Topics in Machine Learning (CS729)

Instructor: Saketh

Contents

Contents 1

1 Introduction 3

2 Supervised Inductive Learning 5

2.1 Statistical Learning Theory (for SIL case) 6

2.1.1 ERM Consistency — Finite $\mathcal{F}$ case 8

2.1.2 ERM Consistency — General $\mathcal{F}$ case 10

2.1.3 Example of function/loss class with ERM consistency — Linear functions 12

2.1.4 Other Examples 14

2.2 Support Vector Machines (SVMs) 15

2.3 Model Selection Problem 17

2.3.1 SRM consistency 19

2.4 Non-linear Function-classes 20

2.4.1 Kernels and Kernel-trick 21

2.4.2 Universal Kernels 25

2.5 Bayes Consistency 26

2.6 Operator-valued Kernels 26

2.7 Kernel/Feature Learning 28

2.7.1 Hyperkernels 30

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 (25%)
- Kernel Methods (40%)

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

- Kernel Learning

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 $f = \operatorname{argmin}_{f\in\mathcal{F}}\mathbb{E}[1_{f(X)\neq Y}] = \operatorname{argmin}_{f\in\mathcal{F}}P[f(X) \neq Y]$.

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: $\mathcal{F}^l_W = \{f \mid f(x) = w^\top x,\ \|w\| \leq W\}$. 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: $\mathcal{F}^{ld} = \{f \mid f(x) = \text{sign}(w^\top x)\}$. 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 datapoints. Clearly the empirical risk with f is zero and the ERM picks it. With 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

[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.

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

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: $P\left[\max_{f\in\mathcal{F}}\left(R[f]-\hat{R}_m[f]\right)>\right.$ $\left.\sum_{f\in\mathcal{F}} P\left[R[f]-\hat{R}_m[f]>\epsilon\right]\right.$. 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$\leq e^{-s\epsilon}\mathbb{E}[e^{s\left(R[f]-\hat{R}_m[f]\right)}]$
- Use the fact that the random variables[5] $L_1(f), L_2(f), \ldots, L_m(f)$ are independent (infact iid): LHS$\leq e^{-s\epsilon}\Pi_{i=1}^{m} E[e^{\frac{s}{m}(\mathbb{E}[L_i(f)]-L_i(f))}]$

[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 $|calF|=1$). Here we provide the version with the relevant random variables for the present context.

[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 $L_i(f)$.

- 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 $f_m^{SRM} = \text{argmin}_{f\in\mathcal{F}_{i^*}} \hat{R}_m[f]$.

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

[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.

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: $\mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i\left(l(Z_i', f) - l(Z_i, f)\right)\right] = \mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i$ $\mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} -\sigma_i l(Z_i, f)\right] = 2\mathbb{E}\left[\max_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i l(Z_i, f)\right]$. 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 $1 - \delta$:

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

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

[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.

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 began with the case of binary classification, linear discriminant function class and 0-1 loss. In this case we gave an intuition why/how the Rademacher complexity provides a measure for complexity of the function class. This intuition lead to the definition of VC-dimension Burges [1998], Vapnik [1998]: the maximum number of datapoints that can be shattered, i.e. given all possible labelings, using the function class. It was easy to see that the VC-dim.(denoted by h henceforth) is $d+1$ for the linear discriminant function class over d-dimensional input space. We then noted the Haussler-Dudley bound on empirical Rademacher complexity: $\hat{\mathcal{R}}_m(\mathcal{F}) \leq a\sqrt{\frac{h_\mathcal{F}}{m}}$ (for some $a > 0$). This gives us that the Rademacher complexity indeed decays to zero with m and hence ERM is consistent.

We noted two reasons why the above analysis is not attractive: i) the bound above is NOT independent of dimensionality of the input data. This seems restrictive because on one hand one might want to use as many features as possible for describing the data to improve learning (say, empirical risk), however, it seems that the complexity term increases though. This is usually referred to as the curse of dimensionality. In the subsequent paragraphs we present a function class with no curse of dimensionality and is essentially linear. ii) the 0-1 loss is not attractive for two reasons: a) in binary classification problems one may want a hold on the 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). b) 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 $l(x, y, f) = \phi(yf(x))$, where $\phi(z) = \min(\max(0, 1-z), 1)$

[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

(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.

