart]:

Universal Kernel: A positive kernel k over an input space $\mathcal{X}$ is said to be a universal kernel (for that space) iff the function class induced by the kernel i.e., $\mathcal{F}_k = \{f \mid f(x) = \langle w, \phi_k(x) \rangle_{\mathcal{H}_k}, w \in \mathcal{H}_k\}$ is dense in the set of all continuous functions $\mathcal{C}(\mathcal{X})$.

Now lets show an example of a universal kernel on the Euclidean space. We claim that the Gaussian kernel (un-normalized one and hence the normalized one[32]) is universal. The proof[33] simply follows from the Stone-Weierstrass theorem [Rudin, 1976]. Refer theorem 1 in Steinwart for a version relevant to us.

It is easy to verify that Gaussian kernel satisfies all conditions of Stone-Weierstrass theorem: the function class induced by Gaussian kernel

$$\mathcal{F}_k = \left\{ f \mid f(x) = \sum_{i=1}^{m} \alpha_i e^{x_i^\top x}, x_i \in \mathbb{R}^n \right\},$$

is i) an algebra because it is ofcourse a vector space and product of two functions in this class will again be linear combinations of exponential functions and hence the space is closed under multiplication[34]. ii) non-vanishing because for any $x \in \mathbb{R}^n$, we can take $f(x) = k(z, x) = e^{z^\top x} > 0$ for any $z \in \mathbb{R}^n$. iii) separates $\mathcal{X}$ because $x, y \in \mathbb{R}^n, x \neq y \Rightarrow f_z(x) = e^{z^\top x} \neq f_z(y) = e^{z^\top y}$ for any $z \in \mathbb{R}^n$. Hence the Gaussian/RBF kernel is universal on the Euclidean space.

With this machinery one can show that ERM implemented using SVM with Gaussian kernel and model selection implemented using SRM leads to Bayes consistency. This is discussed in the subsequent section. On passing, we note the following paper Christmann and Steinwart [2010], which provides examples of universal kernels over non-Euclidean spaces.

[31] We are assuming true risk functional is continuous.

[32] The normalized version of a universal kernel is universal [Steinwart].

[33] You may also refer to Steinwart for an alternate proof which is more insightful.

[34] Note that closedness wrt. multiplication is what fails in case of linear or polynomial kernel. Infact one can show that such kernels are not universal [Steinwart].

2.5 Bayes Consistency

Though we know from the previous section that the function class induced by Gaussian kernels is big enough, using it for ERM may not lead to consistency (the estimation error might be high though the approximation error is low — because the conditional Rademacher average for this class blows up.). Hence the idea is to use the class of functions induced by Gaussian kernel with an additional restriction that $\|w\|_{\mathcal{H}_k} \leq W$. We know that this class is "good" in the sense that the conditional Rademacher average decays with m. Now we might get low estimation error but high approximation error. The trade-off can be achieved by SRM:

Consider the sequence of function classes induced by the Gaussian kernel: $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n, \ldots$, where $\mathcal{F}_n = \{f \mid f(x) = \langle w, \phi_k(x) \rangle_{\mathcal{H}_k}, w \in \mathcal{H}_k, \|w\|_{\mathcal{H}_k} \leq n\}$. Now if one implements SRM, we will achieve Bayes consistency because i) SRM is consistent (section 2.3.1) ii) $\cup_{i=1}^{\infty} \mathcal{F}_i = \{f \mid f(x) = \langle w, \phi_k(x) \rangle_{\mathcal{H}_k}, w \in \mathcal{H}_k\}$, which we already showed well approximates the Bayes optimal function. In summary, in this case, we get both low estimation error (as SRM is consistent) and low approximation error as the essential function class (union over the sequence) is big enough.

This completes the first milestone of our analysis: we are able to show an algorithm which achieves Bayes consistency i.e., an algorithm which produces a function whose risk is arbitrarily close to Bayesian risk with high probability (ofcourse this is an asymptotic result i.e., holds as $m \to \infty$). In the subsequent section we illustrate that the risk bounds/learning theory we have done till now might also be useful for learning problems other than Supervised learning we began with.

2.6 Other Applications of Risk Bounds: Kernel/Feature Learning

The learning theory developed till now is not only useful for showing theoretical results like consistency or for motivating SVM, but infact such results motivate many of the existing learning formulations. In this section we show yet another example of a learning formalization motivate from our (2.7,2.8) risk bound.

It is easy to see that the performance of a learning algorithm crucially depends on the feature representation for the input data, which in case of kernel-based algorithms (as the ones we use) depends on the kernel itself. Using the risk

bounds (2.7,2.8) one can infact study the influence of the kernel on the learning bound and hence try to optimize the kernel for the data in hand.

We refer to the following seminal paper: Lanckriet et al. [2004] for the details. Following is a short summary of this work along with the work in Rakotomamonjy et al. [2007].

One way to optimize the kernel is to consider conic combinations of given set of p base kernels $k_1, \ldots, k_p$ and then learn the optimal weights in the conic combination i.e., $k = \sum_{i=1}^{p} \lambda_i k_i,\ \lambda_i \geq 0 \forall\ i$ and the weights λ_i are learnt. Such a kernel learning setting would be particularly interesting for multi-modal data[35], where each base kernel is constructed from a different mode of describing the data.

Let $\mathcal{H}_i, \phi_i$ be the RKHS, feature map with the kernel k_i and let $\hat{\mathcal{H}}_i, \hat{\phi}_i$ be those for the kernel $\lambda_i k_i$. It is easy to see that $\sqrt{\lambda_i}\phi_i = \hat{\phi}_i$ and the RKHS of k is direct sum of individual RKHS i.e., $\mathcal{H} = \hat{\mathcal{H}}_1 \oplus \ldots \oplus \hat{\mathcal{H}}_p$, inner product for k is $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{p} \langle f_i, g_i \rangle_{\hat{\mathcal{H}}_i}$ (here, f_i, g_i represent the component/projection of f, g onto the i^{th} RKHS). Using this notation, a linear function in $\mathcal{H}$ can be written as: $f(x) = \langle w, \hat{\phi}(x) \rangle_{\mathcal{H}} = \sum_{i=1}^{p} \langle w_i, \hat{\phi}(x)_i \rangle_{\hat{\mathcal{H}}_i} = \sum_{i=1}^{p} \sqrt{\lambda_i} \langle w_i, \phi_i(x) \rangle_{\hat{\mathcal{H}}_i}$.

