CSE522, Winter 2011, Learning Theory Lecture 1 and 2 - 01/04/2011, 01/06/2011

Introduction and Models

Lecturer: Ofer Dekel *Scribe: Jessica Chang*

Machine learning algorithms have emerged as the dominant and ubiquitous tool in computer science. From computational biology to search engine problems, these algorithms are being applied to the whole gamut of real-world problems, and the fundamental underlying theory behind these tools must be understood to fully harness the power of their design and application. Via breakthroughs in learning theory, which builds from results in mathematics and theoretical computer science, we are only just beginning to witness payoff in other fields.

1 Prediction Problems

In this course, we will work mostly in the *prediction problem* framework, which captures many common machine learning problems. The goal of a prediction problem is to give the correct *label* (e.g. prediction or output) to an *instance* (e.g. context or input). For example:

- **search engine revenue:** search engines receive queries and want to predict the revenue made from (ads displayed for) that query
- **objection recognition:** USPS scans a bitmap of a zipcode and wishes to predict the digits of the zip code
- **decision problems:** a robot is given some state of the world and needs to predict the next best action.

As a first naive option, one could solve prediction problems via hard-coded logic (expert systems): for every possible instance, output the correct label. However, humans are better at providing examples than finding the actual solution. Machine learning algorithms leverage this strength by learning from provided examples. The general prediction paradigm is then

- find a representative set of m instances of the problem, $u_1, \ldots, u_m$
- human (the "teacher") provides the correct labels $y_1, \ldots, y_m$
- each (u_i, y_i) pair is a "labeled example"
- ML algorithm attempts to identify the simple *hypothesis* which explains the relationship between the inputs and outputs

The notions of "representative" and "simple" are important and we will discuss their significance more formally. This latter approach lends itself well to *generalization*, which is the idea of looking at m examples, identifying a hypothesis, and applying it to the next example. Notice, in stark contrast, that the task of *memorization*, i.e. anti-learning, is more trivial. The hypothesis

$$h(u) = \begin{cases} y_m & \text{if } u = u_m \\ 0 & \text{otherwise} \end{cases}$$

perfectly fits the m examples, but does not generalize. Hypothesis $h(u)$ can encapsulate any relationship and is not simple; this is not learning.

2 What is Learning?

This question has been asked by natural scientists and philosophers alike long before the birth of Computer Science as a field. The following list highlights a few notable perspectives on the notion of "learning" that have developed over the centuries.

- Occam's Razor (14th century): "entities must not be multiplied beyond necessity," i.e., all things being equal, tend toward the simplest explanation. This intuitive perspective is also referred to as "regularization".
- Karl Popper (20th century): "we prefer simpler hypotheses because they are easier to test/falsify." For example, suppose you are sitting in a field and observe three times that after a cow moos, her calf runs to her. You might be inclined to conclude that when animals make noise, their children run to them. However, as you are walking home, you observe that a dog barks and that her puppy does not run to her. Your initial conclusion has been easily falsified.
- **Underdeterminism.** Descartes (17th century) asked, "We only see a finite amount of examples, so how can we choose between two different hypotheses?"
- Swinburne (20th century): "infinitely many hypotheses fit the data, and we have to choose something, so choose the simplest one."

It is clear that people have sought answers to this question for centuries. We are now on the cusp of finding rigorous answers to some of these questions. Intuitively, to truly understand goes beyond the ability to explain something that has already been seen; understanding is closely tied to the capacity to predict what has yet to be observed. In other words, understanding is closely related to generalization.

