CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 1**

Overview plus probabilistic formulations of prediction problems

Lecturer: Peter Bartlett *Scribe: Peter Bartlett*

1 Organizational issues

The course web page is at http://www.cs.berkeley.edu/∼bartlett/courses/281b-sp08/
See the web page for details of office hours, the syllabus, assignments, readings, lecture notes, and announcements.

1.1 Assignments

There will be roughly five homework assignments, approximately one every two weeks. The first has been posted on the web site. It is due at the lecture on Thursday, January 31. You will also need to act as scribe for a small number of lectures, preparing a latex version of lecture notes. There is a template on the web site, and the latex file of the lecture notes for this lecture. (Please email the GSI, David, to choose the lecture that you'd like to prepare lecture notes for.) Also, there will be a final project, in an area related to the topics of the course.

2 Overview

The course will focus on the theoretical analysis of prediction methods.

1. Probabilistic formulation of prediction problems
2. Algorithms:
   (a) Kernel methods
   (b) Boosting algorithms
3. Risk bounds
4. Game theoretic formulation of prediction problems
5. Model selection

3 Probabilistic Formulations of Prediction Problems

In a prediction problem, we wish to predict an outcome y from some set $\mathcal{Y}$ of possible outcomes, on the basis of some observation x from a feature space $\mathcal{X}$. Some examples:

x	y
phylogenetic profile of a gene (i.e., relationship to genomes of other species)	gene function
gene expression levels of a tissue sample	patient disease state
image of a signature on a check	identity of the writer
email message	spam or ham

For such problems, we might have access to a data set of n pairs, $(x_1, y_1), \ldots, (x_n, y_n)$, and we would like to use the data to produce a function $f : \mathcal{X} \to \mathcal{Y}$ so that, for subsequent (x, y) pairs, $f(x)$ is a good prediction of y.

To define the notion of a 'good prediction,' we can define a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, so that $\ell(\hat{y}, y)$ quantifies the cost of predicting $\hat{y}$ when the true outcome is y. Then the aim is to ensure that $\ell(f(x), y)$ is small. For instance, in *pattern classification* problems, the aim is to classify an x into one of a finite number of classes (that is, the label space $\mathcal{Y}$ is finite). If all mistakes are equally bad, we could define

$$\ell(\hat{y}, y) = 1[\hat{y} \neq y] = \begin{cases} 1 & \text{if } \hat{y} \neq y, \\ 0 & \text{otherwise.} \end{cases}$$

As another example, in a *regression* problem, with $\mathcal{Y} = \mathbb{R}$, we might choose the quadratic loss function, $\ell(\hat{y}, y) = (\hat{y} - y)^2$.

We can formulate such problems using probabilistic assumptions: we assume that there is a probability distribution P on $\mathcal{X} \times \mathcal{Y}$, and that the pairs $(X_1, Y_1), \ldots, (X_n, Y_n), (X, Y)$ are chosen independently according to P. The aim is to choose f so that the *risk* of f,

$$R(f) = \mathbb{E}\ell(f(X), Y),$$

is small. For instance, in the pattern classification example, this is the misclassification probability.

$$R(f) = \mathbb{E}1[f(X) \neq Y] = \Pr(f(X) \neq Y).$$

Some things to notice:

1. We are using capital letters to denote random variables.
2. The distribution P can be viewed as modelling both the relative frequency of different features or covariates X, together with the conditional distribution of the outcome Y given X.
3. The assumption that the data is i.i.d. is a strong one.
4. The function $x \mapsto f_n(x) = f_n(x; X_1, Y_1, \ldots, X_n, Y_n)$ is random, since it depends on the random (X_i, Y_i). Thus, the risk

$$\begin{aligned} R(f_n) &= \mathbb{E}\left[\ell(f_n(X), Y) | X_1, Y_1, \ldots, X_n, Y_n\right] \\ &= \mathbb{E}\left[\ell(f_n(X; X_1, Y_1, \ldots, X_n, Y_n), Y) | X_1, Y_1, \ldots, X_n, Y_n\right] \end{aligned}$$

   is a random variable. We might aim for $\mathbb{E}R(f_n)$ small, or $R(f_n)$ small with high probability (over the training data).

We might choose f_n from some class F of functions, for instance, by choosing the structure and parameters of a decision tree, or by choosing the parameters of a neural net or a kernel machine.

There are several questions that we are interested in:

1. Can we design algorithms for which f_n is close to the best that we could hope for, given that it was chosen from F? (that is, is $R(f_n) - \inf_{f \in F} R(f)$ small?)
2. How does the performance of f_n depend on n? On other parameters of the problem?
3. Can we ensure that $R(f_n)$ approaches the best possible performance (that is, the infimum over all f of $R(f)$)?
4. What do we need to assume about P? About F?

In this course, we are concerned with results that apply to large classes of distributions P, such as the set of *all* joint distributions on $\mathcal{X} \times \mathcal{Y}$. In contrast to parametric problems, we will not (often) assume that P comes from a small (e.g., finite-dimensional) space, $P \in \{P_\theta : \theta \in \Theta\}$.

Several key issues arise in designing a prediction method for these problems:

Approximation How good is the best f in the class F that we are using? That is, how close to $\inf_f R(f)$ is $\inf_{f \in F} R(f)$?

Estimation Since we only have access to the distribution P through observing a finite data set, how close is our performance to that of the best f in F?

Computation We need to use the data to choose f_n, typically through solving some kind of optimization problem. How can we do that efficiently?

In this course, we will not spend much time on the approximation properties, beyond observing some universality results (that particular classes can achieve zero approximation error). We will focus on the estimation issue. We will take the approach that efficiency of computation is a constraint. Indeed, the methods that we spend most of our time studying involve convex optimization problems. (For example, kernel methods involve solving a quadratic program, and boosting algorithms involve minimizing a convex criterion in a convex set.)

4 The Probabilistic Formulation of Pattern Classification Problems

Assume, for simplicity, that $\mathcal{Y} = \{\pm 1\}$ (We'll consider extensions of the results of this lecture to the multiclass case in a homework problem.) Let's fix some notation: We'll represent the joint distribution P on $\mathcal{X} \times \mathcal{Y}$ as the pair (μ, η), where μ is the marginal distribution on $\mathcal{X}$ and η is the conditional expectation of Y given X,

$$\eta(x) = \mathbb{E}(Y|X = x) = P(Y = 1|X = x).$$

If we knew η, we could use it to find a decision function that minimized risk. To see this, notice that we can write the expected loss as an expectation of a conditional expectation,

$$\begin{aligned}
R(f) &= \mathbb{E}\ell(f(X), Y) \\
&= \mathbb{E}\mathbb{E}[\ell(f(X), Y)|X] \\
&= \mathbb{E}\left(\ell(f(X), 1)P(Y = 1|X) + \ell(f(X), -1)P(Y = -1|X)\right) \\
&= \mathbb{E}\left(1[f(X) \neq 1)\eta(X) + 1[f(X) \neq -1)(1 - \eta(X))\right) \\
&= \mathbb{E}\left(1[f(X) \neq 1)\eta(X) + (1 - 1[f(X) \neq 1])(1 - \eta(X))\right) \\
&= \mathbb{E}\left(1[f(X) \neq 1)(2\eta(X) - 1) + 1 - \eta(X)\right). \qquad (1)
\end{aligned}$$

Clearly, this expectation is minimized by choosing $f(x) = 1$ when $\eta(x) > 1/2$ and $f(x) = -1$ when $\eta(x) < 1/2$. Obviously, if $\eta(x) = 1/2$, the choice does not affect the risk. Let's define f^* as a function of this kind:

$$f^*(x) = \begin{cases} 1 & \text{if } \eta(x) \geq 1/2, \\ -1 & \text{otherwise.} \end{cases}$$

Denote the optimal risk (the *Bayes risk*), by $R^* = \inf_f R(f)$. We have shown that f^* achieves the Bayes risk. It is called the *Bayes decision function*.

Notice that any choice for $f^*(x)$ is equally good when $\eta(x) = 1/2$, so there can be several Bayes decision functions.

The following theorem shows something a little stronger: that the amount by which the risk of any other decision function exceeds the Bayes risk can be quantified in terms of a certain distance from f^*. (Actually, it's not quite a distance, since differences between functions at an x with $\eta(x) = 1/2$ have no influence on the risk.)

Theorem 4.1. For any $f : \mathcal{X} \to \mathcal{Y}$,

$$R(f) - R(f^*) = \mathbb{E}\left(1[f(X) \neq f^*(X)]|2\eta(X) - 1|\right).$$

Proof. Using the identity (1), we have

$$R(f) - R(f^*) = \mathbb{E}\left(1[f(X) \neq 1] - 1[f^*(X) \neq 1]\right)(2\eta(X) - 1).$$

But

$$\begin{aligned} &\left(1[f(X) \neq 1] - 1[f^*(X) \neq 1]\right)(2\eta(X) - 1) \\ &= 1[f(X) \neq f^*(X)]\left(1[f(X) \neq 1] - 1[f^*(X) \neq 1]\right)(2\eta(X) - 1) \\ &= \begin{cases} 1[f(X) \neq f^*(X)](2\eta(X) - 1) & \text{if } 2\eta(X) - 1 \geq 0, \\ 1[f(X) \neq f^*(X)](-1)(2\eta(X) - 1) & \text{if } 2\eta(X) - 1 < 0. \end{cases} \\ &= 1[f(X) \neq f^*(X)]|2\eta(X) - 1|, \end{aligned}$$

where the second inequality used the definition of f^*. □

This suggests one family of approaches to the pattern classification problem, known as *plug-in* methods. The idea is to use the data to come up with an estimate $\hat{\eta}$ of η, and then use

$$f_{\hat{\eta}}(x) = \begin{cases} 1 & \text{if } \hat{\eta}(x) \geq 1/2, \\ -1 & \text{otherwise.} \end{cases}$$

In estimating η, what criterion should we aim to minimize? We can use the earlier result to show that if the $L_1(\mu)$ distance between $\hat{\eta}$ and η is small, that suffices to ensure that the risk of $f_{\hat{\eta}}$ is close to the Bayes risk.

Theorem 4.2. For any $\hat{\eta} : \mathcal{X} \to \mathbb{R}$,

$$R(f_{\hat{\eta}}) - R^* \leq 2\mathbb{E}\left|\eta(X) - \hat{\eta}(X)\right|.$$

Proof. The previous theorem shows that the excess risk of $f_{\hat{\eta}}$ can be written as

$$R(f_{\hat{\eta}}) - R^* = 2\mathbb{E}1[f_{\hat{\eta}}(X) \neq f^*(X)]|\eta(X) - 1/2|. \tag{2}$$

Now, if $f_{\hat{\eta}}(X) \neq f^*(X)$, then $\hat{\eta}(X)$ and $\eta(X)$ must lie on opposite sides of $1/2$, and so we can write

$$|\eta(X) - \hat{\eta}(X)| = |\eta(X) - 1/2| + |\hat{\eta}(X) - 1/2| \geq |\eta(X) - 1/2|.$$

Thus, when the indicator inside the random variable in (2) is 1, we have

$$1[f_{\hat{\eta}}(X) \neq f^*(X)]2|\eta(X) - 1/2| \leq 2|\eta(X) - \hat{\eta}(X)|$$

And this inequality is also true when the indicator is zero, since the right hand side is non-negative. Plugging this inequality into (2) gives the result. □

Another family of approaches to pattern classification problems is to fix a class F of functions that map from $\mathcal{X}$ to $\mathcal{Y}$ and choose f_n from F. Next lecture, as an introduction to kernel methods, we'll consider the class of linear threshold functions on $\mathcal{X} = \mathbb{R}^d$,

$$F = \left\{ x \mapsto \text{sign}(\theta' x) : \theta \in \mathbb{R}^d \right\}.$$

The decision boundaries are hyperplanes through the origin ($d-1$-dimensional subspaces), and the decision regions are half-spaces.

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 2**

Linear threshold functions and the perceptron algorithm

Lecturer: Peter Bartlett *Scribe: Julia Palacios*

We'll consider a general family of approaches to the pattern classification problem, known as *empirical risk minimization* methods. We fix a class F of functions that map from $\mathcal{X}$ to $\mathcal{Y} = \{\pm 1\}$, and the method is such that, for all $(x_1, y_1), \ldots, (x_n, y_n)$, there is an $f \in F$ such that

$$f_n(x; x_1, y_1, \ldots, x_n, y_n) = f(x)$$

where $f \in F$ minimizes the empirical risk,

$$\hat{R}(f) = \hat{\mathbb{E}}\ell(f(X), Y) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i).$$

In the case of the 0-1 loss, the empirical risk is the proportion of training data that are misclassified.

1 Linear threshold functions

Consider the class of linear threshold functions on $\mathcal{X} = \mathbb{R}^d$,

$$F = \left\{x \mapsto \text{sign}(\theta' x) : \theta \in \mathbb{R}^d\right\}.$$

The decision boundaries are hyperplanes through the origin ($d-1$-dimensional subspaces), and the decision regions are half-spaces.

Because of their simplicity, such functions are widely used. For instance, in an object recognition problem, the x vectors might be (hand-crafted) features of locations in an image. In a document classification problem, they might be counts of different words (called a bag-of-words representation).

Notice that we work with thresholded linear functions, so the decision boundaries are subspaces. If we wish to consider thresholded affine functions, so the decision boundaries are arbitrary hyperplanes, this involves a simple transformation: We can write

$$\begin{aligned} F &= \left\{x \mapsto \text{sign}(\theta' x + c) : \theta \in \mathbb{R}^d,\, c \in \mathbb{R}\right\} \\ &= \left\{x \mapsto \text{sign}(\theta' \tilde{x}) : \theta \in \mathbb{R}^{d+1}\right\}, \end{aligned}$$

where we define $\tilde{x}' = (x'1)$. For notational simplicity, we'll stick to the linear case.

Let's consider empirical risk minimization over the class of linear threshold functions. Looking at it through the lens of the three key issues of approximation, estimation, and computation also gives a taste of some of the later parts of the course.

Approximation There are probability distributions for which linear threshold functions are optimal (for instance, if the class-conditional distributions are isotropic Gaussians with the same variances). However, it is a very restricted class of decision functions on $\mathbb{R}^d$. We'll see later that we can overcome

this difficulty, and retain many of the attractive properties of linearly parameterized functions, by first considering a nonlinear transformation $\phi : \mathbb{R}^d \to \mathbb{R}^D$ for some $D \gg d$. (This is the approach taken by kernel methods. There, we can avoid explicitly computing $\phi(x)$, which can be advantageous if D is very large.)

Estimation If we minimize empirical risk over the class, will that ensure that the risk is near minimal? It suffices if the size n of the data set is large compared to the dimension d (the number of parameters). This is not necessary: we'll see that we can achieve good estimation properties in high dimensional parameter spaces if we consider, for example, Euclidean balls in parameter space, rather than the whole space.

Computation It turns out that minimization of empirical risk over the class of linear threshold functions is easy if the best f has $\hat{R}(f) = 0$. In that case, it corresponds to solving a linear program. Otherwise, empirical risk minimization over this class is a difficult problem: NP-hard, in general. One strategy we'll investigate—it's a strategy used in support vector machines and AdaBoost—is to replace the 0-1 loss with a convex loss function, so that the minimization of this new empirical risk becomes a convex optimization.

2 The perceptron algorithm

The perceptron algorithm maintains a parameter vector, which it updates using a currently misclassified (x_i, y_i) pair. Figure 2 illustrates the intuition: each time the parameter vector is updated, the decision boundary is shifted so that the example used for the update becomes closer to being correctly classified.

The following theorem shows that whenever there is a linear threshold function that correctly classifies all of the training data, then the perceptron algorithm terminates after a finite number of updates and returns such a function. The number of iterations depends on how much slack the data allows in defining a suitable decision boundary.

input : training data, $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n) \in \mathbb{R}^d \times \{\pm 1\}$
output: linear threshold function, $f : \mathbb{R}^d \to \{\pm 1\}$

1.1 Set $\theta_0 := 0$, $t := 0$;
1.2 **while** *some* i *has* $y_i \neq \text{sign}(\theta_t' x_i)$ **do**
1.3 | pick some i with $y_i \neq \text{sign}(\theta_t' x_i)$;
1.4 | $\theta_{t+1} := \theta_t + y_i x_i$;
1.5 | $t := t + 1$;
1.6 **end**
1.7 return $x \mapsto \text{sign}(\theta_t' x)$

Algorithm 1: perceptron

Theorem 2.1. Suppose that, for some $\theta \in \mathbb{R}^d$ and all i, $y_i = \text{sign}(\theta' x_i)$. Then for any sequence of choices made at step (1.3) of the algorithm, we have

(a) The perceptron algorithm terminates.