From the risk bounds (2.7,2.8) it follows that the capacity of the induced function class is bounded as long as $\|w\|_{\mathcal{H}}^2 = \sum_{i=1}^{p} \|w_i\|_{\hat{\mathcal{H}}_i}^2 \leq W$ and $trace(\sum_{i=1}^{p} \lambda_i K_i) = \sum_{i=1}^{p} \lambda_i trace(K_i) \leq T$ for some W and T. Now, one can write the ERM problem as:

$$\begin{aligned} \min_{\lambda, w} \quad & \textstyle\sum_{i=1}^{m} l(y_i \sum_{i=1}^{p} \sqrt{\lambda_i} \langle w_i, \phi_i(x) \rangle_{\hat{\mathcal{H}}_i}), \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{p} \|w_i\|_{\hat{\mathcal{H}}_i}^2 \leq W,\ \lambda_i \geq 0,\ \sum_{i=1}^{p} \lambda_i trace(K_i) \leq T \end{aligned} \tag{2.20}$$

In this form it is not clear whether (2.20) is a convex program. Convexity is seen by replacing $\hat{w}_i = \sqrt{\lambda_i} w_i$ and re-writing (2.20) as:

$$\begin{aligned} \min_{\lambda, \hat{w}} \quad & \textstyle\sum_{i=1}^{m} l(y_i \sum_{i=1}^{p} \langle \hat{w}_i, \phi_i(x) \rangle_{\hat{\mathcal{H}}_i}), \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{p} \frac{\|\hat{w}_i\|_{\hat{\mathcal{H}}_i}^2}{\lambda_i} \leq W,\ \lambda_i \geq 0,\ \sum_{i=1}^{p} \lambda_i trace(K_i) \leq T \end{aligned} \tag{2.21}$$

This program is convex[36] as $\frac{\|\hat{w}_i\|_{\hat{\mathcal{H}}_i}^2}{\lambda_i}$ is a convex function in $\hat{w}_i$ and λ_i [Boyd and Vandenberghe, 2004]. The work of Rakotomamonjy et al. [2007] presents an efficient projected gradient descent algorithm for solving (2.21).

Intuitively, the condition $\sum_{i=1}^{p} \lambda_i trace(K_i) \leq T$ implies that the weights for kernels where the data is spread out will be less. Hence the ERM problem above

[35] For e.g., a meeting described using video, audio, scribes etc. Here video, audio and scribes are the different modes

[36] Provided the loss is a convex function.

looks for a kernel combination that gives a good trade-off for: (low) empirical risk, (large) margin and (low) radius/spread of data.

Chapter 3

Semi-Supervised Learning

The learning theory bounds presented in the previous chapter suggest that more the number of training examples, the better. However obtaining training data is a laborious task, primarily because it involves an expert to provide labels for the simulated/synthesized datapoints. What we observed is that in many application domains, obtaining training datapoints alone without the labels is easy (for eg. text categorization). The key question then is can unlabeled examples be useful?

We argued intuitively that there are atleast three ways in which unlabeled data might be useful: i) Inductive Learning: In this context, the question translates to asking whether knowledge of $p(x)$ (with enough number of unlabeled examples $p(x)$ can be accurately estimated) improves our guess of $p(y/x)$? Empirical studies on object recognition in babies show that this indeed is the case. Babies tend to identify objects easily if their names are frequently heard by them!

The next scenario is transduction: where labeled and unlabeled examples are provided and the goal is to label the unlabeled correctly. This is in contrast with the case of induction where the goal is to predict label of any unseen example irrespective of whether observed during training stage or not. Clearly, one would expect this to be a "simpler" task than induction and hence better learning bounds.

Thirdly, unlabeled examples might be useful to restrict the learning algorithm to not over-generalize. For eg. consider the task of classifying digit 2 vs 4. A SIL learner would be given examples of 2s and 4s. However the learning algorithm might start over-generalizing and try to classify a digit 1 to one of these classes. In such cases, perhaps one wants to provide examples of all digits which are not $2, 4$, which is called the universum and insist that the universum should not be classified or classified with low confidence. This idea is explored in Weston et al. [2006], Sinze et al. [2007]. In summary, unlabeled examples might

be useful for improved generalization, improved learning bounds, or restricting over-generalization etc.

The focus of this course is on induction and transduction. The slides in the appendix (end of this notes) give a rough definition of both these cases and provide a comparison. One key observation is a inductive algorithm can be used for transduction and vice-versa. We will begin with analysis of Semi-supervised Transductive learning (SSTL)[1] in the subsequent section.

3.1 Semi-Supervised Transductive Learning

We began with discussing the paper by Derbeko et al. [2004], which provides learning bounds for the case of transduction. Later on we will devise algorithms based on these bounds.

3.1.1 Transductive Learning Theory

Transduction can be analyzed under two settings (see page 120 in Derbeko et al. [2004]). In setting-1, a set of $m + u$ datapoints is given. At random (selection without replacement) m of them are chosen and given to an expert for labeling. We further assume that $p(y/x)$ is peaked i.e., non-noisy supervisor ($p(y/x)$ is either 0 or 1). With such a scheme, training examples are no more independent. Setting-2, is the extension of iid case to SSL. Theorem 2 in the paper connects both the settings. As mentioned in the paper, the analysis is easy with setting-1 and is henceforth employed in the reminder of this notes.

Following the notation in the paper, we define $R_f(X_u)$ (refer eqn. (1) in paper) to be the true risk with f on the unlabeled set X_u. The goal in transduction is to find a $f \in \mathcal{F}$ that minimizes this risk. $\hat{R}_f(S_m)$ is the empirical risk with f on the labeled set S_m. The following couple of observations provide a way to write a learning bound:

- Lemma 5 that relates $R_f(X_u)$ and $R_f(X_{m+u})$.
- Eqn. 20 that shows $\mathbb{E}[\hat{R}_f(S_m)] = R_f(X_{m+u})$.

In view of the above, we first try to bound $P[\mathbb{E}[\hat{R}_f(S_m)] - \hat{R}_f S_m > \epsilon]$ (for the 0-1 loss or truncated hinge-loss case) using the Sefling bound (theorem 14 in

[1]Henceforth, SSL stands for Semi-Supervised Learning. SSIL stands for Semi-Supervised Inductive Learning.

the paper). Then using Lemma5 we have that, with probability atleast $1-\delta$, the following holds for a given f:

$$R_f(X_u) \leq \hat{R}_f(S_m) + \sqrt{\frac{(m+u)(u+1)}{2mu^2} \log\left(\frac{1}{\delta}\right)}$$

Note that as $m \to \infty, u \to \infty$, the confidence term in the above bound goes to zero. This suggests the principle of ERM for transduction. Again, in the case of finite-sized[2] $\mathcal{F}$, using a union bound one obtains[3]: with probability atleast $1-\delta$:

$$R_f(X_u) \leq \hat{R}_f(S_m) + \sqrt{\frac{(m+u)(u+1)}{2mu^2} \log\left(\frac{|\mathcal{F}|}{\delta}\right)}, \ \forall \ f \in \mathcal{F} \tag{3.1}$$

With such a uniform bound, it is easy to show that ERM is consistent. Hence PAC learning can be performed in case of Semi-supervised transduction. While this result is interesting, the bound in the above raises the following concerns: i) According to the bound, learning may not possible in cases m or u alone go to infinity while the other is constant. ii) is the learning rate faster than in supervised learning?

Uma and Agam answered i) by looking at an alternative bound given by Vapnik that is tighter than (3.1). Refer corollary 9 in the paper for details of the Vapnik bound. They concluded that PAC learning is possible when either of m or u go to infinity and empirical risk is zero. They also observed that the Vapnk's bound improves as u increases for a given m (whereas with the loose bound (3.1), lesser the u, for a fixed m, the better). Answering ii) conclusively is more difficult. What is possible is a comparison of bounds for SSL and SIL cases, which can be taken up by some of you[4]. Though the bound in (3.1) is loose, it motivate SSL algorithms, whereas the Vapnik bound, though tight, is implicit and hard to analyze.