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 $m \to \infty$.

---

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 $|f(x) - f(y)| \leq L\|x - y\| \ \forall \ x, y \in dom(f)$.

[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.

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.

On passing we made an important comment that the function class we started with had no "curse of dimensionality", as the expression for guaranteed risk is (not very loosely) upper-bounded by an expression independent of the input space dimensionality. We also commented that, in early 1990s (time of birth of SVMs), such non-cursed set of linear functions were not known[20].

2.1.4 Other Examples

We looked at the 1-norm constrained function class: $\mathcal{F}_W^1 = \{f \mid f(x) = w^\top x, \|w\|_1 \leq W\}$. The Rademacher bound derived above can be derived in this case too; just by replacing the Cauchy-Schwartz Inequality by the Holder's Inequality. This would lead to the bound: $\frac{W}{m}\mathbb{E}\left[\|\sum_{i=1}^m \sigma_i y_i x_i\|_\infty\right]$. However, since always $\|z\|_\infty \leq \|z\|_2$, we obtain exactly the same radius-margin bound as above (which has no curse of dimensionality).

We then looked at: $\mathcal{F}_W^\infty = \{f \mid f(x) = w^\top x, \|w\|_\infty \leq W\}$. In this case, the Holder's inequality would give the bound: $\frac{W}{m}\mathbb{E}\left[\|\sum_{i=1}^m \sigma_i y_i x_i\|_1\right]$. We finally obtain the bound $\frac{Wr\sqrt{d}}{\sqrt{m}}$, because $\|z\|_1 \leq \sqrt{d}\|z\|_2$. We commented that there seems to be a curse of dimensionality for this case.

Infact, one can easily generalize to function class with a generic norm bound. It is easy to see that one would obtain its dual-norm Saketh [2012] in the Rademacher bound. Actually, one can even start with a function class with some convex function of w being bounded, as long as its support function Saketh [2012] is bounded. In the next section we will write down the ERM problems with some of these function class as optimization problems.

[20]Recall that the VC-dim of set of all linear classifiers is $d+1$, where d is dimensionality of the input space; and is indeed NOT independent of dimensionality.

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[21], whereas the truncated hinge-loss 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)[22], 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"

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

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

(which we have shown is equivalent to ERM). Here regularized risk refers to the weighted sum of the regularizer and empirical risk.

Morozov regularization:

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

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):

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

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:

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

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[23]. 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"

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

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.

We ended this discussion by writing down the optimization problems corresponding to various loss functions and functions classes: $\mathcal{F}_W^l$ with square-loss is known as ridge-regression Hoerl and Kennard [1970] or regularized least-squares or min. norm least squares[24]. $\mathcal{F}_W^l$ with square-hinge loss is referred to as l_2-SVM. $\mathcal{F}_W^l$ with ϵ-insensitive loss is called as Support Vector Regression Smola and Schölkopf [2004]. $\mathcal{F}_W^1$ with square-loss is called as LASSO Tibshirani [1996]. $\mathcal{F}_W^1$ with hinge-loss is called as l_1-regularized SVM etc.

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^{**} is called the Bayes (optimal) function[25]. 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 $\mathcal{F}$)

[24] Refer to the limit defn. of Pseudo-inverse.

[25] 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.

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[26]). The candidate for SRM is $f_m^{SRM} = \text{argmin}_{f \in \mathcal{F}_{i^*}} \hat{R}_m[f]$.

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

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

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.

Note that it is clear from the above discussion that validation or SRM with finite hierarchy does not actually solve the model selection problem as this can be repeated recursively ad inf. However they give a reasonable working heuristic. The actual model selection problem will be solved if be design a hierarchy that includes the Bayes optimal (for any problem) and then prove SRM is consistent. Since it is reasonable to expect that Bayes optimal need not lie in any finite-capacity function class, we prove SRM consistency with a sequence of function classes in the subsequent section. Later in other sections we explicitly show this "universal" hierarchy.