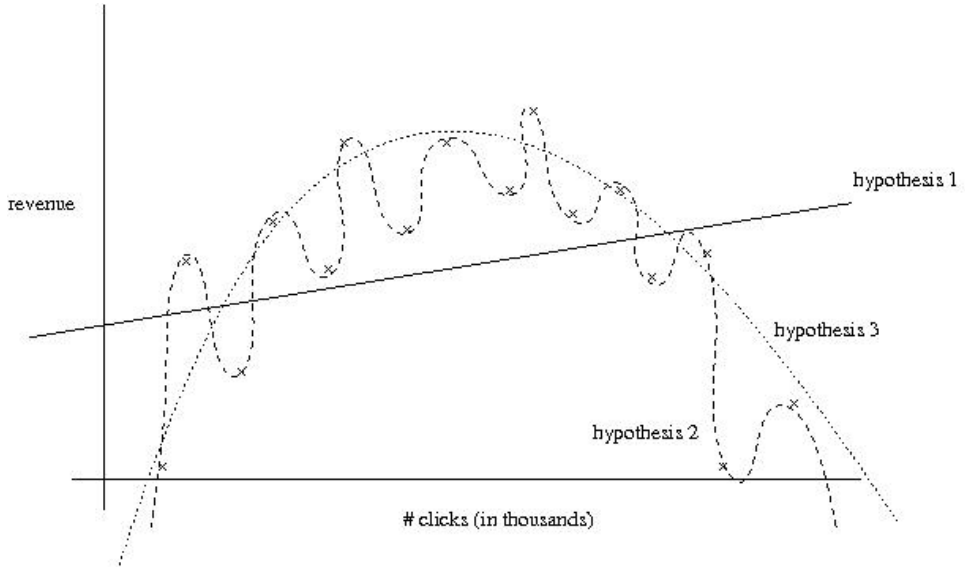

Figure 1: Candidate hypotheses for examples.

Let's revisit our search engine example. Consider the three candidate hypotheses for a set of examples, as depicted in Figure 1. Hypothesis 1, the best linear fit, is too coarse to yield any insightful relationship. On the other hand, hypothesis 2, which is the best fit for the examples, is underdetermined, arbitrary, and also not insightful for making future predictions. Hypothesis 3 is probably the best hypothesis.

How do we achieve generalization? The following list is a high-level preview of concepts to be covered in this course. They demonstrate how learning theory has taken philosophical questions about learning forward.

1. **Simplicity yields generalization.** We will see this when we cover Rademacher theory and VC theory.
2. **Inability to explain noise yields generalization.** This will be seen in P.A.C.-Bayes analysis, structural risk minimization, and O2B.
3. **No surprise yields generalization.** See future notes on SRM and P.A.C.-Bayes analysis.
4. **Algorithmic stability yields generalization.** The learning algorithm should not be highly sensitive to individual points.
5. **No regret yields generalization.** This concept intersects with game theory.

3 Supervised Statistical Learning

In this section, we introduce the main components and corresponding notation of the (statistical) supervised learning problem, beginning with the concept of features.

Definition 1. *For any prediction model, feature extraction maps an input u via a feature mapping X to a feature vector $X(u)$. $X(u)$ is a vector of n features which are representative of the input. The feature space $\mathcal{X}$ is the space of all feasible feature vectors.*

Typically, $\mathcal{X} \subseteq \mathbb{R}^n$. In our search engine example, $X(u) \in \mathbb{R}^n$ would denote the feature vector of query u. The first coordinate x_1 might denote the number of query words; x_2 might be the indicator for whether the query contains the word "buy"; x_3 might be the fraction of words that appear is some dictionary; etc. How to properly extract features depends on the user. In this course, it will be assumed that the feature mapping is given.

Definition 2. *The label space $\mathcal{Y}$ is the domain of our predictions. When $\mathcal{Y}$ is defined to be finite, it is referred to as the label set.*

$\mathcal{Y}$ unless otherwise noted is $\mathbb{R}_+$. The label set can differentiate between several supervised learning problems, as can be seen in the following list of a few common problem types.