(b) If we define

$$r = \max_i \|x_i\| \qquad \delta = \min_i \left(\frac{\theta' x_i y_i}{\|\theta\|} \right),$$

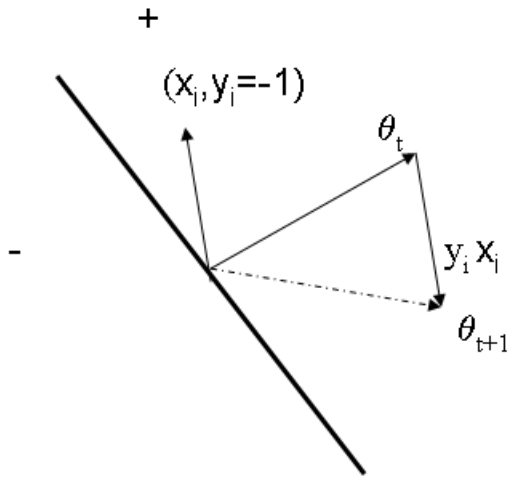


Figure 1: *Perceptron algorithm*

then the algorithm terminates after

$$t \leq \frac{r^2}{\delta^2}$$

updates.

Notice that r is the radius of a ball containing the data, and δ is the margin between the separating hyperplane and the closest point. (Also, we could rescale the data so the δ and r both get larger, but the bound on the number of iterations of the algorithm would be unchanged.)

PROOF. Clearly $(b) \implies (a)$, so we'll prove (a).

Fix a parameter vector θ that satisfies the conditions of the theorem, that is, that separates the data with a large margin. The intuition behind the proof is that θ_t and θ become more aligned with each update, because $\theta_t'\theta$ increases linearly with t, whereas $\|\theta_t\|$ does not increase that rapidly.

First, $\theta_0'\theta = 0$ and

$$\begin{aligned}\theta_{t+1}'\theta &= \theta_t'\theta + y_i x_i'\theta \\ &\geq \theta_t'\theta + \delta\|\theta\|,\end{aligned}$$

where the inequality follows from the margin assumption. Clearly, $\theta_t'\theta \geq t\delta\|\theta\|$.

On the other hand, $\|\theta_0\| = 0$, and

$$\begin{aligned}\|\theta_{t+1}\|^2 &= \|\theta_t + y_i x_i\|^2 \\ &= \|\theta_t\|^2 + \|x_i\|^2 + 2y_i\theta_t' x_i \\ &\leq \|\theta_t\|^2 + r^2,\end{aligned}$$

where the inequality follows from the definition of r and the fact that the (x_i, y_i) pair chosen at step (1.3) is misclassified. Clearly, $\|\theta_t\|^2 \leq tr^2$.

Using Cauchy-Schwarz, we have

$$t\delta\|\theta\| \leq \theta_t'\theta \leq \|\theta_t\|\|\theta\| \leq \sqrt{t}r\|\theta\|.$$

Squaring and dividing both sides by $t\delta^2\|\theta\|^2$ gives the result. □

Note. If we let i_t denote the (x_i, y_i) pair chosen in the ith update, we can write

$$\theta_t = \sum_{s \le t} y_{i_s} x_{i_s} = \sum_{i=1}^{n} \alpha_i x_i,$$

where $\sum |\alpha_i| \le t$. That is, instead of representing the parameter vector θ_t directly, it can be expressed in terms of the coefficients α of this combination of the x_is. And this combination is sparse (has few non-zero components) if n is large compared to t.

The perceptron algorithm relies on $x_i's$ only via $\theta_t^{'} x_i = \sum_{j=1}^{n} \alpha_j (x_j^{'} x_i)$. Thus, we do not need to manipulate the points x_i directly, only their inner products with each other. This suggests the idea of mapping the data points to a higher-dimensional feature-space $\phi : \mathbb{R}^d \to \mathbb{R}^D$, with $D \gg d$. As long as we can efficiently compute the inner products between the $\phi(x_i)$, we need never explicitly compute the representation in $\mathbb{R}^D$.

3 A minimax lower bound on risk

Suppose we are in a situation where $n \le d$. In particular, if $x_1, \ldots, x_n$ are linearly independent, then for any $y_1, \ldots, y_n$, we can find $\theta \in \mathbb{R}^d$ such that

$$\theta' \left[x_1 | x_2 | \cdots | x_n \right] = [y_1, y_2, \ldots, y_n],$$

and hence $\text{sign}(\theta' x_i) = y_i$. If we can fit *any* labels, we should not expect to predict the labels of subsequent points accurately.

The following theorem makes this precise, by showing that, under these conditions, any method will perform poorly. It shows a little more: when n/d is not too large, there is a probability distribution that makes the risk at least as large as $\Omega(n/d)$.

Theorem 3.1. For any $n \ge 1$ and any mapping $f_n : \mathbb{R}^d \times \left(\mathbb{R}^d \times \{\pm 1\} \right)^2 \to \{\pm 1\}$, there is a probability distribution on $\mathbb{R}^d \times \{\pm 1\}$ for which some linear threshold function $f \in F$ has $R(f) = 0$ but

$$\mathbb{E} R(f_n) \ge \min \left(\frac{n-1}{2n}, \frac{d-1}{2n} \right) \left(1 - \frac{1}{n} \right)^n.$$

This is an example of a minimax lower bound, since it gives a lower bound on

$$\min_{f_n} \max_{P} \mathbb{E} R(f_n),$$

where the max is over all P for which some $f \in F$ has zero risk, and the min is over all methods f_n—not just the perceptron algorithm, or other algorithms that predict with linear threshold functions, but any deterministic prediction rule.

One thing to notice about the result is that the probability distribution is allowed to depend on n. So it's not saying that, for every P the risk decreases to zero at a rate slower than $1/n$.

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 3**

Minimax risk bounds for linear threshold functions

Lecturer: Peter Bartlett *Scribe: Hao Zhang*

1 Review

We assume that there is a probability distribution P on $\mathcal{X} \times \mathcal{Y}$, and that the pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ and (X, Y) are chosen independently according to P. The aim is to choose f so that the *risk* of f,

$$R(f) = \mathbb{E}\ell(f(X), Y),$$

is small. For instance, in the pattern classification example, $\mathcal{Y} = \{\pm 1\}$, and the risk is the misclassification probability.

$$R(f) = \mathbb{E}1[f(X) \neq Y] = \Pr(f(X) \neq Y).$$

One family of approaches to pattern classification problems is to fix a class F of functions that map from $\mathcal{X}$ to $\mathcal{Y}$ and choose f_n from F. For example, consider the class of linear threshold functions on $\mathcal{X} = \mathbb{R}^d$,

$$F = \left\{ x \mapsto \text{sign}(\theta' x) : \theta \in \mathbb{R}^d \right\}.$$

The decision boundaries are hyperplanes through the origin ($d-1$-dimensional subspaces), and the decision regions are half-spaces.

We have seen that, when there is a linear threshold function that classifies all of the training data correctly, the perceptron algorithm terminates after a finite number of mistakes and returns such a function. The number of mistakes depends on the scale of the margin between the two classes.

Although the perceptron algorithm performs well when the margin is large, notice that it does not necessarily maximize the margin. (Later, we'll consider other methods that do maximize the margin.)

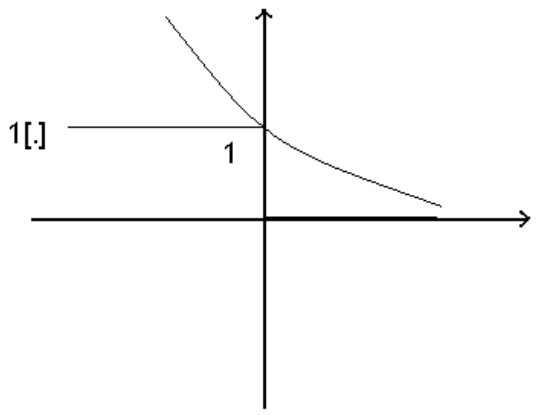
The perceptron algorithm converges only when there is a linear threshold function that correctly classifies all of the training data. In those cases, it can be viewed as choosing a linear function $g(x) = \theta' x$ such that the risk of the decision function, $f(x) = \text{sign}(g(x))$, is zero, and hence minimal. A question was asked about what can be done if there is no perfect linear threshold function. Notice that we can write

$$R(f(X)) = R(\text{sign}(g(X))) \leq Pr(Yg(X) \leq 0) = \mathbb{E}(1[Yg(X) \leq 0]).$$

If G denotes the set of linear functions, then the perceptron algorithm can be thought of as minimizing over G the sample average of the indicator function of a misclassification, that is,

$$\min_{g \in G} \hat{\mathbb{E}}(1[Yg(X) \leq 0]),$$

where $\hat{\mathbb{E}}$ denotes the expectation under the empirical distribution. In general, if the minimum of this expectation is non-zero, this minimization problem is intractable. An alternative is to replace the indicator function $1[\cdot \leq 0]$ with some convex loss function $\phi(\cdot)$. This loss function should penalize negative values of its argument so as to favor solutions for which Y and $g(X)$ tend to have the same sign. Typically, ϕ is some kind of convex approximation of the step function, as indicated in the figure below. This is the approach taken by the SVM (with ϕ the hinge loss) and AdaBoost (with ϕ the exponential loss).



2 A Minimax lower bound on risk

We made the observation that if $n \leq d$, then we should anticipate difficulties. In particular, if $x_1, \ldots, x_n$ are linearly independent, then for any $y_1, \ldots, y_n$, we can find $\theta \in \mathbb{R}^d$ such that

$$\theta' [x_1 | x_2 | \cdots | x_n] = [y_1, y_2, \ldots, y_n],$$

and hence $\text{sign}(\theta' x_i) = y_i$. If we can fit *any* labels, we should not expect to predict the labels of subsequent points accurately.

The following theorem makes this precise, by showing that, under these conditions, any method will perform poorly.

Theorem 2.1. For any $n \geq 1$ and any mapping $f_n : \mathbb{R}^d \times \left(\mathbb{R}^d \times \{\pm 1\}\right)^n \to \{\pm 1\}$, there is a probability distribution on $\mathbb{R}^d \times \{\pm 1\}$ for which some linear threshold function $f \in F$ has $R(f) = 0$ but

$$\mathbb{E}R(f_n) \geq \min\left(\frac{n-1}{2n}, \frac{d-1}{2n}\right)\left(1 - \frac{1}{n}\right)^n.$$

This is an example of a minimax lower bound, since it gives a lower bound on

$$\min_{f_n} \max_{P} \mathbb{E}R(f_n),$$

where the max is over all P for which some $f \in F$ has zero risk, and the min is over all methods f_n—not just the perceptron algorithm, or other algorithms that predict with linear threshold functions, but any deterministic prediction rule.

One thing to notice about the result is that the probability distribution is allowed to depend on n. So it's not saying that, for some distribution, the risk must decrease to zero at slower than a $1/n$ rate.

Proof. The broad idea of the proof is to use the probabilistic method: choose the probability distribution P uniformly at random from a class $\mathcal{P}$, and show that the expectation of $R(f_n)$ (expectation both under this random choice and under the choice of the data) is large. This implies that there is a distribution in the class that makes $R(f_n)$ large. Notice that this kind of proof is not constructive: we don't find out which distribution is bad for a particular algorithm f_n (like the perceptron algorithm). Certainly the bad

distribution must depend on f_n. Randomization is an elegant way of showing that each f_n must fail in some case.

Each distribution in the class satisfies two properties:

1. There is a linear threshold function f with $R(f) = 0$. To achieve this, we fix a linearly independent set $S = \{z_1, \ldots, z_n\} \in \mathbb{R}^d$. Then we restrict our marginal distributions on $\mathcal{X}$ to have support in S. By the observation above, for any $b = (b_1, \ldots, b_d) \in \{\pm\}^d$, we can find a linear threshold function f_b satisfying $f_b(z_i) = b_i$ for all i.
2. A sample of size n is unlikely to contain much information about f. To do this, we'll concentrate most of the probability on a single point, say z_d, and make the others unlikely.

For each $b \in \{\pm 1\}^d$, define the probability distribution P_b via

$$P_b(x, y) = \begin{cases} \frac{\epsilon}{d-1} & \text{if } (x, y) = (z_i, b_i) \text{ for } i = 1, \ldots, d-1, \\ 1 - \epsilon & \text{if } (x, y) = (z_d, b_d), \\ 0 & \text{otherwise.} \end{cases}$$

We'll choose b uniformly at random. Under this choice of b and of the data, we have

$$\mathbb{E}R(f_n) = \sum_{k=0}^{d-1} \mathbb{E}\left[R(f_n) || U| = k\right] \Pr(|U| = k),$$

where $U = \{z_1, \ldots, z_{d-1}\} - \{X_1, \ldots, X_n\}$ is the set of 'light' elements of S that are unseen. The key observation is that, for these unseen elements, the corresponding b_i might as well have been chosen afterwards, since they are independent of the earlier choices. So on those points, the decision rule can do no better than tossing a coin. Formally, we can write

$$\mathbb{E}\left[R(f_n) || U| = k\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(f_n(X), Y) | \text{data}, \, |U| = k\right] || U| = k\right],$$

and

$$\begin{aligned} \mathbb{E}\left[\ell(f_n(X), Y) | \text{data}\right] &= \sum_{i=1}^{d} \mathbb{E}\left[\ell(f_n(z_i), b_i) | \text{data}\right] \Pr(X = z_i) \\ &\geq \sum_{z_i \in U} \mathbb{E}\left[\ell(f_n(z_i), b_i) | \text{data}\right] \Pr(X = z_i) \\ &= |U| \times \frac{1}{2} \times \frac{\epsilon}{d-1}. \end{aligned}$$

Combining, we have

$$\mathbb{E}R(f_n) \geq \sum_{k=0}^{d-1} \frac{k\epsilon}{2(d-1)} \Pr(|U| = k), = \frac{\epsilon}{2(d-1)} \mathbb{E}|U|.$$

But the expected number of unseen light elements is

$$\begin{aligned} \mathbb{E}|U| &= \mathbb{E} \sum_{i=1}^{d-1} 1\left[z_i \notin \{X_1, \ldots, X_n\}\right] \\ &= \sum_{i=1}^{d-1} \Pr\left(z_i \notin \{X_1, \ldots, X_n\}\right) \\ &= (d-1)\left(1 - \frac{\epsilon}{d-1}\right)^n. \end{aligned}$$

Thus,

$$\mathbb{E}R(f_n) \geq \frac{\epsilon}{2}\left(1 - \frac{\epsilon}{d-1}\right)^n.$$

And now we just need to choose appropriate values of ϵ. If $n \geq d-1$, choose $\epsilon = (d-1)/n$ shows that $\mathbb{E}R(f_n) \geq (d-1)(1-1/n)^n/(2n)$. If $n < d-1$, choose $\epsilon = (n-1)/n (< (d-1)/n)$ shows that

$$\mathbb{E}R(f_n) \geq \frac{n-1}{2n}\left(1 - \frac{n-1}{(d-1)n}\right)^n \geq \frac{n-1}{2n}\left(1 - \frac{1}{n}\right)^n.$$

□

We'll see that, if n is large compared to d, then over the set of linear threshold functions, the empirical risk is uniformly close to the true risk, and so any linear function with zero empirical risk must have small risk. In particular, if the perceptron algorithm terminates, the function that it returns will have small risk.

Is d the right measure of the complexity of the class of linear threshold functions? Not always. If there is a large margin classifier, so that the perceptron converges quickly, then the solution it finds incorporates only few (r^2/δ^2) of the n (x_i, y_i) pairs. If this number is small compared to n, then the data has been *compressed* in some sense: the algorithm could have seen just this subset and produced the same (empirically correct) classifier.

The following theorem shows that the risk of the classifier returned by (something like) the perceptron algorithm is small whenever n is large compared to (something like) r^2/δ^2.

Theorem 2.2. Suppose that, for some $r > 0$, $\theta \in \mathbb{R}^d$ and $\delta > 0$, we have, almost surely,

$$\|X\| \leq r \qquad \text{and} \qquad \frac{Y\theta' X}{\|\theta\|} \geq \delta.$$

Consider the following randomized variant of the perceptron algorithm: Choose M uniformly from $\{1, \ldots, n\}$. Pass the initial data subsequence $(X_1, Y_1), \ldots, (X_M, Y_M)$ to the perceptron algorithm. Let f_n be the linear threshold function that it returns. Then

$$\mathbb{E}R(f_n) \leq \frac{r^2}{n\delta^2}.$$

(Notice that the expectation includes the randomization of the algorithm.)