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[27] 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 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]).

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

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:

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

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[28], 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)[29].

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

[28] Refer lecture-notes 1-4 in Saketh [2010] 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.

[29] $M \succeq 0 \Leftrightarrow x^\top M x \geq 0 \ \forall \ x$. 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 [2002] (refer definition 2.5).

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[30].

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[31]. This is because a gram matrix can be written 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

[30] 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).

[31] 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 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.

$k_1(x,z) = (x^\top z)$ and $k_2(x,z) = (x^\top z)$ is again positive[32]. 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.

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

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

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

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. We ended the discussion with yet another example of a non-Euclidean kernel that is in the space of strings: Rational Kernels Cortes et al. [2004].

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[33]. 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[34]) is universal. The proof[35] 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[36]. ii) non-vanishing because for any $x \in \mathbb{R}^n$, we can take $f_z(x) = k(z, x) = e^{z^\top x} > 0,\ z \in \mathbb{R}^n$. iii) separates $\mathcal{X}$ because $x, y \in \mathbb{R}^n, x \neq y \Rightarrow \exists z \ni z^\top x < z^\top y$ (separation theorem) and hence $f_z(x) = e^{z^\top x} \neq e^{z^\top y} = f_z(y)$. 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.

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

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

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

[36] 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 present a discussion on operator-valued kernels (a generalization of the notion of kernels) that will enable us to perform structured prediction i.e., induce functions of the form $f : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{Y}$ need NOT be $\mathbb{R}$.

2.6 Operator-valued Kernels

Here we are concerned with the problem of learning functions of the form $f : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{Y}$ need NOT be $\mathbb{R}$. This setting is popularly known as "learning in structured output spaces". Examples: i) multi-task learning[37]: simultaneous prediction of n (mutiple) labels for a given example. Here $\mathcal{Y} = \mathbb{R}^n$. ii) Functional Regression: $\mathcal{X}$ as well as $\mathcal{Y}$ are some sets of functions. This situation commonly arises in weather prediction e.g., given temperature profiles, predict precipitation profiles. Refer Tsochantaridis et al. [2005] for more examples.

We wanted to generalize the notion of kernels to this case as this would then allow us to learn non-linear functions from $\mathcal{X}$ to $\mathcal{Y}$ using a simple SVM algorithm. Carrying forward from the standard case of $\mathcal{Y} = \mathbb{R}$, we let $\mathcal{H}$ be a Hilbert space

[37]Needless to say, multi-class classification is a special case of multi-task learning.

of functions $h : \mathcal{X} \mapsto \mathcal{Y}$. The first road-block we encountered was in putting down the form of the function class itself! Clearly, we cannot go with $f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$ (standard reproducing property) as $\mathcal{Y}$ need NOT be $\mathbb{R}$. Secondly, how would we measure loss? One simple way out for the second problem (that will later on provide answer for the first) is to assume we know how to measure deviations between labels. Formally, we assumed a Hilbert space over $\mathcal{Y}$. Given this, $\langle y_i, f(x_i) \rangle_{\mathcal{Y}}$ would give the match between the predicted label of x_i, which is $f(x_i)$, and the true one, y_i. Now, one can use hinge-loss or square loss or any other loss studied earlier. $l(x_i, y_i, f) = \Phi(\langle y_i, f(x_i) \rangle_{\mathcal{Y}})$, where Φ is hinge loss function etc. This also prompted us to explore the possibility of generalizing the standard reproducing property by comparing two inner-products (in the $\mathcal{Y}$ space and Hilbert space).