- regression problems: $\mathcal{Y} = \mathbb{R}$ or more specifically, $\mathcal{Y} = [0, 1]$ if labels are probabilities
- binary classification: $\mathcal{Y} = \{-1, +1\}$. A binary classifier is a hypothesis $h : \mathcal{X} \to \{-1, +1\}$.
- multiclass classification: $\mathcal{Y} = \{1, \ldots, k\}$
- multiclass multilabel problems: $\mathcal{Y} = 2^{\{1,\ldots,k\}}$.
- ranking problems: given a base set $\tilde{\mathcal{Y}}$ of ground elements, e.g. web pages, $\mathcal{Y}$ is the set of all permutations, e.g. possible rankings, of $\tilde{\mathcal{Y}}$.

Definition 3. *A hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is a mapping of inputs, represented via feature vectors, to outputs or predictions. The hypothesis space $\mathcal{H}$ is the set of all candidate hypotheses.*

For example, $\mathcal{H}$ might be defined as the space of linear functions $h_w(x)$ where $w \in \mathbb{R}^n$ and $h_w(x) = \langle x, w \rangle$. Alternatively, $\mathcal{H}$ might be the space of decision trees, i.e. a piece-wise constant function defined by the structure and labels of the tree; the labeled leaves of the tree comprise $\mathcal{Y}$. We will be interested in finding "good" hypotheses within $\mathcal{H}$.

It is natural to ask how well a given hypothesis h performs on a given labeled example (x, y).

Definition 4. *The loss function $\ell : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}_+$ measures the quality of a prediction on a given example. In other words, $\ell(h; (x, y))$ is the penalty for predicting $h(x)$ when the correct label is y.*

Common examples of loss functions:

- error indicator: $\ell(h; (x, y)) = \mathbb{I}_{\{h(x) \neq y\}}$. This is a natural definition for binary or multiclass classification problems in which all wrong predictions are equally wrong.
- squared error (L2 error): $\ell(h; (x, y)) = (h(x) - y)^2$
- negative log likelihood (for Bayesian Learning)

In general, the loss function can be arbitrary and should be defined by the user in a way that models the problem.

So far, our model says very little about the "correct labels" and where they come from. We need a general representation of our world/problem context. For statistical supervised learning problems, we address this issue by representing the world as a distribution.

Definition 5. *Let $\mathcal{D} : \mathcal{X} \times \mathcal{Y}$ be the probability distribution on the space of examples representing the world or context of the problem.*

In general, $\mathcal{D}$ can be arbitrary. It is not necessarily consistent or without noise and models the full complexity of the world. The standard assumption is that $\mathcal{D}$ exists but is unknown. Now we can pose the following more natural question: how good is a given hypothesis h on average? This motivates the concept of risk.

Definition 6. *The risk of hypothesis h is*

$$\ell(h; \mathcal{D}) = Exp_{(x,y)\sim\mathcal{D}}[\ell(h; (x, y))]$$

Risk is also refered to as "expected loss/error" or "generalization loss/error" in the literature. The goal of statistical supervised learning is to *minimize risk*. Since $\mathcal{D}$ is unknown, we have no way of measuring risk directly. Thus, it is also standard to make the (strong) assumption that we can sample from $\mathcal{D}$.

Definition 7. *Given examples $S = \{(x_i, y_i)\}_{i=1}^m$, the empirical risk is*

$$\ell(h; S) = \frac{1}{m} \sum_{i=1}^{m} \ell(h; (x_i, y_i))$$

Empirical risk is also known as "training loss/error" or "empirical loss/error". Empirical risk is measureable. When is it a good approximation of risk? For example, the memorization hypothesis h has empirical risk $\ell(h; S) = 0$, since it remembers the right outputs for instances of S, but this is no indication of h's true risk.

3.1 Empirical Risk Minimization Algorithm

Due to Vladimir Vapnik, the empirical risk minimization (ERM) algorithm is at the core of the vast majority of machine learning algorithms used today. Its premise is that minimizing empirical risk yields a good approximation to minimizing risk.