Note that the ERM principle does not use any unlabeled data. However, we wish to derive a learning algorithm which uses unlabeled data and perhaps improves the generalization with the given labeled data.

The way out is SRM, which is motivated by the bound (3.1). The idea is to use the unlabeled data to come up with a "good" structure over the function class.

[2]In setting-1 of transduction, since the input space is finite, any function class indeed looks like a finite one. Atleast in case of binary classification with 0-1 loss this is clear. In case of regression or other problems one can choose a suitable loss where any function class is equivalent to a finite one.

[3]A slight variant of this result is proved in theorem 22 of the paper, which is the key result in the paper. The only difference is while taking union bound we consider probabilities as $\delta p(f)$, where $p(f)$ is some prior probability in choosing f. This leads to eqn. (19) in the paper.

[4]Attempt it in case you want bonus marks :)

Since the form of bound remains same in case of induction (2.4) or transduction (3.1), in the following we forget this distinction and focus on how to derive clever structures that use unlabeled data. Each such a structure will lead to a formulation and corresponding SSL algorithm.

3.2 Semi-Supervised Learning Formulations

In this section we present various SSL formulations. Each one of them is motivated from SRM and differ only in the structure built over the function class.

The first example we gave was the case of binary classification where the domain knowledge provides the expected +ve to -ve datapoints ratio i.e., the class balance is known. Without loss of generality, let us assume that the in the given application it is expected that the classes are balanced i.e., no. positives among the m+u datapoints is close to the no. negatives. As mentioned earlier, since the input space is the $m + u$ datapoints and is finite, the effective size of the usual linear function class is finite and $\leq 2^{m+u}$ (all possible labelings). In such a scenario, one would perhaps arrange the usual linear function class as follows: $\mathcal{H}_0 \subset \mathcal{H}_1 \subset \ldots \subset \mathcal{H}$, where $\mathcal{H}_i$ represents all those labelings with the given linear function class that achieve a deviation in class balance by atmost i, i.e., the difference in number of +ves and -ves in the $m + u$ examples with classifiers in $\mathcal{H}_i$ is less than or equal to i. Basically, the structure is built such that classifiers who achieve the right class balance are preferred over the others. Note that this structure can be obtained without using any label information i.e., prior to being exposed to the training data. It only uses the input space, which is the set of the $m + u$ datapoints. In the following section we present another structure discussed in Joachims [1999].

3.2.1 Transductive SVM

The bounds in supervised learning theory motivate an alternative structure. The bounds suggest that linear classifiers that achieve large-margin (in turn high confidence in predictions) are preferred over the others. Hence the idea is to use the following structure: $\mathcal{F}_1 \subset \ldots \subset \mathcal{F}_W \subset \ldots$, where

$$\mathcal{F}_W = \left\{ f \mid \exists w \ni f(x) = w^\top x,\ \|w\| \leq W,\ |w^\top x_i| \geq 1\ \forall\ i = 1, \ldots, m+u \right\}.$$

In plain words, $\mathcal{F}_W$ is all those linear functions that achieve a margin greater than or equal to $\frac{1}{W}$ on all the datapoints. Basically, the structure prefers predictors with high confidence.

Lets write down the corresponding ERM problem as a mathematical program in Morozov form (using hinge-loss):

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \tfrac{1}{2}||w||^2, \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m} \max\left(0, 1 - y_i w^\top x_i\right) \leq A, \ |w^\top x_i| \geq 1 \ \forall \ i = 1, \ldots, m+u \end{aligned}$$

Motivated by the hard-margin SVM (2.15), Joachims considered the case $A = 0$ and re-wrote the above formulation as:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \tfrac{1}{2}||w||^2, \\ \text{s.t.} \quad & y_i w^\top x_i \geq 1 \ \forall i = 1, \ldots, m; \ |w^\top x_i| \geq 1 \ \forall \ i = m+1, \ldots, m+u \end{aligned}$$

One can re-write the above as[5]:

$$\begin{aligned} \min_{w \in \mathbb{R}^n, y_{m+1}, \ldots, y_{m+u}} \quad & \tfrac{1}{2}||w||^2, \\ \text{s.t.} \quad & y_i w^\top x_i \geq 1 \ \ y_i \in \{-1, 1\}, \ \forall i = 1, \ldots, m+u \end{aligned}$$

Again, drawing analogy from SVMs (2.14), we have the following soft-version of the above:

$$\begin{aligned} \min_{w \in \mathbb{R}^n, y_{m+1}, \cdots, y_{m+u}} \quad & \tfrac{1}{2}||w||^2 \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m} \max\left(0, 1 - y_i w^\top x_i\right) \leq A_1, \sum_{i=m+1}^{m+u} \max\left(0, 1 - y_i w^\top x_i\right) \leq A_2, \ y_i \in \{-1, 1\}, \ \forall i = m+1, \ldots, m+u \end{aligned} \tag{3.2}$$

Note that the OP2 formulation in the Joachims paper is same as the Tikhonov form of (3.2).

Though visually the above transductive SVM formulations closely resemble the SVM, they are not convex. Infact, OP1 and OP2 are combinatorial optimization problems and in general, one cannot do better than exhaustive enumeration: one has to enumerate all possible labels $y_i, i = m+1, \ldots, m+u$, which are 2^u in number, and solve a regular SVM those many times and find the minimum objective of them in order to solve (3.2). Since this strategy is computationally infeasible for even moderately sized problems, Joachims suggests an algorithm which finds a local optima for the combinatorial problem. Refer figure 4 in paper for details. We commented that there are multiple strategies suggested by various researchers to solve this problem (refer chapter 3 in Sindhwani [2007] for a survey). Each has its own merit and de-merit. However the way Joachims solves it, the methodology has striking similarity with self-training [Yarowsky, 1995].

Finally, we noted that one can implement SRM with the bound (3.1) in this case provided one knows the essential number of labelings possible with $\mathcal{F}_W$.

[5]This is the final form (OP1) considered in the Joachims paper.

To this end, Vapnik provided a simple bound on the number of labelings. Refer theorem 1 in Joachims paper for the details[6]. In the subsequent another kind of structure over the function class is studied.

3.2.2 Manifold Regularization

We began looking at alternative explanations for the large-margin principle. We argued that large-margin principle, in a way, implements the following related ideas: i) prefers high confidence predictions ii) prefers low density separation (discriminating hyperplane passes through areas of low likelihood of observing datapoints) iii) "clustering assumption": close-by datapoints have close-by labels.

Motivated by the clustering assumption, we wished to arrive at a structure that explicitly implements it. Accordingly, we came-up with this structure: $\mathcal{F}_1 \subset \ldots \subset \mathcal{F}_W \subset \ldots$, where

$$\mathcal{F}_W = \left\{ f \mid \exists w \ \ni \ f(x) = w^\top x, \ \|w\| \leq W_1, \sum_{i=1}^{m+u} \sum_{j=1}^{m+u} M_{ij} \left(w^\top x_i - w^\top x_j \right)^2 \leq W_2 \right\}.$$

here, M_{ij} represents how close x_i, x_j are and is such that $M_{ij} = M_{ji} \geq 0$. One may use $M_{ij} = \frac{1}{\|x_i - x_j\|}$ or $M_{ij} = e^{-\|x_i - x_j\|}$ etc. or any positive kernel[7].