Proof. Explicitly writing the expectation over the random choice of M shows that

$$\begin{aligned}
\mathbb{E}R(f_n) &= \frac{1}{n}\sum_{m=1}^{n} \mathbb{E}\ell(f_m(X; X_1, Y_1, \ldots, X_m, Y_m), Y) \\
&= \frac{1}{n}\sum_{m=1}^{n} \mathbb{E}\ell(f_m(X_{m+1}; X_1, Y_1, \ldots, X_m, Y_m), Y_{m+1}) \\
&= \mathbb{E}\frac{1}{n}\sum_{m=1}^{n} \ell(f_m(X_{m+1}; X_1, Y_1, \ldots, X_m, Y_m), Y_{m+1}),
\end{aligned}$$

where the second equality follows from the fact that (X_i, Y_i) and (X, Y) have the same distribution. (Let (X_{m+1}, Y_{m+1}) denote (X, Y).) Now consider

$$\frac{1}{n}\sum_{m=1}^{n} \ell(f_m(X_{m+1}; X_1, Y_1, \ldots, X_m, Y_m), Y_{m+1}).$$

With probability 1, the data will all satisfy the constraint on the radius and the margin. The perceptron convergence theorem shows that, with this data, the total number of iterations of the algorithm cannot exceed r^2/δ^2. Consider the perceptron algorithm when the misclassified pair chosen for the update is the one with smallest index. Then the total number of updates, U, satisfies

$$\sum_{m=1}^{n} \ell(f_m(X_{m+1}; X_1, Y_1, \ldots, X_m, Y_m), Y_{m+1}) \leq U,$$

because this sum ignores the updates that are made on a pair with an index that is smaller than the biggest update index so far. And since $U \leq r^2/\delta^2$, we have $\mathbb{E}R(f_n) \leq r^2/(n\delta^2)$. □

And we can extend the minimax lower bound to show that, in a minimax sense, this is the best that we can hope for (up to constants) for a probability distribution satisfying the radius and margin conditions. We'll see this in the next lecture.

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 4**

Kernel Methods for Pattern Classification

Lecturer: Peter Bartlett *Scribe: Kristal Sauer*

1 The Perceptron Algorithm

Recall the following upper bound on the expected risk of a randomized version of the perceptron algorithm.

Theorem. Suppose that, for some $r, \delta > 0$ and $\theta \in \mathbb{R}$, we have a.s. $\| X \| < r$ and $\frac{Y\theta' X}{\|\theta\|} \geq \delta$. Then, the following randomized variant of the perceptron algorithm returns f_n with

$$\mathbb{E}R(f_n) \leq \frac{r^2}{n\delta^2}$$

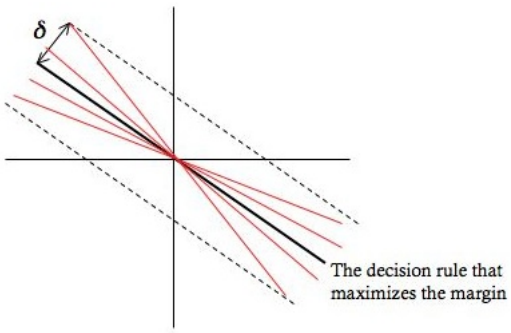


Figure 1. Choose the decision rule that yields the maximum margin.

The following is a converse result: a lower bound on the expected risk of any algorithm.

Theorem. For any decision rule f_n, there is a P on $\mathbb{R}^d \times \{\pm 1\}$ s.t. for some $\theta \in \mathbb{R}^d$, a.s.

$$\frac{\theta' XY}{\| \theta \|} \geq \delta$$

$$\| X \| \leq r$$

but

$$\mathbb{E}R(f_n) \geq \frac{\min\left(d, n, r^2/\delta^2\right) - 1}{2n} \left(1 - \frac{1}{n}\right)^n.$$

The **proof idea** is as follows:

We have support of P_x on the scaled basis vectors,

$$\left\{ re_i : 1 \leq i \leq \left\lfloor \frac{r^2}{\delta^2} \right\rfloor \right\}, \text{where } \left\lfloor \frac{r^2}{\delta^2} \right\rfloor = k$$

$$\theta = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} b_i e_i$$

Then,

$$Y_i \theta' X_i = \frac{r}{\sqrt{k}} \sim \delta \text{ (modulo some factor).}$$

This concludes our discussion of the perceptron algorithm. We now continue to kernel methods.

2 Kernel Methods for Pattern Classification

We have seen that, when there is a large margin linear threshold function, the perceptron algorithm will quickly find a linear threshold function that classifies the data, stopping when it achieves classification of the training examples. Note that it does not choose a large margin linear threshold function.

It is interesting to consider the performance of a method that does maximize the margin, given by $\delta = \frac{Y\theta' X}{\|\theta\|}$. An example of such a method is the *Support Vector Machine* (SVM).

Consider the following optimization problem.

$$\boxed{\max_{\gamma,w} \gamma \text{ s.t. } \frac{y_i w' x_i}{\|w\|} \geq \gamma \; i = 1, \ldots, n}$$

We could add the constraints $\| w \| \leq 1$ or $\| w \| = \frac{1}{\gamma}$ to lower the number of variables, because we only really care about the direction of w.

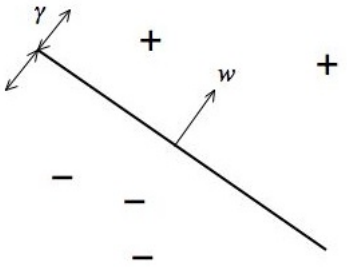


Figure 2. SVM optimizes γ.

We may thus express the optimization problem given as the equivalent optimization problem,

$$\boxed{\min_{w \in \mathbb{R}^d} \| w \|^2 \text{ s.t. } y_i w' x_i \geq 1 \; i = 1, \ldots, n}$$

We call this the *Hard-Margin SVM*. This optimization problem is the *primal* problem. Note that we are maximizing a quadratic criterion subject to linear constraints; thus, we have a *Quadratic Program* (QP).

Recall that for constrained optimization, we use *Lagrange Multipliers*.

We introduce Lagrange Multipliers $\alpha_i \geq 0$. We have one for each $i = 1, \ldots, n$.

Lagrangian:

$$L(w,\alpha) := \frac{1}{2} \parallel w \parallel^2 - \sum_{i=1}^{n} \alpha_i \left(y_i w' x_i - 1\right)$$

(We have added the scaling factor of $\frac{1}{2}$ for convenience—it tidies the constants in what follows.)

We wish to minimize $L(w,\alpha)$ wrt w, the *primal* variables, and maximize it wrt α, the *dual* variables. That is, we wish to find the saddle point of $L(w,\alpha)$.

Suppose we're at a saddle point.

- If $y_i w' x_i < 1$, we could increase α_i to increase L. So at the saddle point, we must have the constraints satisfied.
- If $y_i w' x_i > 1$, then $\alpha_i = 0$.
- In any case, $\alpha_i \left(y_i w' x_i - 1\right) = 0 \forall i$

$\implies$ At the saddle point (w,α), w is primal feasible. $\implies$ At the saddle point, $L(w,\alpha) = \frac{1}{2} \parallel w \parallel^2$.

At the saddle point,

$$\frac{\partial}{\partial w} L(w,\alpha) = 0 \implies w* = \sum_{i=1}^{n} \alpha_i y_i x_i$$

Consider especially points for which $\alpha_i > 0 \implies y_i w' x_i = 1$. Such points are referred to as *support vectors*. Only the support vectors enter into the constraint to determine the solution to the optimization problem.

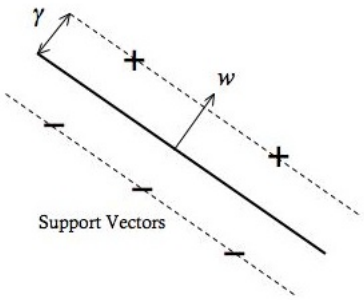


Figure 3. Support vectors enter into constraint.

Let us also consider the dual optimization problem:

Substituting $w*$ into $L(w,\alpha)$:

$$\begin{aligned} g(\alpha) &:= L(w*,\alpha) \\ &= \frac{1}{2} \sum_{i,j} \alpha_i y_i x_i' x_j y_j \alpha_j - \sum_{i,j} \alpha_i y_i x_i' x_j y_j \alpha_j + \sum_i \alpha_i \\ &= \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i y_i x_i' x_j y_j \alpha_j \end{aligned}$$

Dual:

$$\boxed{\max_{\alpha\in\mathbb{R}^d} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i' x_j \text{ s.t. } \alpha_i \geq 0 \; i=1,\ldots,n}$$

Notice that the x_is only enter the optimization problem via the inner products.

As with the perceptron algorithm, we can express the solution in terms of a mapping to an inner product space:

$$\begin{aligned} f_n(x) &= \operatorname{sign}\left(\langle w, \Phi(x)\rangle\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^n \alpha_i y_i \langle \Phi(x_i), \Phi(x)\rangle\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^n \alpha_i y_i k(x_i, x)\right) \end{aligned}$$

where $k(x_i, x)$ is the kernel and α is a solution to

$$\boxed{\max_{\alpha\in\mathbb{R}^d} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(x_i, x_j) \text{ s.t. } \alpha_i \geq 0 \; i=1,\ldots,n}$$

To recap, note the two key ideas of SVMs:

- Maximized margin
- Arbitrary inner product

Let us now consider some examples of kernels.

- Polynomial

$$\begin{aligned} k_2(u,v) &= (u'v)^2 \\ k_2(u,v) &= \left(\sum_{i=1}^d u_i v_i\right)^2 \\ &= \left(u_1^2, \sqrt{2}u_1u_2, u_2^2\right)\begin{pmatrix} v_1^2 \\ \sqrt{2}v_1v_2 \\ v_2^2 \end{pmatrix} \\ &= \Phi_2(u)'\,\Phi_2(v) \;\text{ with } \Phi_2 : \mathbb{R}^2 \mapsto \mathbb{R}^3 \end{aligned}$$

  Note. The feature space might not be unique; eg,

$$\begin{aligned} \psi_2(u) &= \left(u_1^2, u_1u_2, u_2u_1, u_2^2\right) \\ k_2(u,v) &= \psi_2(u)'\,\psi_2(v) \end{aligned}$$

- Gaussian

$$k_G(u,v) = \exp\left(\frac{-\parallel u-v\parallel^2}{2\sigma^2}\right) = \langle \phi_G(u), \phi_G(v)\rangle$$

  Here, u and v are infinite-dimensional vectors.

Above, the kernels are defined on $\mathbb{R}^d$; however, we may talk about any space (e.g., documents) on which we can define an inner product.

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 5**

Constrained Optimization

Lecturer: Peter Bartlett *Scribe: Nimar S. Arora*

The reference for this lecture is Chapter 5 of Boyd and Vanderberghe's *Convex Optimization*.

1 Primal

Consider the optimization problem *(primal problem)*:

$$\begin{aligned} p^* = \min_{x \in \mathbb{R}^n} \; & f_0(x) \\ \text{s.t. } & f_i(x) \leq 0, \qquad i = 1, \ldots, m \end{aligned}$$

The optimal value is

$$p^* = f_0(x^*)$$

Define the Lagrangian:

$$\begin{aligned} L : & \mathbb{R}^{n+m} \to \mathbb{R} \\ L(x, \lambda) = & f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \end{aligned}$$

The λ_is are called dual variables or Lagrange multipliers with $\lambda_i \geq 0$

2 Saddle Point

See Figure 1 for an example of a saddle point.

In a minimax problem, if the min player gets to play second he can achieve a lower value. Thus,

$$d^* = \sup_{\lambda \geq 0} \inf_x L(x, \lambda) \;\leq\; \inf_x \sup_{\lambda \geq 0} L(x, \lambda)$$

Suppose there are x^* and λ^* s.t.,

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$$

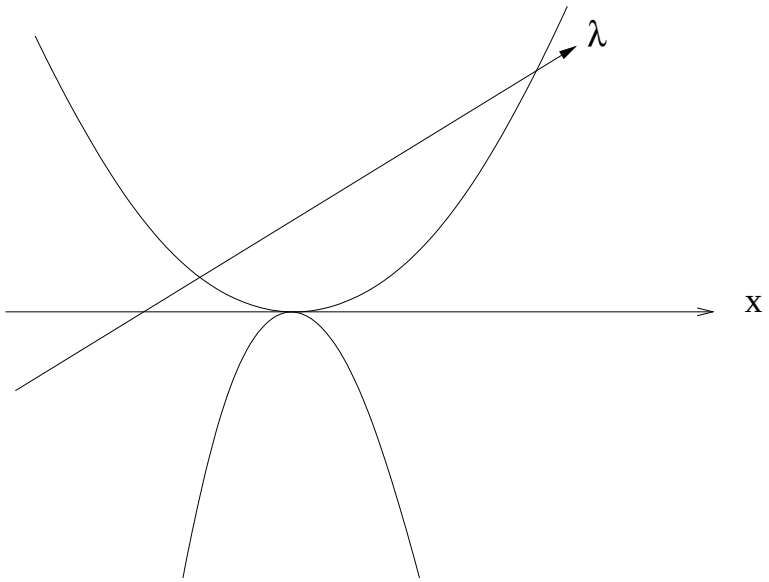


Figure 1: Saddle Point

for all feasible x and $\lambda \geq 0$. Then,

$$\begin{aligned}
\inf_x \sup_{\lambda \geq 0} L(x,\lambda) &\leq \sup_{\lambda \geq 0} L(x^*,\lambda) \qquad (\text{fix } x = x^*) \\
&= L(x^*,\lambda^*) \\
&= \inf_x L(x,\lambda^*) \\
&\leq \sup_{\lambda \geq 0} \inf_x L(x,\lambda) \qquad (\text{fixing } \lambda = \lambda^* \text{ we get previous}) \\
\text{So,} \qquad \inf_x \sup_{\lambda \geq 0} L(x,\lambda) &= \sup_{\lambda \geq 0} \inf_x L(x,\lambda)
\end{aligned}$$

3 Lagrange Dual Function

Define the Lagrange dual function:

$$\begin{aligned}
g(\lambda) &= \inf_x L(x,\lambda) \\
&= \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x)\right)
\end{aligned}$$

Note,

1. $g(\lambda)$ is concave (point-wise minima of concave functions)
2. If $\lambda_i \geq 0$ and x is primal feasible (i.e. $f_i(x) \leq 0$) then,

$$g(\lambda) \leq f_0(x)$$

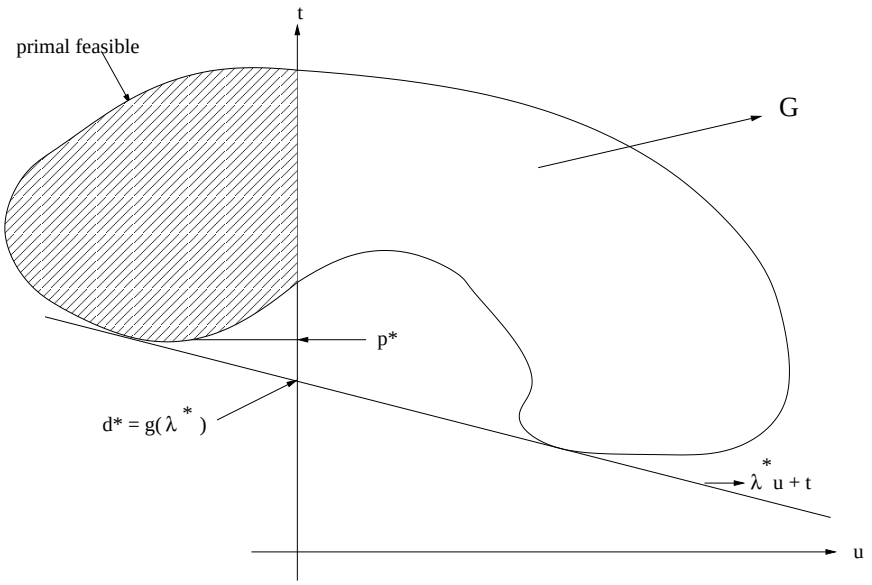


Figure 2: Geometric Interpretation of Duality

3. In particular, $\forall \lambda \geq 0$, $g(\lambda) \leq p^*$

4 Dual

$$\begin{aligned} \text{Dual:} \quad & \max \ g(\lambda) \\ & \text{s.t. } \lambda \geq 0 \\ \text{Optimal Value:} \quad & g(\lambda^*) = d^* \end{aligned}$$

Note,

1. The dual is always a maximization of a concave function with convex constraints
2. Weak duality implies that $d^* \leq p^*$
3. The optimal duality gap is $p^* - d^*$

5 Geometric Interpretation

Define,

$$\mathcal{G} = \{(u,t) : \ \exists x \ f_i(x) = u_i; \ f_0(x) = t\}$$

$$
\begin{aligned}
g(\lambda) &= \inf_x \{f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x)\} \\
&= \inf_{(u,t)\in\mathcal{G}} (t + \sum_{i=1}^{m} \lambda_i u_i) \\
&= \begin{bmatrix} \lambda^T & 1 \end{bmatrix}^T \begin{bmatrix} u \\ t \end{bmatrix}
\end{aligned}
$$

For m=1, the set

$$\{(u,t) : (\lambda\ 1)^T \begin{pmatrix} u \\ t \end{pmatrix} = c\}$$

is a line with slope λ and intercept $t = c = g(\lambda)$. See Figure 2 for an illustration of the set G and the Lagrange Dual.