In order to get an idea of how this generalization will look like we took the standard case and multiplied both sides by y: $yf(x) = \langle f, y\phi(x) \rangle_H$. With this our generalization (guess[38]) of reproducing property is: $\langle y, f(x) \rangle_{\mathcal{Y}} = \langle f, \phi(x, y) \rangle_H$, where $\phi : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{H}$ is a function linear wrt. y. The next step was to introduce the notion of kernel, for which we repeated the exercise we did in case of $\mathcal{Y} = \mathbb{R}$ of writing ERM problem and then investigating a representer theorem. Here, the ERM problem is:

$$\min_{w \in \mathcal{H}} \quad \frac{1}{2}\|w\|_{\mathcal{H}}^2 + C \sum_{i=1}^{m} \Phi(\langle y_i, f(x_i) \rangle_{\mathcal{Y}})$$

This is same as:

$$\min_{w \in \mathcal{H}} \quad \frac{1}{2}\|w\|_{\mathcal{H}}^2 + C \sum_{i=1}^{m} \Phi(\langle f, \phi(x_i, y_i) \rangle_{\mathcal{H}})$$

Now going through the steps of proof of the standard representer theorem gives: at optimality $w = \sum_{i=1}^{m} \alpha_i \phi(x_i, y_i)$ for some αs. Again, as earlier, we do not need w explicitly for prediction; what we need is $w(x) = \sum_{i=1}^{m} \alpha_i \phi(x_i, y_i)(x)$. This expression gave us the form of the generalized kernel: $k : \mathcal{X} \times \mathcal{X} \mapsto L(\mathcal{Y})$ and $k(x_i, x_j) \equiv \phi(x_i, \cdot)(x_j)$. Here, $L(\mathcal{Y})$ is the space of linear operators on $\mathcal{Y}$ i.e., $l \in L(\mathcal{Y}) \Leftrightarrow l : \mathcal{Y} \mapsto \mathcal{Y}$ and l is a linear function.

By taking the example of $\mathcal{Y} = \mathbb{R}^n$ ($L(\mathcal{Y}) = \mathbb{R}^{n \times n}$), we gave intuitive explanations for this (generalized) kernel. The kernel value $k(x_i, x_j)$, which is a matrix, tells how correlated the labels to be predicted are for the given pair of examples. Moreover, by representer theorem, $w(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x)(y_i)$ i.e., the label of x is a weighted linear combination of labels of the training examples. In this sense too, the notion of kernel is completely analogous to the $\mathcal{Y} = \mathbb{R}$ case.

Once the form of reproducing property and kernel are realized, it is easy to give the definition and characterization of these, "operator-valued kernels". Please

[38]Though we present it here as an intuitive guess, this infact is the statement of Riesz representer theorem and the correct way to generalize the reproducing property.

refer proposition 2.1 and theorem 2.1 in Micchelli1 and Pontil [2005]. Note that the properties (b)-(c) in prop. 2.1 define a kernel and are analogous to the conditions of being symmetric psd in standard case. We ended the discussion by noting examples of kernels (given on pages 4,5 of Micchelli1 and Pontil [2005]) and some universal kernels Caponnetto et al. [2008].

2.7 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 i) multi-modal data[39], where each base kernel is constructed from a different mode of describing the data. ii) non-linear feature selection. Obviously, one would like to promote non-sparse combinations for i) and sparse ones for ii).

Let $\mathcal{H}_i, \phi_i$ be the RKHS, feature map with the kernel k_i and let $\mathcal{H}_i, \hat{\phi}_i$ 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., $\mathcal{H} = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_p$. Hence, inner product $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{p} \langle f_i, g_i \rangle_{\mathcal{H}_i}$ (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: $f(x) = \langle w, \hat{\phi}(x) \rangle_{\mathcal{H}} = \sum_{i=1}^{p} \langle w_i, \hat{\phi}(x)_i \rangle_{\mathcal{H}_i} = \sum_{i=1}^{p} \sqrt{\lambda_i} \langle w_i, \phi_i(x) \rangle_{\mathcal{H}_i}$.