In plain words, this structure while preferring large margins (when used with variants of hinge-loss), also prefers those functions that vary slowly when M_{ij} is high (i.e., when x_i and x_j are close) and varies considerably whenever M_{ij} is low. Note that this is one particular way of implementing the clustering assumption.

Now lets write the corresponding ERM problem as a mathematical program (Ivanov form):

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \textstyle\sum_{i=1}^m l(y_i, w^\top x_i), \\ \text{s.t.} \quad & \|w\| \leq W_1, \ \textstyle\sum_{i=1}^{m+u} \sum_{j=1}^{m+u} M_{ij} \left(w^\top x_i - w^\top x_j \right)^2 \leq W_2 \end{aligned}$$

The following Tikhonov form of the above is discussed in detail in Belkin et al. [2006]:

$$\min_{w \in \mathbb{R}^n} \quad \frac{1}{2}\|w\|^2 + C_1 \sum_{i=1}^{m} l(y_i, w^\top x_i) + C_2 \sum_{i=1}^{m+u} \sum_{j=1}^{m+u} M_{ij} \left(w^\top x_i - w^\top x_j \right)^2 \tag{3.3}$$

[6]Note that Abhinav corrected the typos in this theorem.

[7]Note that the matrix M with entries as $M_{ij} \geq 0$ needs to be symmetric. It need NOT be psd. However one may employ a positive kernel such that $M_{ij} = k(x_i, x_j)$. In this case M will be psd.

The third term in the objective of (3.3) is usually referred to as the manifold regularization term: it penalizes large deviations in predictions for near-by datapoints and smoothens the manifold of the optimal function learnt.

The above formulation is interesting in multiple ways: i) unlike the transductive svm (3.2), the above formulation is a convex program[8] and hence can be solved efficiently. Infact, with the square loss, the Tikhonov form is unconstrained minimization of a convex quadratic function, which has an analytical solution ii) Theorem 2 in the paper presents a representer theorem[9], which gives that $w = \sum_{i=1}^{m+u} \alpha_i x_i$. Hence this formulation can be kernelized (extended to non-linear functions using the kernel trick).

We then noted an interesting way of re-writing (3.3):

$$\min_{\hat{w} \in \mathbb{R}^n} \quad \frac{1}{2}\|\hat{w}\|^2 + C_1 \sum_{i=1}^{m} l(y_i, \hat{w}^\top \hat{x}_i), \tag{3.4}$$

where $\hat{w} = \left(I + 4C_2 X L X^\top\right)^{\frac{1}{2}} w$, $\hat{x}_i = \left(I + 4C_2 X L X^\top\right)^{\frac{-1}{2}} x_i$, X is the data matrix containing labeled and unlabeled datapoints as column vectors and $L = D - M$, D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{m+u} M_{ij}$. Infact, the matrix L is well studied and is known as the Laplacian of the graph with adjacency matrix given by M.

Note that (3.4) is nothing but an SVM constructed using the labeled points alone and kernel as: $k(x_i, x_j) = \hat{x}_i^\top \hat{x}_j = x_i^\top \left(I + 4XLX^\top\right)^{-1} x_j$. The corresponding gram-matrix G_k for the $m+u$ points is $G_k = X^\top \left(I + 4XLX^\top\right)^{-1} X$. Thus, the manifold regularization formulation (3.3) can be understood as a two step process: i) Get the right kernel using the labeled and unlabeled examples. ii) train an SVM using labeled examples.

This key observation motivates alternative SSL algorithms where the step-i) of choosing the right kernel is done leading to a kernel different than in (3.4). Since the key ingredient in the vanilla kernel in (3.4) is L, in the following text we will explore some interesting properties of L, which will later on motivate alternate SSL algorithms.

The graph Laplacian, L, is a psd matrix[10]: $z^\top L z = \sum_{i=1}^{m+u} \sum_{j=1}^{m+u} M_{ij} (z_i - z_j)^2 \geq 0 \ \forall \ z$. Infact, it is a diagonally dominant matrix. Since L is psd we have its eigen-value decomposition (EVD): $L = V \Lambda V^\top = \sum_{i=1}^{m+u} \lambda_i v_i v_i^\top$, where V is the matrix with column vectors as v_i, Λ is a diagonal matrix with entries as $\lambda_i \geq 0$

[8] Provided the loss is convex.

[9] Contrast this with the usual representer theorem: the only difference is here the unlabeled examples are also involved in the linear combination!

[10] Provided the adjacency matrix M is non-negative and symmetric, which is indeed the case in our discussion.

and $v_i^\top v_j$ is 0 if $i \neq j$ and is 1 if $i = j$. Without loss of generality lets assume $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_{m+u}$. The EVD of L provides key insights into the extent of connectedness of the graph:

Theorem 3.2.1. *A graph has k connected components if and only if the number of 0 eigen-values (i.e., the algebraic multiplicity of 0) of L is k. Infact, a basis for the eigen-space of 0 is $\{v_1, v_2, \ldots, v_k\}$ where each v_i has non-zero equal-valued entries for the nodes in the i^{th} component and has zero as entry for all the other nodes.*

Refer Proposition 2.3 in Mohar [1997] for a proof. Also, $z^\top Lz$ will be higher as z has more component from the larger eigen-vectors. Infact, it will be highest if $z = v_{m+u}$, the last eigen-vector and the corresponding eigen-value λ_{m+u} gives $v_{m+u}^\top L v_{m+u}$, the weighted mean-square variation in entries of $z = v_{m+u}$. In other words, the eigen-vectors corresponding to large eigen-values have large deviations in entries; while those corresponding to small eigen-values have less deviations and for the zero eigen-values, the variation is zero in each connected-component. Refer figure 1.1 in Zhu et al. [2006] for a visualization of this theorem and the above observation. We will present alternate SSL formulations that exploit these key observations in the subsequent section.

3.2.3 SSL via Kernel Learning

This section presents alternative kernels to be employed in SSL. Most of the works try to choose a kernel k from among the following family of kernels: all those kernels with whom the gram-matrix with $m+u$ datapoints looks like $G_k = \sum_{i=1}^{m+u} \mu_i v_i v_i^\top$, where v_i are the eigenvectors of graph Laplacian and $\mu_i \geq 0$ are the weights (eigen-values of gram-matrix) that need to be chosen. Different works suggest different schemes for choosing the μ_is.

Lets have a closer look at the manifold regularization term: we want the term $\sum_{i=1}^{m+u} \sum_{j=1}^{m+u} M_{ij} \left(\langle w, \phi_k(x_i)\rangle_k - \langle w, \phi_k(x_j)\rangle_k\right)^2$ to be low. Lets use representer theorem: $w = \sum_{i=1}^{m+u} \alpha_i \phi_k(x_i)$. Substituting this in the term of interest gives: $\alpha^\top G_k L G_k \alpha$ should be low. Whereas SVM, as per (2.18), chooses α such that $\alpha^\top G_k \alpha$ is low. Hence, choice of μ_i (and hence G_k) must be such that $\alpha^\top G_k \alpha$ is low $\Rightarrow \alpha G_k L G_k \alpha$ is low. One simple choice is $G_k = L^{-1}$ i.e., $\mu_i = \frac{1}{\lambda_i}$, then both the quadratic terms are equal. Since L may not always be invertible, hence Smola and Kondor [2003] suggest the following kernel: $G_k = (L + \epsilon I)^{-1}$, where ϵ is a small quantity. This kernel is called as the regularized Laplacian kernel.