6 Strong Duality

Weak duality states that $d^* \leq p^*$. Strong duality states $d^* = p^*$. Strong duality holds if f_0 and f_i are convex and there is a suitable qualification on the constraint. For example, Slater's condition requires that the primal is strictly feasible:

$$\exists x \qquad f_i(x) < 0 \qquad i = 1 \ldots m$$

7 Complementary Slackness

If there is zero duality gap,

$$
\begin{aligned}
f_0(x^*) &= g(\lambda^*) \\
&= \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x) \right) \\
&\leq f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) \qquad \text{(fixing } x = x^*\text{)}
\end{aligned}
$$

$$
\begin{aligned}
\text{Hence,} \qquad \sum_{i=1}^{m} \lambda_i^* f_i(x^*) &\geq 0 \\
\text{But,} \qquad f_i(x^*) &\leq 0 \\
\text{and} \qquad \lambda_i^* &\geq 0 \\
\text{So,} \qquad \sum_{i=1}^{m} \lambda_i^* f_i(x^*) &= 0 \\
\text{and hence} \qquad \lambda_i^* f_i(x^*) &= 0 \qquad \forall i
\end{aligned}
$$

If constraint i is inactive at x^* (i.e. $f_i(x^*) < 0$) then $\lambda_i^* = 0$.

8 KKT Optimality Conditions

If f_0 and f_i are differentiable, $\exists\ x^*,\ \lambda^*$ which are optimal, and the duality gap is zero

$$\Rightarrow\ KKT(x^*,\lambda^*)\ =\ \begin{cases} f_i(x^*)\ \leq\ 0 & i=1\dots m \\ \lambda_i^*\ \geq\ 0 & i=1\dots m \\ \lambda_i^* f_i(x^*)\ =\ 0 & i=1\dots m \\ \bigtriangledown f_0(x^*)\ +\ \sum_{i=1}^m \lambda_i^*\ \bigtriangledown\ f_i(x^*)\ =\ 0 & \end{cases}$$

Also, $KKT(x,\lambda)$ and f_0, f_i convex $\Rightarrow\ x,\ \lambda$ are optimal and the duality gap is zero.

If f_0, f_i are convex, differentiable, and the duality gap is zero then $KKT(x,\lambda)\ \Leftrightarrow\ (x,\lambda)$ optimal.

9 SVM

Primal

$$\begin{array}{|ll|} \hline \min & \frac{1}{2}\|w\|^2 \\ \text{s.t.} & y_i w' x_i\ \geq\ 1 \qquad i=\dots m \\ \hline \end{array}$$

$$L(w,\alpha)\ =\frac{1}{2}\|w\|^2\ +\ \sum_{i=1}^n \alpha_i(1\ -\ y_i w' x_i)$$

substituting $\quad w^*\ =\sum_{i=1}^n \alpha_i y_i x_i$

$$g(\alpha)\ =\ \sum_{i=1}^n \alpha_i\ -\ \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i' x_j$$

Strong duality (if feasible).

Complementary Slackness:

$$y_i {w^*}' x_i\ >1\ \Rightarrow\ \alpha_i^*\ =\ 0$$

for i s.t. $\quad \alpha_i^*\ >0 \qquad x_i$ is a support vector

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 6**

Non-separable (soft) SVMs

Lecturer: Peter Bartlett *Scribe: Joseph Austerweil*

Outline of lecture:

1. Another geometric interpretation of hard-margin SVMs
2. Standard soft-margin SVM (C-SVM)
3. ν-SVM (interpretable reparameterization of C-SVM)

1 Another Geometric Interpretation of Hard-Margin SVMs

Previously, we explored the following definition of the SVM and the resulting geometric interpretation of its dual function (also shown in figure 1):

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & \|w\|^2 \\ \text{s.t.} \quad & \forall_i, y_i w' x_i \geq 1 \end{aligned}$$

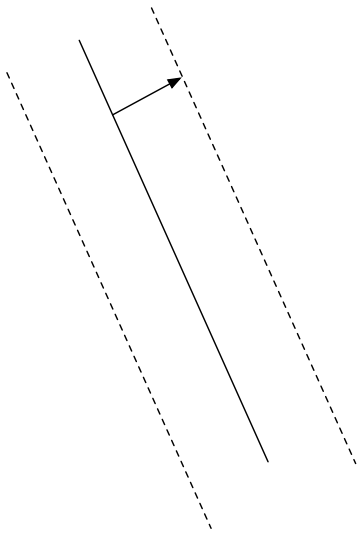


Figure 1: Hard SVM with its margins and decision boundary. The size of the margin is $\frac{1}{\|w\|^2}$

Instead, we can directly represent the margin, γ, which yields an equivalent optimization, but a different

dual function.

$$\begin{array}{lll} \max\limits_{w\in\mathbb{R}^d} & \gamma & \\ \text{s.t.} & \forall_i, y_i w' x_i \geq \gamma & \text{(dual parameter } \lambda_i\text{)} \\ & \|w\|^2 \leq 1 & \text{(dual parameter } \beta\text{)} \end{array}$$

We form the Lagrangian (switching the criterion to be a minimization of $-\gamma$):

$$L(w,\gamma,\lambda,\beta) = -\gamma + \sum_{i=1}^{n} \lambda_i(\gamma - y_i w' x_i) + \beta(\|w\|^2 - 1)$$

and at the minimum over w and γ, we have

$$\begin{aligned} \sum_i \lambda_i &= 1 \\ w &= \frac{1}{2\beta}\sum_i \lambda_i y_i x_i \end{aligned}$$

This gives the dual function,

$$g(\lambda,\beta) = -\frac{1}{4\beta}\sum_{i,j}\lambda_i\lambda_j y_i y_j x_i' x_j - \beta$$

and the dual optimization problem is

$$\begin{array}{ll} \min & \frac{1}{4\beta}\left\|\sum \lambda_i y_i x_i\right\|^2 + \beta \\ \text{s.t.,} & \sum \lambda_i = 1, \\ & \lambda_i \geq 0, \\ & \beta \geq 0 \end{array}$$

We can remove $\beta : \beta^2 = \frac{1}{4}\left\|\sum \lambda_i y_i x_i\right\|^2$. This results in the dual optimization:

$$\min \left\|\sum \lambda_i y_i x_i\right\| \text{ s.t. } \lambda_i \geq 0, \sum_i \lambda_i = 1$$

Slater's condition implies strong duality.
We have:

$$w* = \frac{1}{2\beta}\sum_i \lambda_i y_i x_i = \frac{\sum_i \lambda_i y_i x_i}{\|\sum_i \lambda_i y_i x_i\|}$$

Or in other words, w* is the unit vector in the direction of the smallest norm element of the set

$$\text{co}(\{\sum_i y_i x_i : 1 \leq i \leq n\}) = \{\sum_i \lambda_i y_i x_i : \lambda_i \geq 0, \sum_i \lambda_i = 1\}.$$

From this formulation and Figure 2, we can observe that $w*$ points from the origin to the closest point on the convex hull formed by the positive and negative points (reflected through origin).

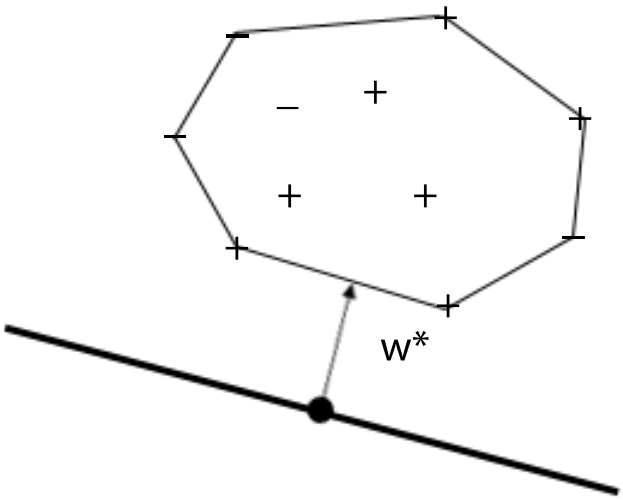


Figure 2: Optimal weight vector is from origin to closest point on convex hull formed from positive and reflected negative examples

2 Non-separable ("soft") SVMs

Non-separable SVMs allow the decision boundary to misclassify some examples, but it pays a cost for the number of violated constraints. We could form the optimization to be:

$$\min_{w\in\mathbb{R}^d} \|w\|^2 + \frac{C}{n}|S^c| \quad \text{s.t.} \quad \forall i \in S : y_i w' x_i \geq 1$$

$$\min_{w\in\mathbb{R}^d} \|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} 1[y_i w' x_i < 1]$$

However, this yields a nasty combinatorial optimization problem, so instead we replace the indicator function with a convex function.

$$\min \|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \phi(y_i w' x_i)$$

One possible function that is used for the soft SVMs of today's lecture (C-SVM and ν-SVM) is the hinge loss (see Figure 3):

$$\phi(\alpha) = (1-\alpha)_+ = \begin{cases} 1-\alpha & 1-\alpha > 0 \\ 0 & \text{otherwise} \end{cases}$$

Using the hinge-loss function, we form the primal optimization of the soft SVM to be:

$$\min_{w\in\mathbb{R}^d} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} (1 - y_i w' x_i)_+$$

$$\min_{w\in\mathbb{R}^d} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \quad \text{s.t., } \forall i : \underbrace{\xi_i \geq 0}_{\lambda_i} \quad \forall i : \underbrace{1 - \xi_i \leq y_i w' x_i}_{\alpha_i}$$

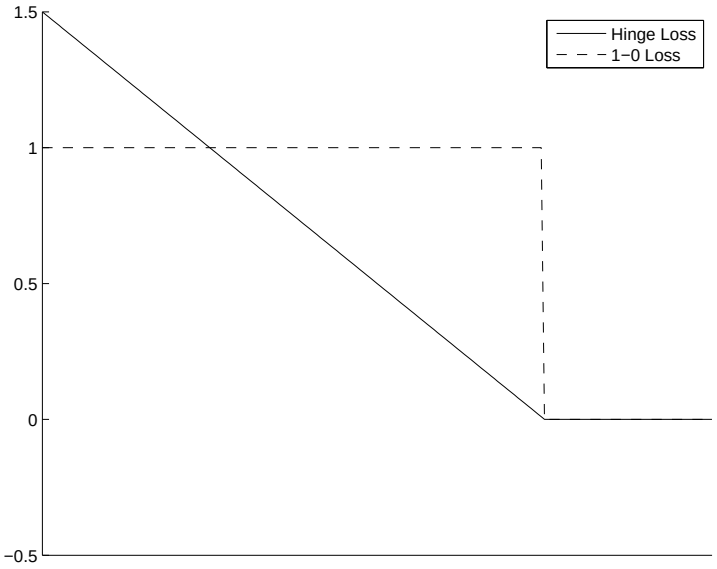


Figure 3: The hinge loss function (the solid line) is the typical loss function used for soft-margin SVMs.

C balances between the two parts of the criterion, so the larger the C the more we care about misclassified points. From this formulation, we can form the Lagrangian and derive the dual optimization:

$$L(w, \xi, \alpha, \lambda) = \frac{1}{2}\|w\|^2 + \frac{c}{n}\sum \xi_i + \sum_i \alpha_i(1 - y_i w' x_i - \xi_i) - \sum_i \lambda_i \xi_i$$

Minimizing, we remove primal variables w and ξ from the optimization.

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_i \alpha_i y_i x_i$$
$$\frac{\partial L}{\partial \xi} = 0 \quad \Rightarrow \quad \alpha_i + \lambda_i = \frac{c}{n}$$

We form dual:

$$g(\alpha, \lambda) = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i' x_j + \sum_i \alpha_i$$

Notice that we removed $\sum_i \xi(\frac{c}{n} - \alpha_i - \lambda_i)$ because $\forall i : \alpha_i + \lambda_i = \frac{c}{n}$, from the minimization. Thus, the form of the dual for the soft SVM is:

$$\begin{array}{ll} \max\limits_{\alpha,\lambda} & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i' x_j \\ \text{s.t.} & \alpha_i \geq 0 \\ & \lambda_i \geq 0 \\ & \alpha_i + \lambda_i = \frac{C}{n} \end{array}$$

We can eliminate the λ_i variables, and replace the constraints with $0 \leq \alpha_i \leq \frac{C}{n}$. This constraint tells us that we cannot include too much weight on any point (at most $\frac{C}{n}$). In the hard margin case, we saw, via

complementary slackness, that $\alpha_i > 0$ only when the corresponding example is on a margin. What is the similar condition for the soft-margin SVM?

- $\alpha_i > 0 \Rightarrow y_i w' x_i = 1 - \xi_i \leq 1$ (we are either at or on the wrong wide of the margin). The corresponding examples for $\alpha_i > 0$ are called the *support vectors*.

- $\underbrace{y_i x_i' w < 1}_{\text{"margin error"}} \Rightarrow \xi_i > 0$, and so $\lambda_i = 0 \rightarrow \alpha_i = \frac{C}{n}$

Note some examples that are classified correctly will still be considered a margin error and will have $\alpha = \frac{C}{n}$. Figure 4 shows this case. In the separable case, if C is greater than n times the largest α_i value, then the soft-margin SVM is equivalent to the hard-margin SVM.

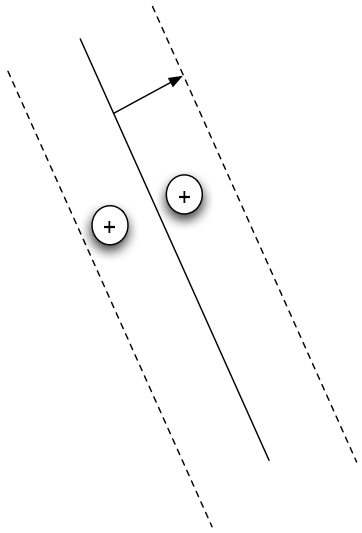


Figure 4: Both positive points, even though only one of which is misclassified, are considered margin errors and their corresponding α_i weight are $\frac{C}{n}$.

3 ν-SVM

The interpretation of C is not intuitive. We show that solving ν-SVM is an equivalent optimization problem, but ν has a more intuitive interpretation. We will show later that this can be understood as a reparamaterization of the C-SVM problem. We form ν-SVM:

$$\begin{array}{ll} \min\limits_{w,\rho} & \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}(\rho - y_i w' x_i)_+ \\ \text{s.t.} & \rho \geq 0 \end{array}$$

Figure 5 shows the ν-SVM's decision boundary. An equivalent optimization problem (a quadratic program), stated in terms of slack variables, is

$$\begin{aligned} \min_{w,\rho,\xi} \quad & \tfrac{1}{2}\|w\|^2 - \nu\rho + \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \underbrace{\rho \geq 0}_{\gamma}, \\ & \underbrace{\xi_i \geq 0}_{\beta_i}, \\ & \underbrace{\xi_i \geq \rho - y_i w' x_i}_{\alpha_i} \end{aligned}$$

Using this definition, we can derive the Lagrangian and dual formulation.