From the risk bounds (2.7,2.8) it follows that the capacity of the induced function class is bounded as long as $\|w\|_{\mathcal{H}}^2 = \sum_{i=1}^{p} \|w_i\|_{\mathcal{H}_i}^2 \leq W$ and $trace(\sum_{i=1}^{p} \lambda_i K_i) =$

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

$\sum_{i=1}^{p} \lambda_i trace(K_i) \leq T$ for some W and T. Now, one can write the ERM problem as:

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

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[40] (2.20) as:

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

This program is convex[41] as $\frac{\|w_i\|_{\mathcal{H}_i}^2}{\lambda_i}$ 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 looks for a kernel combination that gives a good trade-off for: (low) empirical risk, (large) margin and (low) radius/spread of data.

Firstly, since (2.21) involved l_1-norm regularization over λs, we expect to obtain a sparse solution. Infact using (10) in Rakotomamonjy et al. [2007], we eliminated λs from a re-parametrized Tikhonov version and rewrote (2.21) as:

$$\min_{w} \quad \frac{1}{2} \left(\sum_{i=1}^{p} \|w_i\|_{\mathcal{H}_i} \right)^2 + C \sum_{i=1}^{m} l(y_i \sum_{i=1}^{p} \langle w_i, \phi_i(x) \rangle_{\mathcal{H}_i}). \tag{2.22}$$

This clearly gives the connection with LASSO: the regularizer in (2.22) is simply a 1-norm over a vector with entries as $\|w_i\|_{\mathcal{H}_i}$. Hence at optimality many $w_i = 0$ i.e., we are performing a sparse combination of base kernels. We also noted the 2-norm version of (2.22) is the usual SVM with the kernel as $k_1 + \ldots + k_p$. We later derived formulations for various other norms Nath et al. [2009], Kloft et al. [2009].

We then went ahead and tried to observe closely why Lasso promotes sparsity. We gave an explanation using optimality conditions. Secondly, we wrote the (conic) dual of (2.22) and realized that infact there is a solution where only one kernel of the base kernels is active (refer theorem 17 in Lanckriet et al. [2004]). Once convinced

[40] Here, we know at optimality $\lambda_i = 0 \Rightarrow w_i = 0, \hat{w}_i = 0$. Hence, for the function $\frac{\|w_i\|_{\mathcal{H}_i}^2}{\lambda_i}$ we define $\frac{0}{0} = 0$.

[41] Provided the loss is a convex function.

that Lasso does promote sparsity, we noted a theorem that answers the question whether the sparsity achieved by Lasso is always the right one? Refer theorems 2,3 in Bach [2008]. In the subsequent section, we present a methodology that enables us to learn non-linear combinations of the given base kernels.

2.7.1 Hyperkernels

This section provides a brief summary of the methodology proposed in Ong et al. [2005] for kernel learning.

In the previous section we looked at linear combinations of kernels. One way to generalize this and look for non-linear combinations is ofcourse use the known trick of searching in an appropriate Hilbert space of standard kernels. i.e., we aim to study kernels whose RKHS itself should include/be the space of standard kernels. Such kernels we will call as hyperkernels, denoted by $\underline{k}$. Now we must define hyperkernels. To this end first lets look at $\mathcal{H}_{\underline{k}}$. We want that every $h \in \mathcal{H}_{\underline{k}}$ to be a standard kernel i.e., $h : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ and h is a valid kernel[42]. First of all this is not going to work since set of kernels forms a cone and hence every element of Hilbert space cannot be a kernel. So we will be happy by insisting that the Hilbert space has "many" kernels (may not all be kernels).