From the above discussion, it is easy to see that, any choice of μ_i that is inversely proportional to λ_i is fine. Accordingly Smola and Kondor [2003] suggest

the diffusion kernel: $\mu_i = e^{-\frac{\sigma^2}{2}\lambda_i}$ (here σ is the diffusion kernel parameter). Other choices of μ are listed on Smola and Kondor [2003], Zhu et al. [2006].

Given this wide choice of SSL kernels, the question arises: which one is the best. One can perform a equivalent of cross-validation etc. to find the best. Alternatively, in Zhu et al. [2006], it is suggested that the weights be learnt (i.e., learn the kernel) from the training data itself. One way to do this is by using the multi-modal kernel learning formulation (2.21) with the base kernels as: $k_i = v_i v_i^\top$. As we know, (2.21) returns the weights μ_i with which k_i needed to be combined: $k = \sum_{i=1}^{m+u} \mu_i k_i$. Additionally one may want to explicitly put order constraints: $\mu_1 > \mu_2 > \ldots > \mu_{m+u}$, so that the inverse proportionality criteria mentioned above is met. This leads to the following formulation:

$$\begin{aligned} \min_{\mu, \hat{w}} \quad & \textstyle\sum_{i=1}^{m} l(y_i \sum_{i=1}^{p} \langle \hat{w}_i, \phi_i(x) \rangle_{\hat{\mathcal{H}}_i}), \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{p} \frac{\|\hat{w}_i\|^2_{\hat{\mathcal{H}}_i}}{\mu_i} \leq W, \ \mu_i \geq 0, \ \sum_{i=1}^{p} \mu_i trace(G_{k_i}) \leq T, \\ & \mu_1 \geq \mu_2, \mu_2 \geq \mu_3, \ldots, \mu_{m+u-1} \geq \mu_{m+u}. \end{aligned}$$

Note that the only difference from (2.21) is the additional order constraints that are appropriate in context of SSL.

The above formulation in addition to the kernel weights μ_i, also learns the optimal linear function (w). One may instead devise a formulation similar in spirit to the above that returns only the kernel weights, so that the learnt kernel can be employed for any task using any kernel-based algorithm. This direction is explored in Zhu et al. [2006]. In order to understand this we first need to come up with a generic criteria that characterizes a good kernel, given the training data. One such criteria is kernel target alignment Cristianini et al. [2002].

The basic idea is very simple. In case we know the true labels of x, say $y(x)$, then the ideal kernel is $k_I(x, z) = y(x)y(z)$. The idea is to maximize the match/alignment of the given kernel to this ideal kernel on the labeled training data. This is similar to minimizing empirical risk. Consistency in this case is proved in Cristianini et al. [2002]. We didn't go into the learning theory details, however used the alignment of kernel to k_I. Suppose the gram-matrix with the kernel to be learnt is G_k and suppose the ideal gram-matrix is $G_{k_I} = yy^\top$, where y is the vector with entries as labels of training datapoints. Then one would like to maximize $\frac{\langle G_k(labeled), yy^\top \rangle_F}{\sqrt{\langle G_k(labeled), G_k(labeled) \rangle_F \langle yy^\top, yy^\top \rangle_F}}$, where $G_k(labeled)$ is the gram-matrix for the labeled datapoints alone and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner-product. This expression is nothing but the cosine of the angle between the given and ideal gram-matrices and hence needs to be maximized for alignment. Using this criteria, Zhu et al. [2006] arrive at a new formulation (refer eqns. 1.18-1.22 in the paper). We leave the details to the reader.

3.2.4 SSL with Multiple Manifolds

Motivated from a real-world Bioinformatics application, here we consider the case where multiple graphs representing closeness in datapoints are given and one wishes to perform SSL. The idea is discussed in detail in Shin et al. [2009]. In the following we present the key ideas.

One way to handle this problem is to create kernels from each graph (perhaps diffusion or regularized Laplacian etc.) then use them as base kernels in the multi-modal kernel learning (2.21). An alternate way is to extend (3.3) to multiple graphs case: let $L_1, \ldots, L_p$ be the Laplacians of the given p number of graphs. A simple idea is to add the corresponding edge weights in each graph[11] and use it in manifold regularization:

$$\min_{w \in \mathbb{R}^n} \quad \frac{1}{2}\|w\|^2 + C_1 \sum_{i=1}^{m} l(y_i, w^\top x_i) + C_2 w^\top X \sum_{i=1}^{p} L_i X^\top w$$

This formulation gives equal importance to all the graphs. Sometimes one may want to regularize the worst-case graph:

$$\min_{w \in \mathbb{R}^n} \quad \frac{1}{2}\|w\|^2 + C_1 \sum_{i=1}^{m} l(y_i, w^\top x_i) + C_2 \max_{i=1,\ldots,p} w^\top X L_i X^\top w$$

This minimizes the maximum deviation in the prediction values. In lecture we derived a dual of this for the case of square-loss (regression):

$$\begin{array}{ll} \min\limits_{\mu \in \mathbb{R}^p} & y^\top X_l^\top \left(I + 2C_1 X_l X_l^\top + 2X(\sum_{i=1}^{p} \mu_i L_i) X^\top\right)^{-1} X_l y, \\ \text{s.t.} & \mu_i \geq 0,\ \sum_{i=1}^{p} \mu_i = C_2, \end{array}$$

where X_l is the $n \times m$ data matrix for the labeled datapoints. The μ_is play the role of weights for the graphs indicating their importance. Whenever $\mu_i = 0$, the corresponding graph is not used in the solution. One efficient way to solve this optimization problem is projected gradient descent. The partial differential of the objective in the above wrt. μ_i is given by

$$y^\top X_l^\top \left(I + 2C_1 X_l X_l^\top + 2X(\sum_{i=1}^{p} \mu_i L_i) X^\top\right)^{-1} L_i \left(I + 2C_1 X_l X_l^\top + 2X(\sum_{i=1}^{p} \mu_i L_i) X^\top\right)^{-1} X_l y.$$

Since the L_i are all sparse matrices, the term $\left(I + 2C_1 X_l X_l^\top + 2X(\sum_{i=1}^{p} \mu_i L_i) X^\top\right)^{-1} X_l y$ can be computed very efficiently. Also, since the feasibility set for this problem is a simplex, the projection step is very easy. The authors claim that this algorithm runs faster than the multi-modal kernel learning algorithm while achieving similar accuracy.

[11] This is equivalent to adding all the Laplacians

Chapter 4

Structured Prediction

We began by looking at learning problems where the goal is to construct a function $g : \mathcal{X} \to \mathcal{Y}$; however, unlike the cases dealt with till now e.g., binary/multi-classification, regression etc., here the labels $y \in \mathcal{Y}$ need not be simple numbers but some complex objects like sequences, graphs, sets etc. In order to get more insight, we looked at applications where such complex prediction functions need to be built: e.g., sequence labeling problem, parse tree construction in NLP, retrieval of diverse web-pages for a given topic etc.