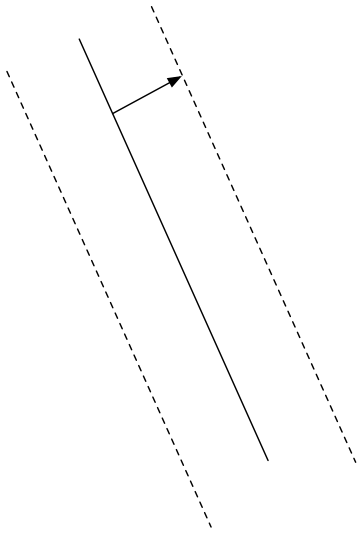


Figure 5: ν-SVM with its margins and decision boundary. The size of the margin is $\frac{\rho}{\|w\|}$

$$L(w,\rho,\xi,\alpha,\beta,\gamma) = \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum_i \xi_i - \gamma\rho - \sum_i \xi_i\beta_i - \sum_i \alpha_i(y_i w' x_i + \xi_i - \rho)$$

Taking the minimum over our primal variables, w, ρ, and ξ, yields:

$$w = \sum_i \alpha_i y_i x_i \quad \nu = \textstyle\sum_i \alpha_i - \gamma \quad \beta_i + \alpha_i = \dfrac{1}{n}$$

This gives us the dual formulation:

$$\begin{aligned} \max_{\alpha} \quad & -\tfrac{1}{2}\textstyle\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i' x_j \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \tfrac{1}{n}, \\ & \textstyle\sum_i \alpha_i \geq \nu. \end{aligned}$$

Using the dual formulation, we can analyze the complementary slackness for the ν-SVM:

- $\alpha_i > 0 \Rightarrow y_i w' x_i = \rho - \xi_i \leq \rho$ (The corresponding vectors for $\alpha_i > 0$ are again called support vectors)

- $y_i w' x_i < \rho \Rightarrow \xi_i > 0 \Rightarrow \beta_i = 0 \Rightarrow \alpha_i = \frac{1}{n}$

Theorem 3.1. If $\rho > 0$ at solution, then:

$$\underbrace{|\{i : y_i w' x_i < \rho\}|}_{\text{\# of margin errors}} \overset{(a)}{\leq} |\{i : \alpha_i = \frac{1}{n}\}| \overset{(b)}{\leq} \nu n \overset{(c)}{\leq} \underbrace{|\{i : \alpha_i > 0\}|}_{\text{\# of support vectors}} \overset{(d)}{\leq} |\{i : y_i w' x_i \leq \rho\}|$$

PROOF. (a) and (d) are given by complementary slackness.

$$\text{(b) } \rho > 0 \Rightarrow \gamma = 0 \Rightarrow \nu = \sum_i \alpha_i \geq \sum_i \alpha_i 1[\alpha_i = \frac{1}{n}] = \frac{1}{n} \sum_i 1[\alpha_i = \frac{1}{n}]$$

$$\text{(c) } \nu \leq \sum_i \alpha_i \leq \frac{1}{n} \sum_i 1[\alpha_i > 0]$$

□

By Theorem 3.1, we can think of νn as roughly the proportion of support vectors. Figure 6 shows the difference between the number of margin errors and the number of support vectors.

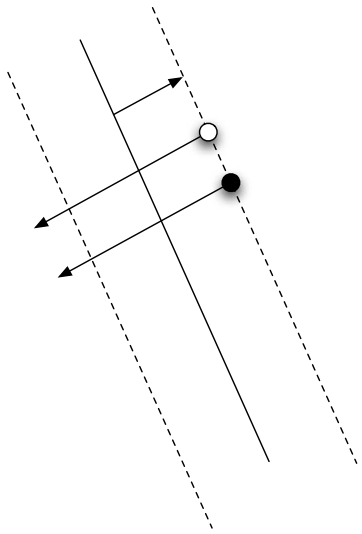


Figure 6: The decision boundary and margins once again for the ν-SVM. The open circle and arrow represent the margin errors whereas the closed circle represents the support vectors.

Theorem 3.2. If ν-SVM has a solution with $\rho > 0$, then C-SVM with $C = \frac{1}{\rho}$ gives an equivalent classifier.

PROOF. If (w_*, ρ_*) is the solution to ν-SVM, we can fix $\rho = \rho_*$ and optimizing over w will not lead to a better value. That is, w^* is a solution to the optimization problem

$$\begin{aligned} \min_w \quad & \frac{1}{2}\|w\|^2 + \frac{1}{n}\sum_i \xi_i \\ \text{s.t.} \quad & \xi_i \geq 0, \\ & y_i w' x_i \geq \rho^* - \xi_i. \end{aligned}$$

We can scale the objective by $1/{\rho^*}^2$ and the constraints by $1/\rho^*$ to obtain an equivalent optimization problem:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\left\|\frac{w}{\rho^*}\right\|^2 + \frac{1}{n\rho^*}\sum_i \frac{\xi_i}{\rho^*} \\ \text{s.t.} \quad & \frac{\xi_i}{\rho^*} \geq 0, \\ & y_i \frac{w'}{\rho^*} x_i \geq 1 - \frac{\xi_i}{\rho^*}. \end{aligned}$$

And if we replace w/ρ^* with w and ξ_i/ρ^* with ξ_i, this is equivalent to the C-SVM with $C = \frac{1}{\rho_*}$. □

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 7**

Reproducing Kernel Hilbert Spaces

Lecturer: Peter Bartlett *Scribe: Chunhui Gu*

1 Reproducing Kernel Hilbert Spaces

1.1 Hilbert Space and Kernel

An inner product $\langle u, v\rangle$ can be

1. a usual dot product: $\langle u, v\rangle = v'w = \sum_i v_i w_i$
2. a kernel product: $\langle u, v\rangle = k(v, w) = \psi(v)'\psi(w)$ (where $\psi(u)$ may have infinite dimensions)

However, an inner product $\langle \cdot, \cdot \rangle$ must satisfy the following conditions:

1. Symmetry
$$\langle u, v\rangle = \langle v, u\rangle \ \forall u, v \in \mathcal{X}$$
2. Bilinearity
$$\langle \alpha u + \beta v, w\rangle = \alpha\langle u, w\rangle + \beta\langle v, w\rangle \ \forall u, v, w \in \mathcal{X}, \forall \alpha, \beta \in \mathbb{R}$$
3. Positive definiteness
$$\langle u, u\rangle \geq 0, \ \forall u \in \mathcal{X}$$
$$\langle u, u\rangle = 0 \iff u = 0$$

Now we can define the notion of a Hilbert space.

Definition. A *Hilbert Space* is an inner product space that is complete and separable with respect to the norm defined by the inner product.

Examples of Hilbert spaces include:

1. The vector space $\mathbb{R}^n$ with $\langle a, b\rangle = a'b$, the vector dot product of a and b.
2. The space l_2 of square summable sequences, with inner product $\langle x, y\rangle = \sum_{i=1}^{\infty} x_i y_i$
3. The space L_2 of square integrable functions (i.e., $\int_s f(x)^2 dx < \infty$), with inner product $\langle f, g\rangle = \int_s f(x)g(x)dx$

Definition. $k(\cdot, \cdot)$ is a *reproducing kernel* of a Hilbert space $\mathcal{H}$ if $\forall f \in \mathcal{H}, f(x) = \langle k(x, \cdot), f(\cdot)\rangle$.

A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space H with a reproducing kernel whose span is dense in H. We could equivalently define an RKHS as a Hilbert space of functions with all evaluation functionals bounded and linear.

For instance, the L_2 space is a Hilbert space, but not an RKHS because the delta function which has the reproducing property

$$f(x) = \int_s \delta(x-u)f(u)du$$

does not satisfy the square integrable condition, that is,

$$\int_s \delta(u)^2 du \not< \infty,$$

thus the delta function is not in L_2.

Now let us define a kernel.

Definition. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *kernel* if

1. k is symmetric: $k(x,y) = k(y,x)$.
2. k is positive semi-definite, i.e., $\forall x_1, x_2, ..., x_n \in \mathcal{X}$, the "Gram Matrix" K defined by $K_{ij} = k(x_i, x_j)$ is positive semi-definite. (A matrix $M \in \mathbb{R}^{n\times n}$ is positive semi-definite if $\forall a \in \mathbb{R}^n$, $a'Ma \geq 0$.)

Here are some properties of a kernel that are worth noting:

1. $k(x,x) \geq 0$. (Think about the Gram matrix of $n = 1$)
2. $k(u,v) \leq \sqrt{k(u,u)k(v,v)}$. (This is the Cauchy-Schwarz inequality.)

To see why the second property holds, we consider the case when $n = 2$:

Let $a = \begin{bmatrix} k(v,v) \\ -k(u,v) \end{bmatrix}$. The Gram matrix $K = \begin{pmatrix} k(u,u) & k(u,v) \\ k(v,u) & k(v,v) \end{pmatrix} \succeq 0 \iff a'Ka \geq 0$

$\iff [k(v,v)k(u,u) - k(u,v)^2]k(v,v) \geq 0.$

By the first property we know $k(v,v) \geq 0$, so $k(v,v)k(u,u) \geq k(u,v)^2$.

1.2 Build an Reproducing Kernel Hilbert Space (RKHS)

Given a kernel k, define the "reproducing kernel feature map" $\Phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$ as:

$$\Phi(x) = k(\cdot, x)$$

Consider the vector space:

$$\text{span}(\{\Phi(x) : x \in \mathcal{X}\}) = \{f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i) : n \in \mathbb{N}, x_i \in \mathcal{X}, \alpha_i \in \mathbb{R}\}$$

For $f = \sum_i \alpha_i k(\cdot, u_i)$ and $g = \sum_i \beta_i k(\cdot, v_i)$, define $\langle f, g\rangle = \sum_{i,j} \alpha_i \beta_j k(u_i, v_j)$.

Note that:

$$\langle f, k(\cdot, x)\rangle = \sum_i \alpha_i k(x, u_i) = f(x), \text{ i.e., } k \text{ has the reproducing property.}$$

We show that $\langle f, g\rangle$ is an inner product by checking the following conditions:

1. Symmetry: $\langle f, g\rangle = \sum_{i,j} \alpha_i \beta_j k(u_i, v_j) = \sum_{i,j} \beta_j \alpha_i k(v_j, u_i) = \langle g, f\rangle$
2. Bilinearity: $\langle f, g\rangle = \sum_i \alpha_i g(u_i) = \sum_j \beta_j f(v_j)$
3. Positive definiteness: $\langle f, f\rangle = \alpha' K \alpha \geq 0$ with equality iff $f = 0$.

From 3 we can also derive:

1. $\langle f, g\rangle^2 \leq \langle f, f\rangle\langle g, g\rangle$

   PROOF. $\forall a \in \mathbb{R}, \langle af + g, af + g\rangle = a^2\langle f, f\rangle + 2a\langle f, g\rangle + \langle g, g\rangle \geq 0$. This implies that the quadratic expression has a non-positive discriminant. Therefore, $\langle f, g\rangle^2 - \langle f, f\rangle\langle g, g\rangle \leq 0$ □
2. $|f(x)|^2 = \langle k(\cdot, x), f\rangle^2 \leq k(x, x)\langle f, f\rangle$, which implies that if $\langle f, f\rangle = 0$ then f is identically zero.

Now we have defined an inner product space $\langle \cdot, \cdot \rangle$. Complete it to give the Hilbert space.

Definition. For a (compact) $\mathcal{X} \subseteq \mathbb{R}^d$, and a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$, we say $\mathcal{H}$ is a *Reproducing Kernel Hilbert Space* if $\exists k : \mathcal{X} \to \mathbb{R}$, s.t.

1. k has the reproducing property, i.e., $f(x) = \langle f(\cdot), k(\cdot, x)\rangle$
2. k spans $\mathcal{H} = \overline{\text{span}\{k(\cdot, x) : x \in \mathcal{X}\}}$

1.3 Mercer's Theorem

Another way to characterize a symmetric positive semi-definite kernel k is via the Mercer's Theorem.

Theorem 1.1 (Mercer's)**.** Suppose k is a continuous positive semi-definite kernel on a compact set $\mathcal{X}$, and the integral operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ defined by

$$(T_k f)(\cdot) = \int_{\mathcal{X}} k(\cdot, x) f(x) dx$$

is positive semi-definite, that is, $\forall f \in L_2(\mathcal{X})$,

$$\int_{\mathcal{X}} k(u, v) f(u) f(v) du dv \geq 0$$

Then there is an orthonormal basis $\{\psi_i\}$ of $L_2(\mathcal{X})$ consisting of eigenfunctions of T_k such that the corresponding sequence of eigenvalues $\{\lambda_i\}$ are non-negative. The eigenfunctions corresponding to non-zero eigenvalues are continuous on $\mathcal{X}$ and $k(u, v)$ has the representation

$$k(u, v) = \sum_{i=1}^{\infty} \lambda_i \psi_i(u) \psi_i(v)$$

where the convergence is absolute and uniform, that is,

$$\lim_{n \to \infty} \sup_{u,v} |k(u, v) - \sum_{i=1}^{n} \lambda_i \psi_i(u) \psi_i(v)| = 0$$

To take an analogue in the finite case, that is, $\mathcal{X} = \{x_1, \ldots, x_n\}$. Let $K_{ij} = k(x_i, x_j)$, and $f : \mathcal{X} \to \mathbb{R}^n$ with $f_i = f(x_i)$. Then,

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

$$\forall f,\ f' K f \geq 0 \Rightarrow K \succeq 0 \Rightarrow K = \sum \lambda_i v_i v_i'$$

Hence,

$$k(x_i, x_j) = K_{ij} = (V \Lambda V')_{ij} = \sum_{k=1}^{n} \lambda_k v_{ki} v_{kj} = \sum_{k=1}^{n} \lambda_k \psi_k(x_i) \psi_k(x_j) \Rightarrow \psi_k(x_i) = (v_k)_i$$

We summarize several equivalent conditions on continuous, symmetric k defined on compact $\mathcal{X}$:

1. Every Gram matrix is positive semi-definite.
2. T_k is positive semi-definite.
3. k can be expressed as $k(u, v) = \sum_i \lambda_i \psi_i(u) \psi_i(v)$.
4. k is the reproducing kernel of an RKHS of functions on $\mathcal{X}$.

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 8**

Representer theorem and kernel examples

Lecturer: Peter Bartlett *Scribe: Howard Lei*

1 Representer Theorem

Recall that the SVM optimization problem can be expressed as follows:

$$J(f^*) = \min_{f \in H} J(f)$$

where

$$J(f) = \frac{C}{n} \sum_{i=1}^{n} \text{hingeloss}\left(f(x_i), y_i\right) + ||f||_H^2$$

and H is a Reproducing Kernel Hilbert Space (RKHS).

Theorem 1.1. Fix a kernel k, and let H be the corresponding RKHS. Then, for a function L: $\mathbb{R}^n \to \mathbb{R}$ and non-decreasing Ω: $\mathbb{R} \to \mathbb{R}$, if the SVM optimization problem can be expressed as:

$$J(f^*) = \min_{f \in H} J(f) = \min_{f \in H} \left(L(f(x_1) \dots f(x_n)) + \Omega\left(||f||_H^2\right)\right)$$

then the solution can be expressed as:

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

Furthermore, if Ω is strictly increasing, then all solutions have this form.

This shows that to solve the SVM optimization problem, we only need to solve for the α_i, which agrees with the solution obtained via the Lagrangian formulation of the problem. Furthermore, our solution lies in the span of the kernels.

Proof.

Suppose we project f onto the subspace:

$$\text{span}\{k(x_i, \cdot)\colon\ 1 \le i \le n\}$$

obtaining f_s (the component along the subspace) and $f_\perp$ (the component perpendicular to the subspace). We have:

$$f = f_s + f_\perp \Rightarrow ||f||^2 = ||f_s||^2 + ||f_\perp||^2 \ge ||f_s||^2$$

Since Ω is non-decreasing,

$$\Omega(||f||_H^2) \ge \Omega(||f_s||_H^2)$$

implying that $\Omega(\cdots)$ is minimized if f lies in the subspace. Furthermore, since the kernel k has the reproducing property, we have:

$$f(x_i) = \langle f, k(x_i, \cdot)\rangle = \langle f_s, k(x_i, \cdot)\rangle + \langle f_\perp, k(x_i, \cdot)\rangle = \langle f_s, k(x_i, \cdot)\rangle = f_s(x_i)$$

Implying that:

$$L(f(x_1), \ldots, f(x_n)) = L(f_s(x_1), \ldots, f_s(x_n))$$

Hence, $L(\cdots)$ depends only on the component of f lying in the subspace: span$\{k(x_i, \cdot)\colon 1 \leq i \leq n\}$, and $\Omega(\cdots)$ is minimized if f lies in that subspace. Hence, $J(f)$ is minimized if f lies in that subspace, and we can express the minimizer as:

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

Note that if $\Omega(\cdot)$ is strictly non-decreasing, then $||f_\perp||$ must necessarily be zero for f to be the minimizer of $J(f)$, implying that f^* must necessarily lie in the subspace: span$\{k(x_i, \cdot)\colon 1 \leq i \leq n\}$.

□

2 Constructing Kernels

In this section, we discuss ways to construct new kernels from previously defined kernels. Suppose k_1 and k_2 are valid (symmetric, positive definite) kernels on $\mathcal{X}$. Then, the following are valid kernels:

1. $k(u, v) = \alpha k_1(u, v) + \beta k_2(u, v)$, for $\alpha, \beta \geq 0$

   Proof.