From the form of $\mathcal{H}_{\underline{k}}$ it is clear that $\underline{k} : \underline{\mathcal{X}} \times \underline{\mathcal{X}} \mapsto \mathbb{R}$, where $\underline{\mathcal{X}} \equiv \mathcal{X} \times \mathcal{X}$. The obvious conditions for $\underline{k}$ being a kernel are: i) symmetry: $\underline{k}\left((x_i, z_i), (x_j, z_j)\right) = \underline{k}\left((x_j, z_j), (x_i, z_i)\right)$. ii) pos.def.: $\alpha^\top G_{\underline{k}} \alpha \geq 0 \; \forall \; \alpha$, where $G_{\underline{k}}$ is a gram matrix with $\underline{k}$. As mentioned earlier, we want many elements of $\mathcal{H}_{\underline{k}}$ themselves as kernels. TO this end, we put this additional constraint that iii) the typical element $\underline{k}\left((x, z), (\cdot, \cdot)\right)$ (which is $\phi_{\underline{k}}(x, z)$) is itself a kernel from $\mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Note that (iii) ensures that the conic hull of all images of $\underline{\mathcal{X}}$ in $\mathcal{H}_{\underline{k}}$ under the map $\phi_{\underline{k}}$ are valid kernels over $\mathcal{X}$. In summary, (i),(ii),(iii) define a hyperkernel.

We then went on to write down the ERM problem:

$$\begin{aligned} \min_{k \in \mathcal{H}_{\underline{k}}} \; \min_{w \in \mathcal{H}_k} \quad & C_2 \textstyle\sum_{i=1}^m l(y_i \langle w, \phi_k(x_i) \rangle_{\mathcal{H}_k}), \\ \text{s.t.} \quad & \|w\|_{\mathcal{H}_k} \leq W_1, \; \|k\|_{\mathcal{H}_{\underline{k}}} \leq W_2 \end{aligned}$$

which can be re-parameterized in Tikhonov form as:

$$\min_{k \in \mathcal{H}_{\underline{k}}} \; \min_{w \in \mathcal{H}_k} \quad \frac{1}{2}\|w\|^2_{\mathcal{H}_k} + \frac{C_1}{2}\|k\|^2_{\mathcal{H}_{\underline{k}}} + C_2 \sum_{i=1}^m l(y_i \langle w, \phi_k(x_i) \rangle_{\mathcal{H}_k}) \tag{2.23}$$

[42]Only for notational convenience we restrict ourselves to scalar-valued kernels. However, the entire discussion can be generalized to operator-valued kernels.

It was then easy to prove the representer theorem (refer lemma7 in Ong et al. [2005]). This enabled us to write down the ERM problem as a convex program[43] in Euclidean space (refer (26) in Ong et al. [2005]). We then motivated and presented examples of hyperkernels (refer section 4 in Ong et al. [2005]). In particular, we noted the choices of hyperkernels that provide the effect of non-linearly (and linearly) combining given base kernels.

[43]The program is convex (and the kernel learnt is valid kernel) if we restrict $\beta \geq 0$. Hence one can only approximately solve this ERM problem, but efficiently.

Bibliography

F. Bach. Exploring Large Feature Spaces with Hierarchical Multiple Kernel Learning. In Advances in Neural Information Processing Systems (NIPS), 2008.

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

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.

Andrea Caponnetto, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. Universal Multi-Task Kernels. 9:1615–1646, 2008.

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.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational Kernels: Theory and Algorithms. Journal of Machine Learning Research, 5:1035–1062, 2004.

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.

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 42(1):80–86, 1970.

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

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskow, K-R. Mueller, and A. Zien. Efficient and Accurate Lp-Norm MKL. In Advances in Neural Information Processing Systems, pages 997–1005, 2009.

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.

Charles A. Micchelli1 and Massimiliano Pontil. On learning vector-valued functions. Neural Computation, 17:177–204, 2005.

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

J. Saketha Nath, G Dinesh, S Raman, Chiranjib Bhattacharyya, Aharon Ben-Tal, and Ramakrishnan K.R. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In Advances in Neural Information Processing Systems 22, pages 844–852, 2009.

Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson. Learning the Kernel with Hyperkernels. Journal of Machine Learning Research, 6:1043–1071, 2005.

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, 2010.

Saketh. Lecture Notes for CS709. Available at http://www.cse.iitb.ac.in/saketh/teaching/cs709.html, 2012.

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

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Statistics and Computing, 14(3):199–222, 2004.

Ingo Steinwart. Journal of Machine Learning Research.

R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B, 58(1):267–288, 1996.

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.