We noted that in each of these applications: i) the label to be predicted in not a number but is a sequence, tree, set respectively ii) The space of labels $\mathcal{Y}$ is extremely large — typically exponential in some input size[1]. iii) As a result, training data of examples of each label $y \in \mathcal{Y}$ cannot be created. Hence one may need to generalize across labels too in addition to generalizing across inputs. And, ofcourse one may need to exploit the specific structure in the label/output-space for such a generalization.

This brings us to the notion of structured prediction[2], where the prediction task is expected to exploit the structure in the label-space. With this in mind, we changed our perspective of looking at learning problems as task of building $g : \mathcal{X} \to \mathcal{Y}$ to that of building $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which measures how compatible is an input $x \in \mathcal{X}$ to an output/label $y \in \mathcal{Y}$. Though both are in some sense equivalent[3], the latter perspective is more attractive for structured prediction as it gives a way to generalize across inputs and outputs simultaneously (as the problem is essentially that where the input is (x, y) and output/label is a real).

[1] Exponential in length of sequence for first two applications and exponential in the number of topic-relevant pages on web in the third case.

[2] This chapter is a summary of Tsochantaridis et al. [2005].

[3] We noted this equivalence starting with per-class models in the lecture.

Recalling the story with kernels, we considered the following form for the compability function $f(x,y) = \langle w, \phi(x,y) \rangle$, where $\phi : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}_{\bar{k}}$ is some feature map from direct-sum space of inputspace $\mathcal{X}$ and outputspace $\mathcal{Y}$ to some Hilbert space $\mathcal{H}_{\bar{k}}$ induced by a positive kernel $\bar{k} : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$. Unlike the kernels we encountered till now that measure similarity between input pairs, this kernel measures similarity betweens pairs of input-output pairs. From our experience of kernels, it is easy to see that with such linear functions in kernel induced spaces, it is easy to handle both complex inputs as well as complex outputs (e.g., input can be a set, output can be a graph etc.). The other interpretation of $f(x,y) = \langle w, \phi(x,y) \rangle$ in case of finite dimensional $\mathcal{H}_{\bar{k}}$ is: f is a weighted sum of basic compatibility functions ϕ_i.

We then noted that the training data $\mathcal{D} = \{(x_i, y_i) \mid \forall\ i\}$ essentially says that compatibility of x_i with y_i, given by $f(x_i, y_i)$, is greater than (or equal to) that of x_i with any $y \in \mathcal{Y} \backslash \{y_i\}$, given by $f(x_i, y)$. That is, $\langle w, \phi(x_i, y_i) \rangle - \max_{y \in \mathcal{Y} \backslash \{y_i\}} \langle w, \phi(x_i, y) \rangle \geq 0$. Employing hinge loss this essentially is the same as saying for each training datapoint x_i, we must minimize the hinge-loss term: $\max \left(0, 1 - \left(\langle w, \phi(x_i, y_i) \rangle - \max_{y \in \mathcal{Y} \backslash \{y_i\}} \langle w, \phi(x_i, y) \rangle \right) \right)$. With such a hinge-loss term and a linear model an SVM-kind of formulation (refer eqn. (7) in Tsochantaridis et al. [2005]) is immediate. The authors call this formulation the struct-SVM[4].

Representer theorem was immediate giving: $w = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y} \backslash \{y_i\}} \alpha_{iy} \delta_\phi(x_i, y)$, where $\delta_\phi(x_i, y) = \phi(x_i, y_i) - \phi(x_i, y)$. Using this, both the formulation (7) above and the function f can be computed using the kernel: $k\left((x_i, y), \left(x_j, y^{'}\right)\right) = \langle \delta_\phi(x_i, y), \delta_\phi(x_j, y^{'}) \rangle$. It is easy to see that given $\bar{k}$ (above) one can obtain k and vice-versa.

We then went over the example applications in section 4 of the paper, and realized the typical ϕ or k employed. We noted that the easiest way of obtaining a kernel k of the above form was by using a kernel k_1 on the inputspace and a kernel k_2 on the outputspace: $k\left((x_i, y), \left(x_j, y^{'}\right)\right) = k_1(x_i, x_j) + k_2\left(y, y^{'}\right)$ or $k\left((x_i, y), \left(x_j, y^{'}\right)\right) = k_1(x_i, x_j)\, k_2\left(y, y^{'}\right)$.

We then went on to derive the Lagrange-dual of the struct-SVM (refer section 3.1 in the paper). We noted that the dual problem is a convex QP, however involving $m(|\mathcal{Y}| - 1)$ number of variables. This makes solving the dual challenging as $|\mathcal{Y}|$ itself could be exponential in some input-dimension[5]. We then intuitively argued that many of the dual variables must be zero (equivalently many inequalities in primal are non-active) at optimality. This motivated us to study an active-set

[4]Note the variants of the above formulation given in the paper: eqns. (6),$\text{SVM}_1^{\Delta s}$,$\text{SVM}_1^{\Delta m}$ and the corresponding variants with squared-hinge-loss: $\text{SVM}_2^{\Delta s}$,$\text{SVM}_2^{\Delta m}$. In the lectures and notes we focus on (7) alone; however the analysis etc. remain similar.

[5]It is exponential in length of sequence in case of sequence labeling problem.

(cutting-plane) algorithm (refer section 3.2 and 3.3 for details). It was easy to see that the per-iteration complexity of the algorithm is polynomial provided the inference problem is tractable (polynomial complexity). Moreover, the number iterations can be shown to be polynomial in m (refer theorem 18 in paper). Hence the active set algorithm is guaranteed to optimally[6] solve the struct-SVM problem in polynomial time.

We then went on and looked at a specific application of struct-SVM in optimizing multivariate performance measures [Joachims, 2005].

[6] upto some numerical precision.

Bibliography

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3: 463–482, 2002.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1248547.1248632`.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

Olivier Bousquet, Stephane Boucheron, and Gabor Lugosi. Introduction to Statistical Learning Theory. *Advanced Lectures on Machine Learning Lecture Notes in Artificial Intelligence*, 3176:169–207, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

C. J .C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

O. Chapelle, V. N. Vapnik, O. Bousquet, and S. Mukherjee. Choosing Multiple Parameters for Support Vector Machines. 46(1–3):131–159, 2002.

H. Chernoff. A Measure of Asymptotic Efficiency of Tests of a Hypothesis based on the Sum of Observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

A. Christmann and I. Steinwart. Universal Kernels on Non-standard Input Spaces. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

C. Cortes and V .N. Vapnik. Support Vector Networks. 20:273–297, 1995.

Nello Cristianini, Jaz Kandola, Andre Elisseeff, and John Shawe-Taylor. On kernel target alignment. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

Philip Derbeko, Ran El-Yaniv, and Ron Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *J. Artif. Int. Res.*, 22(1):117–142, October 2004. ISSN 1076-9757. URL `http://dl.acm.org/citation.cfm?id=1622487.1622492`.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2000.

V. Feldman, V. Guruswami, P. Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 385–394, 2009.

G. B. Folland. *Real Analysis: Modern Techniques and Their Applications*. Wiley, 2 edition, 1996.

Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004. ISSN 1532-4435.