   Since $\alpha k_1(u, v) = \langle\sqrt{\alpha}\Phi_1(u), \sqrt{\alpha}\Phi_1(v)\rangle$ and $\beta k_2(u, v) = \langle\sqrt{\beta}\Phi_2(u), \sqrt{\beta}\Phi_2(v)\rangle$, then:

   $$\begin{aligned} k(u, v) &= \alpha k_1(u, v) + \beta k_2(u, v) && (1)\\ &= \langle\sqrt{\alpha}\Phi_1(u), \sqrt{\alpha}\Phi_1(v)\rangle + \langle\sqrt{\beta}\Phi_2(u), \sqrt{\beta}\Phi_2(v)\rangle && (2)\\ &= \langle[\sqrt{\alpha}\Phi_1(u)\ \sqrt{\beta}\Phi_2(u)], [\sqrt{\alpha}\Phi_1(v)\ \sqrt{\beta}\Phi_2(v)]\rangle && (3)\end{aligned}$$

   and we see that $k(u, v)$ can be expressed as an inner product

   □

2. $k(u, v) = k_1(u, v)k_2(u, v)$

   Proof.

   Note that the gram matrix K for k is the Hadamard product (or element-by-element product) of K_1 and K_2 ($K = K_1 \bigodot K_2$). Suppose that K_1 and K_2 are covariance matrices of $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_n)$ respectively. Then K is simply the covariance matrix of $(X_1Y_1, \ldots, X_nY_n)$, implying that it is symmetric and positive definite. □

3. $k(u, v) = k_1(f(u), f(v))$, where $f\colon \mathcal{X} \to \mathcal{X}$

   Proof.

   Since f is a transformation in the same domain, k is simply a different kernel in that domain:

   $$k(u, v) = k_1(f(u), f(v)) = \langle\Phi(f(u)), \Phi(f(v))\rangle = \langle\Phi_f(u), \Phi_f(v)\rangle$$

   □

4. $k(u, v) = g(u)g(v)$, for g: $\mathcal{X} \to \mathbb{R}$

   Proof.

   We can express the gram matrix K as the outer product of the vector $\gamma = [g(x_1), \ldots, g(x_n)]'$. Hence, K is symmetric and positive semi-definite with rank 1. (It is positive semi-definite because the non-zero eigenvalue of $\gamma\gamma'$ is the trace of $\gamma\gamma'$ which is the trace of $\gamma'\gamma$ which is simply $\gamma'\gamma$ which is greater than or equal to 0).

   □

5. $k(u, v) = f(k_1(u, v))$, where f is a polynomial with positive coefficients.

   Proof.

   Since each polynomial term is a product of kernels with a positive coefficient, the proof follows by applying 1 and 2.

   □

6. $k(u, v) = \exp(k_1(u, v))$

   Proof.

   Since:

   $$\exp(x) = \lim_{i \to \infty} \left(1 + x + \cdots + \frac{x_i}{i!}\right)$$

   The proof follows from 5 and the fact that:

   $$k(u, v) = \lim_{i \to \infty} k_i(u, v)$$

   □

7. $k(u, v) = \exp\left(\frac{-||u-v||^2}{\sigma^2}\right)$

   Proof.

   $$k(u, v) = \exp\left(\frac{-||u-v||^2}{\sigma^2}\right) = \exp\left(\frac{-||u||^2 - ||v||^2 + 2u'v}{\sigma^2}\right) \tag{4}$$

   $$= \left(\exp\left(\frac{-||u||^2}{\sigma^2}\right) \exp\left(\frac{-||v||^2}{\sigma^2}\right)\right) \exp\left(\frac{2u'v}{\sigma^2}\right) \tag{5}$$

   $$= (g(u)g(v))\exp(k_1(u, v)) \tag{6}$$

   $g(u)g(v)$ is a kernel according to 4, and $\exp(k_1(u, v))$ is a kernel according to 6. According to 2, the product of two kernels is a valid kernel.

   □

Note that the Gaussian kernel is translation-invariant, where $k(u, v)$ can be expressed as $f(u - v) = f(x)$.

Example: Translation-invariant kernels

Consider the function f: $[-\pi, \pi] \to \mathbb{R}$, and suppose that f is continuous and even (i.e. $f(x) = f(-x)$). Then, we can express f via the Fourier expansion as:

$$f(x) = \sum_{n=0}^{\infty} a_n \cos(nx)$$

where $a_n \geq 0$.

If we let x be the difference of u and v, then we have:

$$f(x) = f(u-v) = a_0 + \sum_{n=1}^{\infty} a_n(\sin(nu)\sin(nv) + \cos(nu)\cos(nv)) \tag{7}$$

$$= \sum_{i=0}^{\infty} \lambda_i \Psi_i(u)\Psi_i(v), \tag{8}$$

where $\{\Psi_i\} = \{\sin(nu) : n \geq 1\} \cup \{\cos(nu) : n \geq 0\}$.

We see that $f(u-v)$ is a valid kernel that's translation invariant. This example shows that we can choose the kernel by choosing the a_i coefficients, which is equivalent to choosing a filter.

Example: Bag-of-words kernel

Suppose that $\Phi_w(d)$ is the number of times word w appears in document d. If we want to classify documents by their word counts, we can use the kernel $k(d_1, d_2) = \langle \Phi(d_1), \Phi(d_2) \rangle$. (In practice, these counts are weighted to take into account the relative frequency of different words.)

Example: Marginalized kernel

Given the probability distribution $p(x, h)$ (and hence $p(h|x)$) and a kernel defined for (x,h) pairs ($k((x, h), (x', h'))$), we can obtain a kernel on only the x's as follows:

$$k_m(x, x') = \sum_{h,h'} k((x, h), (x', h'))p(h|x)p(h'|x')$$

Exercise: Prove that this is a valid kernel!

Example: Convolution kernel (or "string" kernel)

Define a_i to be a letter of the alphabet, $s = (s_i, \ldots, s_\ell)$ to be a string of letters, and Σ^* to be the space of all possible letter sequences.

Suppose that s has $a = (a_1, \ldots, a_n)$ as a subsequence if there exists a sequence of indices $I = (i_1, \ldots, i_n)$, where $i_1 < i_2 < \cdots < i_n$ with $s_{i_j} = a_j$, where $j = 1, \ldots, n$. Define the length of the set of indices $(i_1, \ldots, i_n)$ forming the subsequence as $\ell(I) = i_n - i_1 + 1$. For simplicity, we use the notation $s[I] = a$.

Define, for fixed n, the feature map for a particular sequence a and string s:

$$\Phi_a(s) = \sum_{I:\ s[I]=a} \lambda^{\ell(I)}$$

where $\lambda \in (0, 1)$. To compare two strings s and s', we can use the following kernel:

$$k(s, s') = \sum_{a \in \Sigma^n} \Phi_a(s)\Phi_a(s')$$

We can also derive the above kernel via convolution. Define the following kernel:

$$k_0((s,i),(s',i')) = 1[s(i) = s'(i')]$$

Set

$$k_n((s,i),(s',i')) = k_0((s,i),(s',i'))(h * k_{n-1})((s,i),(s',i'))$$

where $h(i-j) = 1[i-j>0]\lambda^{-(i-j)}$, and $*$ is the convolution operator. Then:

$$(h * k_{n-1})((s,i),(s',i')) = \sum_{j,j'} h(i-j)h(i'-j')k_{n-1}((s,i),(s',i'))$$

and

$$k(s,s') = \sum_{i,i'} k_n((s,i),(s',i'))$$

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 9**

Convex loss vs. 0–1 loss

Lecturer: Peter Bartlett *Scribe: Shaunak Chatterjee and Norm Aleks*

1 Marginalized kernel, continued

Following up on the prior lecture, we verify that the marginalized kernel is indeed a kernel. First define a kernel on x: $k((x,h),(x',h'))$. (For example, x could be a DNA sequence and h, the hidden variable, could be the role it plays in the genome.) Using $\Pr(x,h)$ and $\Pr(h|x)$,

$$\begin{aligned}
k_m(x,x') &= \sum_{h,h'} k((x,h),(x',h'))\Pr(h|x)\Pr(h'|x') \\
&= \sum_{h,h'} k((x,h),(x',h'))k_1((h,x),(h',x')) \quad (k_1 \text{ is a kernel on } (x,h) \text{ pairs}) \\
&= \sum_{h,h'} k_2((h,x),(h',x')) \quad (\text{The product of two kernels is also a kernel}) \\
&= \sum_{h,h'} \Phi(h,x)^T\Phi(h',x') \\
&= \left[\sum_h \Phi(h,x)\right]^T \left[\sum_{h'} \Phi(h',x')\right] \\
&= \tilde{\Phi}(x)^T\tilde{\Phi}(x') \\
&= \tilde{k}(x,x')
\end{aligned}$$

2 0–1 loss vs. convex loss

Or, what's the impact of using a computationally convenient loss function?

The basic optimization problem we are working with is:

$$\min_w \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^n \phi(y_i w^T x_i)$$

where ϕ is any convex function; so far, we have been using the hinge function, $\phi(\alpha) = (1-\alpha)_+$ (where $x_+ = \max(0,x)$), as our example. In the optimization, the first addend increases with the complexity of the function, and the second decreases with a better fit to the data. This is an example of a *regularized empirical risk criterion*, which in general has the form

$$\text{Regularized empirical risk} = (\text{loss of } f \text{ on sample}) + (\text{complexity penalty on } f)$$

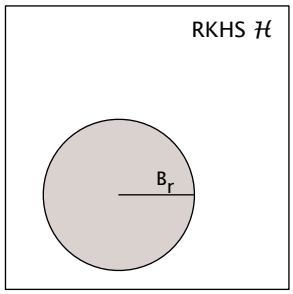


Figure 1: *We are working with functions within this ball, of radius B_r, in the Hilbert space.*

Think of the regularized empirical risk as equivalent to

$$\min_f (\text{loss of } f \text{ on sample}) \quad \text{s.t. complexity } (f) \leq B$$

There is a trade-off between the approximation and estimation errors: as we choose a more complex F (for example, larger radius B), we have a richer function class, and so can obtain smaller approximation error. However, at the same time, the estimation error—that is, the difference between the performance of the function that we choose and the performance of the best function in the class—becomes larger. This is illustrated in Figure 2.

For today we are focusing only on the difference in the expectation of the losses, and considering what happens when we replace the 0-1 loss, which is of interest in pattern classification, with a convex loss. That is, we might wish to choose $f \in F$, where $F = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, to minimize the risk,

$$\Pr(Y \neq f(X)) = \mathbb{E}1[y_i \neq f(x_i)]$$

For computational convenience we work with a convex loss function rather than the indicator/step loss function. That is, we consider:

$$\min_{f \in F} \mathbb{E}\ \phi(Yf(X)) \qquad \text{vs.} \qquad \min_{f \in F} \mathbb{E}\ 1[Y \neq \text{sign}(f(X))]$$

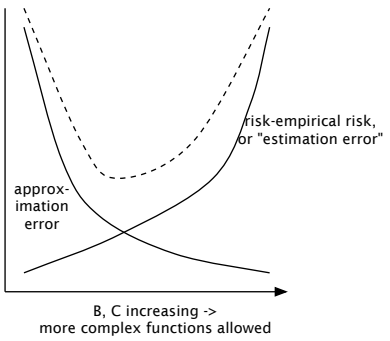


Figure 2: *A look at the trade-off between approximation error and estimation error for increasing values of B_r and C.*

This naturally leads to an interesting question: when does minimization of $R_\phi(f)$ (which equals $\mathbb{E}\ \phi(Yf(x))$) lead to small $R(f)$ (which equals $\mathbb{E}\ 1[Y \neq \text{sign}(f(X))$)?

Observation: If $\phi(\alpha) \geq 1[\alpha \leq 0]$ (that is, the loss according to ϕ is always at least the true loss), then $R(f) \leq R_\phi(f)$. (This is a weak observation if $R^* = \inf R(f) > 0$.)

If it is the case that $R^* > 0$, what more can we say? When does $R_\phi(f) = R_\phi^*(f)$ $(= \inf R_\phi(f))$ imply that $R(f) = R^*$?

Let's consider a fixed $x \in X$. Define $\eta(x) = \Pr(Y = 1|X = x)$.

$$\begin{aligned} R_\phi(f) &= \mathbb{E}\ \phi(Y(f(x))) \\ &= \mathbb{E}\ \mathbb{E}\ [\phi(Yf(x))|X] \end{aligned}$$

and

$$\mathbb{E}(\phi(Yf(x))|X = x) = \eta(x)\phi(f(x)) + (1 - \eta(x))\phi(-f(x))$$

Define the optimal value of this criterion as

$$\begin{aligned} &\mathcal{H} : [0, 1] \to \mathbb{R} \\ &\mathcal{H}(\eta) = \inf_{\alpha \in \mathbb{R}} (\eta\phi(\alpha) + (1 - \eta)\phi(-\alpha)) \end{aligned}$$

(For example, substituting in the hinge function for ϕ, we get $\mathcal{H}(\eta) = 2\min(\eta, 1 - \eta)$). Now let us define:

$$\alpha^*(\eta) = \operatorname*{arg\,min}_{\alpha \in \mathbb{R} \cup \{\pm\infty\}} (\eta\phi(\alpha) + (1 - \eta)\phi(-\alpha))$$

For example, using the hinge function for ϕ, we get:

$$\alpha^*(\eta) = \text{sign}(\eta - \frac{1}{2}).$$

Choice of the wrong sign of α, that is, different from $\text{sign}(\eta - 1/2)$, must result in a value of $\eta\phi(\alpha) + (1 - \eta)\phi(-\alpha)$ that is larger than $H(\eta)$ (otherwise minimization won't yield the correct answer), so we first define an optimizer with the "wrong sign":

$$\mathcal{H}^-(\eta) = \inf_{\alpha \text{ s.t. } \alpha(\eta - 1/2) \leq 0} (\eta\phi(\alpha) + (1 - \eta)\phi(-\alpha))$$

Definition. ϕ is "classification-calibrated" if $\eta \neq \frac{1}{2} \Rightarrow \mathcal{H}^-(\eta) > \mathcal{H}(\eta)$

Theorem 2.1. For ϕ convex and classification-calibrated,

$$\forall f, \Psi(R(f) - R^*) \leq R_\phi(f) - R_\phi^* \text{ , where}$$

$$\begin{aligned} \Psi(\theta) &= \mathcal{H}^- \left(\frac{1 + \theta}{2} \right) - \mathcal{H} \left(\frac{1 + \theta}{2} \right) \\ &= \phi(0) - \mathcal{H} \left(\frac{1 + \theta}{2} \right) \end{aligned}$$

Also, $\Psi(0) > 0$ iff $\phi > 0$.

For example, with the hinge function,

$$\begin{aligned}\Psi(\theta) &= 1 - \mathcal{H}\left(\frac{1+\theta}{2}\right) \\ &= 1 - 2\min\left(\frac{1+\theta}{2}, \frac{1-\theta}{2}\right) \\ &= 1 - 2\left(\frac{1}{2} + \frac{1}{2}\min(\theta, -\theta)\right) \\ &= |\theta|\end{aligned}$$

PROOF. Recall from Lecture 1 that

$$R(f) - R^* = \mathbb{E}\ \left[1\left[\text{sign}(f(x)) \neq \text{sign}\left(\eta(x) - \frac{1}{2}\right)\right]\ |2\eta(x) - 1|\right]$$

By Jensen's inequality,

$$\begin{aligned}\Psi(R(f) - R^*) &\leq \mathbb{E}\,\Psi\left(1\left[\text{sign}(f(x)) \neq \text{sign}\left(\eta(x) - \frac{1}{2}\right)\right]\ |2\eta(x) - 1|\right) \quad \text{because } \phi(0) = 0 \\ &= \mathbb{E}\left[1\left[\text{sign}(f(x)) \neq \text{sign}\left(\eta(x) - \frac{1}{2}\right)\right]\ \Psi(|2\eta(x) - 1|)\right] \quad \text{by } \phi\text{'s def. and symmetry of } \mathcal{H},\ \mathcal{H}^- \text{ around } \frac{1}{2} \\ &= \mathbb{E}\left[1\left[\text{sign}(f(x)) \neq \text{sign}\left(\eta(x) - \frac{1}{2}\right)\right]\ (\mathcal{H}^-(\eta(x)) - \mathcal{H}(\eta(x)))\right] \\ &\leq \mathbb{E}\left[\phi(Yf(x)) - \mathcal{H}(\eta(x))\right] \\ &= R_\phi(f) - R_\phi^* \quad \text{by definitions of } R_\phi(f) \text{ and } \mathcal{H}\end{aligned}$$

The final inequality follows because:

- if $\text{sign}(f) \neq \text{sign}\left(\eta(x) - \frac{1}{2}\right)$, then $\mathbb{E}\left[\phi(Yf(x))|x\right] \geq \mathcal{H}^-(\eta(x))$;
- otherwise, $\mathbb{E}\left[\phi(Yf(x))|x\right] \geq \mathcal{H}(\eta(x))$.