T. Joachims. A Support Vector Method for Multivariate Performance Measures. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-612-2. URL `http://dl.acm.org/citation.cfm?id=645528.657646`.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47:1902–1914, 2001.

G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

M. Ledoux and M. Talagrand. *Probability in Banach Space*. Springer-Verlag, New York, 1991.

G. Lugosi and K. Zeger. Concept learning using complexity regularization. *IEEE Transactions on Information Theory*, 42(1):48–54, 1996.

C. McDiarmid. On the methods of bounded differences. *Surveys in Combinatorics*, pages 148–188, 1989.

Ron Meir and Tong Zhang. Generalization Error Bounds for Bayesian Mixture Algorithms. *Journal of Machine Learning Research*, 4:839–860, 2003.

Bojan Mohar. Some Applications of Laplace Eigenvalues of Graphs. *GRAPH SYMMETRY: ALGEBRAIC METHODS AND APPLICATIONS*, 497, 1997.

J. Saketha Nath. Lecture Notes of cs723. `http://www.cse.iitb.ac.in/saketh/teaching/cs723.html`, 2009.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 775–782, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: http://doi.acm.org/10.1145/1273496.1273594.

Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill, 3rd edition, 1976.

Saketh. Lecture Notes for CS723. Available at `http://www.cse.iitb.ac.in/saketh/teaching/cs723.html`, 2009.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT press, Cambridge, 2002.

Hyunjung Shin, Koji Tsuda, and Bernhard Schlkopf. Protein functional class prediction with a combined graph. *Expert Systems with Applications: An International Journal archive*, 36(2), 2009.

V. Sindhwani. On Semi-supervised Kernel Methods. Doctoral Thesis , University of Chicago, 2007.

Fabian Sinze, Olivier Chapelle, Alekh Agarwal, and Bernhard Schlkopf. An Analysis of Inference with the Universum. In *Advances in Neural Information Processing Systems*, 2007.

A.J. Smola and R. Kondor. Kernels and regularization on graphs. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the Annual Conference on Computational Learning Theory and Kernel Workshop*, Lecture Notes in Computer Science. Springer, 2003.

Ingo Steinwart. *Journal of Machine Learning Research*.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JOURNAL OF MACHINE LEARNING RESEARCH*, 6:1453–1484, 2005.

V. Vapnik and A. Chervonenkis. The necessary and sufficient conditions for consistency in the Empirical Risk Minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998.

Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, and Vladimir Vapnik. Inference with the universum. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 1009–1016, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143971. URL http://doi.acm.org/10.1145/1143844.1143971.

D Yarowsky. Unsupervised word sense disambiguation rivaling super-vised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

Xiaojin Zhu, Jaz Kandola, John Lafferty, and Zoubin Ghahramani. Graph Kernels by Spectral Transforms. Book chapter in Semi-Supervised Learning, MIT Press, 2006.

Appendix - 1 (SRM Consistency)

We will prove consistency of SRM in the following case:

(i) $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_n \subset \ldots$ is the structure over the function classes. Thus the effective reach is over $\bigcup_{i=1}^{\infty} \mathcal{F}_i$.

(ii) each of $\mathcal{F}_i$ is a "good" class, in the sense that the conditional Rademacher average decays with m i.e. $\hat{\mathcal{R}}(\mathcal{F}_i) \to 0$ as $m \to \infty$.

(iii) $\mathcal{F}_i = \{ f \mid f(x) = w^T x, \|w\| \leq i \}$

This collection of function classes satisfies above two criteria as:

$\hat{\mathcal{R}}(\mathcal{F}_i) = \frac{i\sqrt{n}}{\sqrt{m}} \to 0$ as $m \to \infty$. (∵ eqn. (2·8).)

(iv) f_m^{SRM} → denotes the SRM candidate with 'm' examples and is defined as

$$f_m^{SRM} = f_m^{ERM_{i^*}}$$

where $f_m^{ERM_i}$ denotes the ERM candidate with m examples in $\mathcal{F}_i$.

and $i^* = \underset{i}{\operatorname{argmin}} \; \tilde{R}\left[f_m^{ERM_i} \right]$

$\tilde{R}[f]$ for an $f \in \mathcal{F}_i$ is defined as $\hat{R}[f] + 2\hat{\mathcal{R}}_i(\mathcal{F}_i) + \frac{\sqrt{i}}{\sqrt{m}}$,

which is the upper bound on the R[f]

$\downarrow = \frac{i\sqrt{n}}{\sqrt{m}}$

Motivation for $\tilde{R}[f]$ comes from (2.7). The extra term $\frac{\sqrt{i}}{\sqrt{m}}$ is added to get ~~nice convergence~~ SRM consistency. Basically this term 'over penalizes' bigger function classes than said by (2.7).

Note: In traditional SRM, we consider $\tilde{R}[f] = \hat{R}[f] + \hat{R}(\mathcal{F}_i)$; so please note the additional term $\frac{\sqrt{i}}{\sqrt{m}}$.

Proof:

TST: SRM is consistent

i.e. TST: $\left\{ R[f_m^{SRM}] \right\} \xrightarrow{P} R[f^*]$ (as $m \to \infty$)

where $R[f^*] = \min_i R[f^{*i}]$ and $f^{*i} = \underset{f \in \mathcal{F}_i}{\operatorname{argmin}} R[f]$

i.e. TST: $P[\, |R[f_m^{SRM}] - R[f^*]| > \varepsilon \,] \to 0$ (as $m \to \infty$)

Lets start with upper bounding $P[R[f_m^{SRM}] - R[f^*] > \varepsilon]$. The other way bound is similar.

$$P[R[f_m^{SRM}] - R[f^*] > \varepsilon] = P[R[f_m^{SRM}] - \tilde{R}[f_m^{SRM}] + \tilde{R}[f_m^{SRM}] - R[f^*] > \varepsilon]$$

easy to prove by writing $P(e) = E[1_e]$ for an event e. $\leq P[R[f_m^{SRM}] - \tilde{R}[f_m^{SRM}] > \varepsilon/2] +$ (I)

$P[\tilde{R}[f_m^{SRM}] - R[f^*] > \varepsilon/2]$ (II)

$$\text{(I)} \leq P\left[\max_i \left\{R[f_m^{ERM_i}] - \tilde{R}[f_m^{ERM_i}]\right\} > \varepsilon/2\right] \quad \left(\because \text{SRM candidate is one of the ERM}_i \text{ candidate}\right)$$

$$= P\left[\bigcup_{i=1}^{\infty} \left\{R[f_m^{ERM_i}] - \tilde{R}[f_m^{ERM_i}] > \varepsilon/2\right\}\right]$$

$$\leq \sum_{i=1}^{\infty} P\left[R[f_m^{ERM_i}] - \tilde{R}[f_m^{ERM_i}] > \varepsilon/2\right] \quad (\because \text{Boole's inequality})$$

$$= \sum_{i=1}^{\infty} P\left[R[f_m^{ERM_i}] - \hat{R}[f_m^{ERM_i}] - \frac{2i\eta}{\sqrt{m}} - \frac{\sqrt{i}}{\sqrt{m}} > \varepsilon/2\right] \quad (\because \text{defn of } \tilde{R})$$

$$= \sum_{i=1}^{\infty} P\left[R[f_m^{ERM_i}] - \hat{R}[f_m^{ERM_i}] - \frac{2i\eta}{\sqrt{m}} > \frac{\varepsilon}{2} + \sqrt{\frac{i}{m}}\right]$$

$$\leq \sum_{i=1}^{\infty} 2e^{-\frac{2m}{9}\left(\frac{\varepsilon}{2}+\sqrt{\frac{i}{m}}\right)^2} \quad (\because \text{by eqn. (2.7) \& (2.8)})$$

$$\leq 2e^{-\frac{m\varepsilon^2}{18}} \sum_{i=1}^{\infty} e^{-2i/9} \quad \left(\because -(a+b)^2 \leq -(a^2+b^2) \text{ for } a,b \geq 0\right)$$

$$= \frac{2e^{-2/9}}{1-e^{-2/9}} e^{-\frac{m\varepsilon^2}{18}} \quad (\because \text{GP sum})$$