□

Lemma 2.2. For ϕ convex, ϕ is classification-calibrated iff

1. ϕ is differentiable at 0
2. $\phi'(0) < 0$

PROOF. If conditions (1) and (2) are true, it is easy to verify that the convex function ϕ is classification-calibrated.

Now for the "only if" part:

Suppose ϕ is not differentiable at zero, then it can be shown that even small perturbations of η around $\eta = 0.5$ will not move the minimum away from $\phi(0)$. This is illustrated in the figure when $\eta = 0.4$ and $\eta = 0.6$, $\mathcal{H}(\eta) = \mathcal{H}^-(\eta) = \phi(0)$, which obviously violates our definition of ϕ being classification calibrated.

□

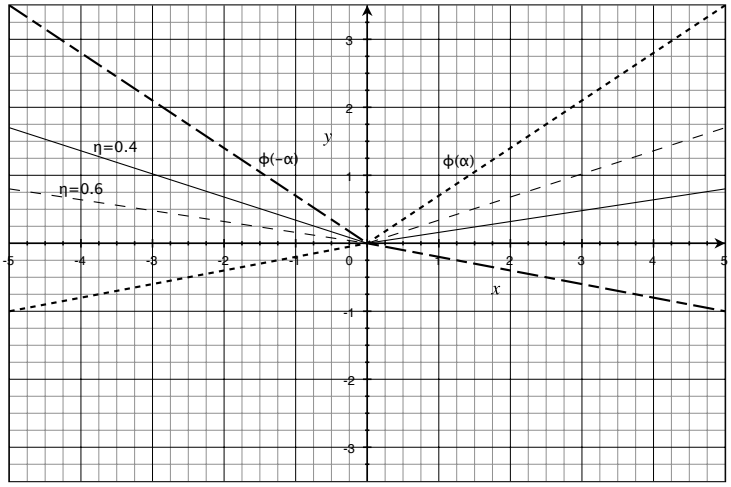


Figure 3: *Convex ϕ where ϕ is not differentiable at 0.*

It is worthwhile to consider the possibility of making the RHS of Theorem 2.1 closer to zero, since it is the upper bound on the function of the excess risk.

Notice that $R_\phi(f) - R_\phi^*$ will decrease monotonically as we increase the radius of the ball in Hilbert space, allowing more and more functions to be considered.

$$R_\phi(f_n) - R_\phi^* = (R_\phi(f_n) - \inf_{f \in B_r} R_\phi(f)) + (\inf_{f \in B_r} R_\phi(f) - R_\phi^*)$$

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 10**

Kernel ridge regression, Gaussian processes, and ensemble methods

Lecturer: Peter Bartlett *Scribe: Kevin Canini*

1 Loss & maximum likelihood

A classification or regression problem is typically formulated with a cost term of the form:

$$\frac{1}{n}\sum \phi(y_i, f(x_i)) + \text{penalty}(f)$$

where $\phi(\cdot,\cdot)$ is the estimation error and penalty(f) is the regularization term.

For certain loss functions, we can interpret minimizing this expression as maximizing the probability of the data. This viewpoint is not useful, although, for the hinge loss function. In that case, because ϕ is not differentiable, the minimizer of $\mathbb{E}\left[\phi(Y, f(X))|X\right]$ is not an invertible function of the conditional probability $P(Y = 1|X)$.

2 Kernel ridge regression

Ridge regression adds a regularization penalty (scaled by λ) to the cost term, as follows:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\|f\|_{\mathcal{H}}^2$$

As alluded to earlier, minimizing the empirical risk of a data set, using the above cost term, is equivalent to maximizing the log-likelihood of the data for a certain probability model and loss function.

If we model $Y = f(X) + Z$, where Z is additive noise, the kernel regression formulation is

$$\begin{aligned} &\min \lambda\|w\|^2 + \sum \xi_i^2 \\ &\text{s.t. } \xi_i = y_i - \langle w, x_i\rangle \end{aligned}$$

Computing the Lagrangian and using calculus to minimize over w and ξ, gives

$$\begin{aligned} w &= \frac{1}{2\lambda}\sum \alpha_i x_i \\ \xi &= \frac{\alpha_i}{2} \end{aligned}$$

and hence the solution to the dual is

$$\alpha = 2\lambda(K + \lambda I)^{-1} y$$

with

$$\begin{aligned} K_{ij} &= \langle x_i, x_j \rangle \\ y &= (y_1, \ldots, y_n)' \end{aligned}$$

Thus, the solution is

$$f(x) = y'(K + \lambda I)^{-1} k,$$

where $k = (k(x_1, x), \ldots, k(x_n, x))'$ is the vector of inner products between the data and the new point, x.

3 Bayesian viewpoint: Gaussian processes

There is a Gaussian process interpretation of kernel ridge regression. We first define a prior on the regression function f. Suppose that f is drawn from a Gaussian process, such that for all n and all $x_1, \ldots, x_n \in \mathcal{X}$, there is a matrix $\Sigma \in \Re^{n \times n}$ and $(f(x_1), \ldots, f(x_n))' \sim N(0, \Sigma)$. The entry $\Sigma_{i,j}$ of the matrix Σ specifies the covariance between $f(x_i)$ and $f(x_j)$.

Consider a model $y = f(x) + \xi$ with $\xi \sim N(0, \sigma^2)$, and suppose that we observe data $(x_1, y_1), \ldots, (x_n, y_n)$ and we wish to predict $f(x_0)$, where x_0 is a new test point. Consider the posterior distribution of

$$(f(x_0), f(x_1), \ldots, f(x_n))$$

given this data. It is

$$\mathbb{P}\left((f(x_0), \ldots, f(x_n))' | x_0, x_1, \ldots, x_n, y_1, \ldots, y_n\right) \propto \mathbb{P}\left(y|t\right) \mathbb{P}\left(t_0, t | x_0, x_1, \ldots, x_n\right)$$

$$\begin{aligned} \mathbb{P}\left(y|t\right) &\propto e^{-\frac{1}{2\sigma^2}\|y-t\|^2} \\ \mathbb{P}\left(t_0, t | x_0, x_1, \ldots, x_n\right) &\propto e^{-\frac{1}{2}(t_0, t')\Sigma^{-1}(t_0, t)'} \end{aligned}$$

where

1. Σ is the prior covariance of $(f(x_0), f(x_1), \ldots, f(x_n))'$,
2. $t = (f(x_1), \ldots, f(x_n))'$, and
3. $t_0 = f(x_0)$.

It is an easy calculation to see that the posterior mean of $t_0 = f(x_0)$ is $y'(\Sigma + \sigma^2 I)^{-1} k$, where k is the first column of Σ. Notice that we can interpret the Gram matrix K in kernel ridge regression as the covariance of a Gaussian process prior.

4 Ensemble methods

In pattern classification problems, we use ensemble methods to form a "committee" of classifiers, using some sort of voting schemes. The hope is that even though any single classifier might not perform well, the ensemble performs better.

For example, if $f_i : \mathcal{X} \to \{\pm 1\}$, we can take a majority vote among $f_1(x), \ldots, f_M(x)$ to determine $f(x)$.

The underlying functions can be many things, e.g.

- linear threshold functions: $\sum \alpha_i f_i(x) = \sum \alpha_i \text{ sign}(w_i' x)$
- decision trees
- decision stumps: a decision tree with a single test, e.g., $y = \mathbf{1}[x_7 \geq 3]$
- a dictionary of simple functions

4.1 AdaBoost (Freund-Schapire '95)

We start with a uniform distribution over the n data points:

$$D_1(i) = \frac{1}{n} \text{ for } i = 1, \ldots, n$$

and the function $F_0(x) = 0$.

We go through a specified number of iterations; for each $t \in \{1, \ldots, T\}$, choose $f_t \in G$ to (approximately) minimize $\epsilon_t = \sum_{i=1}^n D_t(i)\mathbf{1}[f_t(x_i) \neq y_1]$. Then make the following updates:

$$F_t = F_{t-1} + \alpha_t f_t$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{\alpha_t} & \text{if } f_t(x_i) \neq y_i \\ e^{-\alpha_t} & \text{otherwise} \end{cases}$$

Here,

$$\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}$$

$$Z_t = 2\sqrt{\epsilon_t(1 - \epsilon_t)}$$

AdaBoost will overfit if you let T get very large, e.g., as big as n^2. However, T any smaller than linear in n will not overfit in a precise sense (for a suitably rich class of base classifiers, AdaBoost with $T \to \infty$ slower than linearly in n will be *universally consistent*, that is, the risk of the classifier it produces will approach the Bayes risk).

4.1.1 Theorem:

The empirical probability

$$\begin{aligned} \hat{P}(YF_T(X) \leq 0) = \frac{1}{n}|\{i : y_i F_t(x_i) \leq 0\}| &\leq \prod_{t=1}^T Z_t \\ &= \prod_{t=1}^T 2\sqrt{\epsilon_t(1 - \epsilon_t)} \end{aligned}$$

Furthermore, if $\epsilon_t \leq \frac{1}{2} - \gamma$ for all t, then

$$\begin{aligned} \prod_{i=1}^T Z_t &\leq \prod_{i=1}^T 2\sqrt{\frac{1}{2^2} - \gamma^2} \\ &= (1 - 4\gamma^2)^{T/2} \\ &\leq \delta, \text{ for } T \geq \frac{\ln 1/\delta}{2\gamma^2} \end{aligned}$$

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 11**

AdaBoost and large margin classifiers

Lecturer: Peter Bartlett *Scribe: Matt Johnson*

1 Review

Algorithm 1 AdaBoost

1: $D_1(i) \Leftarrow \frac{1}{n}, \forall i \in \{1, \ldots, n\}$
2: $F_0(x) \Leftarrow 0$
3: **for** $t = 1, \ldots, T$ **do**
4: choose $f_t \in G$ to approximately minimize $\sum_{i=1}^{n} D_t(i) 1\left[f_t(x_i) \neq y_i\right]$
5: $\alpha_t \Leftarrow \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
6: $F_t \Leftarrow F_{t-1} + \alpha_t f_t$
7: $D_{t+1}(i) \Leftarrow \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{\alpha_t} & \text{if } f_t(x_i) \neq y_i \\ e^{-\alpha_t} & \text{otherwise} \end{cases}$
8: **end for**

Note that the Z_t term on line 1 can be thought of as simply a normalizer to ensure that $D_t(i)$ remains a distribution. We will see later in this lecture that $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

2 AdaBoost Analysis

2.1 Performance Bound

The following theorem shows that, if the ϵ_ts are significantly below 1/2, then we can get the proportion of training data misclassified arbitrarily small. The proof actually shows that we can view AdaBoost as an algorithm that greedily minimizes $\hat{\mathbb{E}} e^{-Yf(X)}$.

Theorem 2.1.

$$\hat{P}\left(YF_T(x) \leq 0\right) = \frac{1}{n}\left|\{i : y_i F_T(x_i) \leq 0\}\right| \tag{1}$$

$$\leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} \tag{2}$$

Furthermore, if we know that ϵ_t is slightly less than $\frac{1}{2}$, say $\epsilon_t \leq \frac{1}{2} - \gamma \, \forall t$, the product above is no more than $(1-4\gamma^2)^{\frac{T}{2}}$.

Proof. Instead of the event $YF_T(X) \leq 0$, look at the equivalent event $\exp(-YF_T(X)) \geq 1$. So, plugging in for F_T, we have

$$\hat{P}\left(YF_T(X) \leq 0\right) \leq \hat{\mathbb{E}}\left[\exp(-YF_T(X))\right] \tag{3}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i\sum_{t=1}^{T}\alpha_t f_t(x_i)) \tag{4}$$

$$= \frac{1}{n}\sum_{i}\prod_{t}\exp(-y_i\alpha_t f_t(x_i)) \tag{5}$$

We also know that, since $y_i, f(x_i) \in \{\pm 1\}$, their product is also in $\{\pm 1\}$. Note that the exponentiation in the above expression is in the D_{t+1} expression of the algorithm, so we have

$$= \frac{1}{n}\sum_{i}\prod_{t}\frac{D_{t+1}(i)}{D_t(i)}Z_t \tag{6}$$

$$= \frac{1}{n}\sum_{i}\left(\prod_{t}Z_t\right)\frac{D_{T+1}}{D_1(i)} \tag{7}$$

$$= \prod_{t}Z_t \tag{8}$$

Where we have the final equality because $D_1(i) = 1/n$ and D_{T+1} is a distribution, so it sums over i to one.

If we choose α_t to minimize

$$Z_t = \sum_{i:y_i=f_t(x_i)}D_t(i)e^{-\alpha_t} + \sum_{i:y_i\neq f_t(x_i)}e^{\alpha_t} \tag{9}$$

$$= (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \tag{10}$$

We can differentiate w.r.t. α_t and set to zero to solve the optimization to get

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Which gives

$$Z_t = (1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \tag{11}$$

$$= 2\sqrt{\epsilon_t(1-\epsilon_t)} \tag{12}$$

We can plug in to (8) to get the desired result. □

We can extend the above theorem to include a margin as well.

Theorem 2.2. If we define $\overline{F} = \frac{F_T}{\sum_{t=1}^{T}\alpha_t} = \frac{\sum_t \alpha_t f_t}{\sum_t \alpha_t} \in co(\mathcal{G})$ (like ℓ_1 normalization) then

$$\hat{P}\left(\overline{F}(X) \leq \gamma\right) \leq \prod_t 2\sqrt{\epsilon_t^{1-\gamma}(1-\epsilon_t)^{1+\gamma}}$$

and if $\epsilon_t \leq \frac{1}{2} - 2\gamma \, \forall t$, then this decreases exponentially fast.

We can think of the first theorem (in the previous subsection) as saying: for all D_t, there exists $f_t \in \mathcal{G}$ with weighted empirical risk less than $1/2 - \gamma$, then $\exists \overline{F} \in co(\mathcal{G})$ with $\hat{P}\left(Y\overline{F}(X) \leq 0\right)$. The second theorem replaces the zero in the empirical probability with $\gamma/2$.

The converse result has a similar flavor: if $\exists \overline{F} \in co(\mathcal{G})$ margin better than γ, then we have $\epsilon_t \leq 1/2 - \gamma$.

Below we examine the converse:

Theorem 2.3. If, for $(x_1, y_1), \ldots, (x_n, y_n)$, $\exists F \in co(\mathcal{G})$ with $y_i F(x_i) > \gamma \, \forall i$, then for all probability distributions D on $\{1, \ldots, n\}, \exists f \in \mathcal{G}$ such that

$$\sum D(i) 1\left[y_i \neq f(x_i)\right] \leq \frac{1-\gamma}{2}$$

Proof. We proceed with the probabilistic method:

Suppose $F = \sum_t \alpha_t f_t$ with α_t as convex coefficients. Choose f randomly according to distribution given by $P(f = f_t) = \alpha_t$. Then

$$\begin{aligned}
0 &\leq \mathbb{E}\left[\sum_i D(i) 1[y_i = f(x_i)]\right] && (13)\\
&= \sum_t \alpha_t \sum_i D(i) 1[y_i \neq f_t(x_i)] && (14)\\
&= \sum_i D(i) \sum_t \alpha_t \frac{1 - y_i f_t(x_i)}{2} && (15)\\
&= \frac{1}{2}\left(1 - \sum_i D(i) \sum_t \alpha_t f_t(x_i)\right) && \leq \frac{1}{2}(1-\gamma) \quad (16)
\end{aligned}$$

□

2.2 Another interpretation: gradient descent

From last time, we know $\hat{\mathbb{E}} \exp(-Y F_T(X)) = \frac{1}{n} \sum_i \frac{D_{T+1}(i)}{D_1(i)} \prod_t Z_t$. Recall also that $\frac{1}{n} \exp(-y_i F_{T-1}(x_i)) = D_T(i) \prod_{t=1}^{T-1} Z_t$.

Observation: Choosing f_t to minimize $\epsilon_t = \sum_{i=1}^n D_t(i) 1[y_i \neq f_t(x_i)]$ and setting $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ is equivalent to choosing α_t, f_t to minimize

$$\hat{\mathbb{E}} \exp(-YF_t(X)) = \frac{1}{n}\sum_{i=1}^{n} \exp(-y_i(F_{t-1}(x_i) + \alpha_t f_t(x_i))) \tag{17}$$

$$\hat{\mathbb{E}} \exp(-YF_t(X)) = \frac{1}{n}\sum_{i=1}^{n} \left[(e^{\alpha_t} - e^{-\alpha_t})1[y_i \neq f_t(x_i) + e^{\alpha_t}]\right] e^{-y_i F_{t-1}(x_i)} \tag{18}$$

$$= (e^{\alpha_t} - e^{-\alpha_t}) \prod_{s=1}^{t-1} Z_s \sum_{i=1}^{n} D_t(i) 1[y_i \neq f_t(x_i)] + \frac{e^{-\alpha_t}}{n} \sum_{i=1}^{n} e^{-y_i F_{t-1}(x_i)} \tag{19}$$

Where the last equality holds from noting that $\frac{1}{n} e^{-y_i F_{t-1}(x_i)}$ is the weighting term recalled above. We also see that $\forall \alpha_t$, the best choice of f_t minimizes the first summation term above.

Given f_t, we can take a partial derivative with respect to α_t and set it equal to zero to find

$$\sum_{i:y_i \neq f_t(x_i)} \left(\frac{1}{n} e^{-y_i F_{t-1}(x_i)}\right) e^{\alpha_t} - \sum_{i:y_i = f_t(x_i)} \left(\frac{1}{n} e^{-y_i F_{t-1}(x_i)}\right) e^{-\alpha_t} = 0 \tag{20}$$

$$\left(\epsilon_t e^{\alpha_t} - (1-\epsilon_t) e^{-\alpha_t}\right) \prod_{s=1}^{t-1} Z_s = 0 \tag{21}$$

Which implies $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

So this is like a coordinate descent along the α_t with the objective of $\min \frac{1}{n} \sum e^{-y_i \sum_t \alpha_t f_t(x_i)}$.

3 An alternative formulation

We can create a more general interpretation with other cost functions than the exponential:

$$\min_F J(F) = \hat{\mathbb{E}} \phi(YF(X)) = \hat{\mathbb{E}} \left[\phi(Y(F_{t-1}(X) + \alpha_t f_t(X)))\right]$$

Gradient descent would be to choose a direction $v = (\alpha_t f_t(x_i), \ldots, \alpha_t f_t(x_n))$ to minimize $v' \nabla_z J_n(F_{t-1} + z)$, i.e. choose a direction from restricted options.

$$v \text{ minimizes } \sum v_i y_i \phi'(y_i F_{t-1}(x_i)) \tag{22}$$

$$\Leftrightarrow \min \sum (-v_i y_i)(-\phi'(y_i F_{t-1}(x_i))) \tag{23}$$

$$\Leftrightarrow \min 1[v_i \neq y_i] D_t(i) \tag{24}$$

With $D_t(i) = \frac{-\phi'(y_i F_{t-1}(x_i))}{Z_t}$ and Z_t is a normalization term.

CS281B/Stat241B (Spring 2008) Statistical Learning Theory **Lecture: 12**

Ada Boost, Risk Bounds, Concentration Inequalities

Lecturer: Peter Bartlett *Scribe: Subhransu Maji*

1 AdaBoost and Estimates of Conditional Probabilities

We continue with our discussion on AdaBoost and derive risk bounds of the classifier. Recall that for a function f, we have the following relationship between the expected excess risk and the excess ϕ approximation risk for a loss function ϕ,

$$\Psi\left(R(f) - R^*\right) \leq R_\phi(f) - R_\phi^*$$

where, R^* is the optimal Bayes Risk, R_ϕ^* is the risk of the optimal f i.e.$R_\phi^* = \inf_f R_\phi^*(f)$ and $H(\eta)$ is the function

$$H(\eta) = \inf_{\alpha \in R} \left[\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\right]$$

and $\Psi(\theta)$ is the function

$$\begin{aligned}\Psi(\theta) &= H^-\left(\frac{1+\theta}{2}\right) + H\left(\frac{1+\theta}{2}\right) \\ &= \phi(0) - \inf_{\alpha \in R}\left[\frac{1+\theta}{2}\phi(\alpha) + \frac{1-\theta}{2}\phi(-\alpha)\right]\end{aligned}$$

In the context of AdaBoost the loss function $\phi(\alpha) = e^{-\alpha}$ is convex and classification calibrated. Thus,

$$H(\eta) = \inf_{\alpha \in R}\left[\eta e^{-\alpha} + (1-\eta)e^{\alpha}\right]$$

Differentiating w.r.t. α and setting to zero gives us the optimal

$$\alpha(\eta) = \ln\sqrt{\frac{\eta}{1-\eta}}$$

This suggests that if we could choose $f(x)$ separately for each x, it would be a monotonically transformed version of conditional probability(see next section). Plugging this α^* into H yields

$$H(\eta) = 2\sqrt{\eta(1-\eta)},$$

which is concave and symmetric around 1/2. Then $\Psi(\theta)$ simplifies to

$$\Psi(\theta) = 1 - \sqrt{(1+\theta)(1-\theta)} = 1 - \sqrt{1-\theta^2}.$$

Finally, plugging this in to the original inequality yields

$$1 - \sqrt{1 - (R(f) - R^*)^2} \leq R_\phi(f) - R_\phi^*.$$

Examining the Taylor series of the left side about 0 shows that this is equivalent, for some constant c, to

$$R(f) - R^* \leq c\sqrt{(R_\phi(f) - R_\phi^*)}$$

when the excess ϕ-risk is sufficiently small. Thus, driving the excess ϕ-risk to zero will drive the discrete loss to zero as well, which justifies AdaBoost's use of this particular convex loss function.

2 Relationship to logistic regression

It turns out that we can interpret the value of $F(x)$ (where F is the boosted classifier returned by AdaBoost) as a transformed estimate of $\Pr(Y = 1|X = x)$. Consider a logistic model where

$$\Pr(Y = 1|X = x) = \frac{1}{1 + e^{-2f(x)}} = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}},$$

a rescaled version of the logistic function. In this model, the log loss (negative log likelihood) takes the form

$$\begin{aligned} -\ln \prod_{i=1}^{n} \Pr(Y = y_i|X = x_i) &= -\sum_{y_i=1} \ln \frac{1}{1 + e^{-2f(x)}} - \sum_{y_i=-1} \ln \left(1 - \frac{1}{1 + e^{-2f(x)}}\right) \\ &= \sum_{y_i=1} \ln \left(1 + e^{-2f(x)}\right) + \sum_{y_i=-1} \ln \left(1 + e^{2f(x)}\right) \\ &= \sum_{i=1}^{n} \ln \left(1 + e^{-2y_i f(x_i)}\right). \end{aligned}$$

Thus, the maximum likelihood logistic regression solution attempts to minimize the sample average of $\phi(\alpha) = \ln\left(1 + e^{-2\alpha}\right)$. This is closely related to AdaBoost, which minimizes the sample average of $\phi(\alpha) = e^{-\alpha}$. To see the connection, note that the first few terms of the Taylor expansion of $\ln\left(1 + e^{-2\alpha}\right) + 1 - \ln 2$ about 0,

$$1 - \alpha + \frac{\alpha^2}{2} - \ldots,$$

are identical to those of $e^{-\alpha}$.

While the two functions are very similar near zero, their asymptotic behavior is very different. In general we have that

$$\ln\left(1 + e^{-2\alpha}\right) \leq e^{-\alpha};$$

furthermore, the former grows linearly as α approaches $-\infty$, whereas the latter grows exponentially. Thus, we can view AdaBoost as approximating the maximum likelihood logistic regression solution, except with (sometimes exponentially) larger penalties for mistakes. A further similarity between the methods is that the α^* for $\phi(\alpha) = \ln\left(1 + e^{-2\alpha}\right)$ is the same as for AdaBoost.

3 Risk Bounds and Uniform Convergence

So far, we've looked at algorithms (including AdaBoost) that optimize over a set of training samples:

$$\min_{f \in F} \hat{R}(f) = \hat{\mathbb{E}} l(y, f(x)) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)).$$

If the empirical minimizer is $\hat{f}$, we are interested in bounding the true loss $R(\hat{f}) = \mathbb{E} l(y, \hat{f}(x))$ under this function. In particular, we hope that $\hat{R}(\hat{f})$ will converge to $\inf_{f \in F} R(f)$ as $n \to \infty$

For the (trivial) case where our function class F contains only a single function, we can simply appeal to the law of large numbers. For example, in the case of discrete loss, the Chernoff bound gives an upper bound on $\Pr(|\hat{R}(f) - R(f)| > \epsilon)$ that shrinks exponentially in n for any given ϵ.

This argument, however, fails when F is not a singleton. We cannot simply apply the law of large numbers to each $f \in F$ and then argue that the desired property holds when minimizing over all of F. The problem

is that we are considering $R(\text{argmin}_{f\in F}\hat{R}(f))$, where the inner part depends on the data. In particular, if F is such that for any n and data set there are functions $f \in F$ with small $\hat{R}(f)$ but large $R(f)$, then choosing an f that minimizes $\hat{R}(f)$ may not tend to minimize $R(f)$.

Example. Let $F = F_+ \cup F_-$ with

$$F_+ = \{x \mapsto f(x) : |\{x : f(x) = +1\}| < \infty\}$$
$$F_- = \{x \mapsto f(x) : |\{x : f(x) = -1\}| < \infty\}$$

Note that for any finite sequence, we can choose f from either F_+ or F_- to explain it. Now, suppose we have a distribution P such that $P(Y = 1|X) = 0.95$ almost surely, and for all x, $P(X = x) = 0$. Then, we have

$$f \in F_+ \Rightarrow R(f) = 0.05 = R^*$$
$$f \in F_- \Rightarrow R(f) = 0.95 > R^*$$

where, R^* is the Bayes risk. However, for any finite sample there is an $f \in F_+$ with $R(f) = 0$ but $R(f) - R^* = 0.9$. So, choosing a function from a class via empirical risk minimization does not guarantee risk minimization with such a rich class. Restated,

$$R(\text{argmin}_{f\in F}R(f)) \neq \inf_{f\in F} R(f)$$

Example. If the set of functions $F \in \{+1, -1\}^X$ is finite, then we *can* say something about the true risk given that the empirical risk is zero. The following theorem makes this explicit.

Theorem 3.1.

$$\Pr(\exists f \in F \text{ and } \hat{R}(f) = 0 \ \&\ R(f) \geq \epsilon) \leq |F|e^{-\epsilon n}$$

i.e., with probability at least $1 - \delta$,

$$\text{if, } \hat{R}(f) = 0, \text{ then } R(f) \leq \frac{\log|F|}{n} + \frac{\log 1/\delta}{n}$$

Proof. To show this we use the properties of the exponential functions and union bounds. For any $f \in F$ with $R(f) \geq \epsilon$, we have

$$\begin{aligned}\Pr(\hat{R}(f) = 0) &\leq (1-\epsilon)^n \\ &= \exp(n\log(1-\epsilon)) \\ &\leq \exp(-n\epsilon) \end{aligned} \tag{1}$$

Using the union bound (Boole's inequality: the probability of a union of events is no more than the sum of their probabilities), we have

$$\begin{aligned}\Pr\left(\bigcup\{f \in F : R(f) \geq \epsilon \ \&\ \hat{R}(f) = 0\}\right) &\leq \sum_{f\in F} \Pr\left(R(f) \geq \epsilon \ \&\ \hat{R}(f) = 0\right) \\ &\leq |F|e^{-\epsilon n} \end{aligned} \tag{2}$$

□

Example. (Decision Trees) Consider the class of decision trees of finite number of nodes N over $x \in \{+1, -1\}^d$. Thus $|F| \leq (d + 2)^N$, because we can specify the tree by listing, in breadth-first order, the N nodes of the tree, and each can be either one of the covariates or outputs $\{+1, -1\}$. Thus, if $\hat{R}(f) = 0$, then with probability $\geq 1 - \delta$,

$$R(f) \leq \frac{N\log(d+2)}{n} + \frac{\log 1/\delta}{n}$$

Example. F is parameterized using N bits, i.e. $F = \{x \mapsto \phi(x,b), b \in \{0,1\}^N\}$ with $f_b(x) = \phi(x,b)$. $|F| = 2^N$ and thus if $\hat{R}(f) = 0$, then with probability $\geq 1-\delta$,

$$R(f) \leq \frac{N}{n} + \frac{\log 1/\delta}{n}$$

Typically when we learn classifiers on training data the empirical risk is small but not zero and the above theorem can not be applied directly. In the next few sections we will be developing tools to show properties relating the empirical risk minimizer and the minimal risk.

4 Concentration Inequalities

We will be interested not only in whether $R(\arg\min_{f\in F}\hat{R}(f)) \to \inf_{f\in F} R(f)$, but how fast this convergence happens, called the rate of convergence.

4.1 Classic bounds

For this, several classic inequalities are useful that impose upper bounds on the total probability mass contained within the tail of a distribution.

Theorem 4.1. (Markov's Inequality) If $X \geq 0$ a.s. and $t > 0$, then $\Pr(X \geq t) \leq \frac{\mathbb{E}(X)}{t}$

Proof. $\mathbb{E}X \geq \mathbb{E}[X1(X \geq t)] \geq t\Pr(X \geq t) + 0\Pr(X < t) = t\Pr(X \geq t)$ □

Theorem 4.2. (Chebyshev's Inequality) If $t > 0$, then $\Pr(|X - \mathbb{E}X| \geq t)) \leq \frac{Var(X)}{t^2}$

Proof. Apply Markov's inequality to $(X - \mathbb{E}X)^2$ □

These upper bounds are not necessarily tight as seen in the following example

Example. Let $Z_i = \{0,1\}$ be i.i.d with $\Pr(Z_i = 1) = p$. Denote $S_n = \sum_i^n z_i$, then using Chebyshev's inequality on the variable S_n/n we have

$$\Pr\left(\left|\frac{S_n}{n} - \frac{\mathbb{E}S_n}{n}\right| > t\right) \leq \frac{Var(S_n/n)}{t^2} = \frac{p(1-p)}{nt^2}$$

On the other hand, the central limit theorem says

$$\sqrt{\frac{n}{\sigma^2}}\left(\frac{S_n}{n} - p\right) \to N(0,1)$$

Thus,

$$\lim_{n\to\infty}\Pr\left(\sqrt{\frac{n}{\sigma^2}}\left(\frac{S_n}{n} - p\right)\right) = 1 - \Phi(t) \leq \frac{c}{t}\exp\frac{-t^2}{2}$$

where $\Phi(t)$ is the cumulative distribution function of $N(0,1)$. So, $\Pr(\frac{S_n}{n} - p \geq \epsilon)$ should decrease as $exp\left(\frac{-\epsilon^2 n}{2\sigma^2}\right)$, which is much faster than the rate implied by Chebyshev's inequality.

4.2 Hoeffding's Inequality

We can show concentration inequalities for sums of independent random variables more generally. Note that the following bounds leverage independence, but don't require identical distributions among the variables involved.

Theorem 4.3. (Hoeffding's Inequality) Consider independent $X_i \in [a_i, b_i]$ and their sum, $S_n = \sum_{i=1}^n X_i$. Then,

$$\Pr(S_n - \mathbb{E}S_n \geq t) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

Proof. A monotonic transformation and exponentiation using $s > 0$, gives us a positive random variable. Applying Markov's inequality we get,

$$\begin{aligned}
\Pr(S_n - \mathbb{E}S_n \geq t) &= \Pr(e^{s(S_n - \mathbb{E}S_n)} \geq e^{st}) \\
&\leq e^{-st}\mathbb{E}\left[e^{s(S_n - \mathbb{E}S_n)}\right] \text{ (Markov's Inequality)} \\
&= e^{-st}\mathbb{E}\left[\exp(s(\sum_{i=1}^n (X_i - \mathbb{E}X_i)))\right] \\
&= e^{-st}\mathbb{E}\prod_{i=1}^n \left[e^{s(X_i - \mathbb{E}X_i)}\right]
\end{aligned} \tag{3}$$

where, the last inequality uses the independence of the variables. We will see in the next lecture a bound on the last inequality, which will complete the proof.

Example. Let $X_i = \{0, 1\}$ denote the loss on the $i'th$ example. Then, $S_n = n\hat{R}(f)$ and $\mathbb{E}S_n = nR(f)$. Applying Hoeffding's inequality we get

$$P(|\hat{R}(f) - R(f)| > \epsilon) \leq 2\exp\left(\frac{-2\epsilon^2 n^2}{\sum_{i=1}^n 1^2}\right) = 2\exp(-2n\epsilon^2)$$

Note that though the rate is right and this bound is tighter than Markov's, there is still a factor of σ^2 missing compared to the bounds one would expect from central limit theorem. □