Now, we wish to upper bound (II). Towards this end we require a lower bound on $R[f^*]$ involving $R[f^{*j}]$ for some j. This is possible by realizing that $\{R[f^{*i}]\} \longrightarrow R[f^*]$ (as $i \to \infty$) and is infact a non-increasing sequence. By monotone convergence theorem we get that $\exists\, j \ni R[f^*] \leq R[f^{*j}] + \varepsilon/4$. Note that given $\varepsilon > 0$, we can fix this j.

Using this inequality in (II) gives:

$$(II) \leq P\left[\tilde{R}\left[f_m^{SRM}\right] - R\left[f^{*j}\right] > \varepsilon/4\right]$$

$$\leq P\left[\tilde{R}\left[f_m^{ERM_j}\right] - R\left[f^{*j}\right] > \varepsilon/4\right] \quad (\because \text{by SRM defn.})$$

$$= P\left[\hat{R}\left[f_m^{ERM_j}\right] + \frac{2j\eta}{\sqrt{m}} - R\left[f^{*j}\right] > \frac{\varepsilon}{4} - \sqrt{\frac{j}{m}}\right]$$

$$\leq P\left[\hat{R}\left[f^{*j}\right] + \frac{2j\eta}{\sqrt{m}} - R\left[f^{*j}\right] > \frac{\varepsilon}{4} - \sqrt{\frac{j}{m}}\right] \quad (\because \text{by ERM defn.})$$

(Now given ε & hence j, one can choose $m_0 = \frac{64j}{\varepsilon^2}$ so that $\sqrt{\frac{j}{m}} \leq \frac{\varepsilon}{8}$)

$$\leq P\left[\hat{R}\left[f^{*j}\right] + \frac{2j\eta}{\sqrt{m}} - R\left[f^{*j}\right] > \varepsilon/8\right]$$

$$\leq 2e^{-\frac{2m}{9}\left(\varepsilon/8\right)^2} = 2e^{-\frac{m\varepsilon^2}{288}} \quad (\because \text{by eqn. 111th to other way round (2.7) \& (2.8)})$$

The bound in the other direction is also same. Hence:

$$P\left[\left|R\left[f_m^{SRM}\right] - R\left[f^*\right]\right| > \varepsilon\right] \leq \frac{4e^{-2/9}}{1-e^{-2/9}} e^{-m\varepsilon^2/18} + 4e^{-m\varepsilon^2/288}$$

$$\longrightarrow 0 \quad \text{as } m \to \infty$$

Hence SRM is consistent
(the modified version)

Semi-Supervised Learning

28-Feb-12

Motivation

- Supervised Learning $\Rightarrow$ more the training examples the better
- Creating large training data is difficult:
  - Speech recognition
  - Relevance feedback
  - Medical diagnosis
  - Sentiment analysis
- However un-labeled data is easy to obtain (in above cases)

Motivation

- Supervised Learning $\Rightarrow$ more the training examples the better
- Creating large training data is difficult:
  - Speech recognition
  - Relevance feedback
  - Medical diagnosis
  - Sentiment analysis
- However un-labeled data is easy to obtain (in above cases)

Can un-labeled data boost performance?

Can un-labeled data help?

Inductive Learning:

- Association of pictures with words in babies
  - modeling $p(x)$ helps in better modeling $p(y/x)$? (any assumptions?)

Can un-labeled data help?

Inductive Learning:

- Association of pictures with words in babies
  - modeling $p(x)$ helps in better modeling $p(y/x)$? (any assumptions?)

Transductive Learning:

- Un-labelled data is the only test data (applications;domain adaptation?)
  - better learning bounds; use to learn structure (SRM)?

Can un-labeled data help?

Inductive Learning:

- Association of pictures with words in babies
  - modeling $p(x)$ helps in better modeling $p(y/x)$? (any assumptions?)

Transductive Learning:

- Un-labelled data is the only test data (applications;domain adaptation?)
  - better learning bounds; use to learn structure (SRM)?

Learning with universum:

- Training: red-ish objects, blue-ish objects; Un-labelled: magenta (availability?)
  - Avoid un-necessary generalization; use to learn structure (SRM)?

Can un-labeled data help?

Inductive Learning:

- Association of pictures with words in babies
  - modeling $p(x)$ helps in better modeling $p(y/x)$? (any assumptions?)

Transductive Learning:

- Un-labelled data is the only test data (applications;domain adaptation?)
  - better learning bounds; use to learn structure (SRM)?

Learning with universum:

- Training: red-ish objects, blue-ish objects; Un-labelled: magenta (availability?)
  - Avoid un-necessary generalization; use to learn structure (SRM)?

Perhaps there are more ways!

Semi-supervised Learning

Given:

- Labeled set: $\mathcal{D}^l = \{(x_i, y_i) \mid x_i \in \mathcal{X}, y_i \in \mathcal{Y}\, i = 1, \ldots, m_l\}$
- Un-labeled set: $\mathcal{D}^u = \{z_i \mid z_i \in \mathcal{X}, i = 1, \ldots, m_u\}$

Semi-supervised Learning

Given:

- Labeled set: $\mathcal{D}^l = \{(x_i, y_i) \mid x_i \in \mathcal{X}, y_i \in \mathcal{Y}\, i = 1, \ldots, m_l\}$
- Un-labeled set: $\mathcal{D}^u = \{z_i \mid z_i \in \mathcal{X}, i = 1, \ldots, m_u\}$

Inductive:

Choose $\mathcal{F}$ and $f \in \mathcal{F}$ for which $R[f]$ is minimum

Transduction:

Choose $\mathcal{F}$ and $f \in \mathcal{F}$ for which "risk of employing f on $\mathcal{D}^u$" is minimum

(define formally later!)

Induction vs. Transduction [chp. 25, Chapelle et.al., 2006]

Induction	**Transduction**
Any transduction algo works	Any induction algo works
Theoretical investigation complex	Theoretical investigation simple (basic step?)
Loose bounds	Tighter bounds
Seems to benefit as $u \longrightarrow \infty$	Seems tougher as $u \longrightarrow \infty$ (approaches induction)
Seems to make assumptions	Seems to make no assumptions (but most algo do!)
Prediction not dependent on testset	Prediction depends on testset (domain adaptation?)