Definition 8. ***ERM algorithm.***

1. *Choose a hypothesis class H, which is the set of all candidate hypotheses.*
2. *Choose a loss function ℓ.*

3. *Sample a training set of examples $S \sim \mathcal{D}^m$, i.e. such that each example is sampled i.i.d. from $\mathcal{D}$.*

4. *Find $h = ERM(\mathcal{H}, \ell, S) = \min_{h \in \mathcal{H}} \ell(h; S)$.*

Despite the vast attention that Step 4 has received, the most important aspect of the ERM algorithm is in choosing the appropriate hypothesis class. That there is very little theory for choosing $\mathcal{H}$ is one of ERM's weaknesses. The third step is not imperative to the definition of ERM, only to the theorems that we will be proving next. There are techniques to relax the independence assumption, but for now, suppose that the samples are drawn i.i.d.

Example 1 Suppose that $S = \{(x_i, y_i)\}_{i=1}^m$, where all $x_i \in \mathbb{R}^n$ and all $y_i \in \mathbb{R}^+$, is such that $S \sim \mathcal{D}^m$. Let A be the matrix whose rows consist of the x_i's. Additionally, let $\mathcal{H} = \{h_w : \mathcal{X} \to \mathcal{Y} : h_w(\cdot) = \langle \cdot, w \rangle$ and $w \in \mathbb{R}^n\}$, i.e. $\mathcal{H}$ is the set of linear functions. If loss is measured via the squared error function, then we informally want a hypothesis h_w which, for every example (x, y) minimizes $\ell(h_w; (x, y)) = (h_w(x) - y)^2$. The ERM algorithm finds the hypothesis which is

$$\operatorname*{argmin}_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 = \operatorname*{argmin}_{w \in \mathbb{R}^n} \frac{1}{2} \|Aw - y\|_2^2$$

Suppose that $rank(A^T A) = n$. Since the objective function is convex, ERM finds the w^* for which the derivative $\nabla \frac{1}{2} \|Aw^* - y\|_2^2 = A^T(Aw^* - y)$ is zero, i.e. the w^* for which $A^T A w^* = A^T y$. By invertability of $A^T A$,

$$w^* = (A^T A)^{-1} A^T y$$

Note that if $rank(A^T A) < n$, then we can apply a regularization term $R(h)$ to help us get around the rank issue and simultaneously bias our solution toward "simpler" hypotheses. Regularized ERM finds

$$\operatorname*{argmin}_{w \in \mathbb{R}^n} \ell(h; S) + R(h) = \operatorname*{argmin}_{w \in \mathbb{R}^n} \frac{1}{2} \|Aw - y\|_2^2 - \frac{\lambda}{2} \|w\|_2^2$$

for λ being the regularization parameter. Then w^* should satisfy $(A^T A + \lambda I) w^* - A^T y = 0$, i.e.

$$w^* = (A^T A + \lambda I)^{-1} A^T y$$

4 Loss Estimation

The basic steps of learning are extensions of estimation; if we can't estimate, we can't learn. In this section, we aim to find a low-variance unbiased estimator of $\ell(h; \mathcal{D})$.

Definition 9. *A random variable Z is an unbiased estimator of a value z if $Exp[Z] = z$.*

Toward this end, we let a test set $S = \{(x_i, y_i)\}_{i=1}^m$ be such that $S \sim \mathcal{D}^m$.

4.1 Review of Probability Theory

Let Ω denote the sample space, e.g. the space of all test sets; $w \in \Omega$ is a set of m examples. Let $\mathcal{F}$ be a set of events. In this course, events will usually be defined to capture the notion of a "good" test set. A random variable $Z : \Omega \to \mathbb{R}$ allows us to treat a random object in Ω as a number. For example, the feature vector $X(u)$ is a random variable of input u. A random variable is "simple" if it takes on a finite number of non-negative values.

Definition 10. *The expectation of a simple random variable Z taking on values in $\{z_1, \ldots, z_k\}$ is*

$$Exp[Z] = \sum_{i=1}^{k} z_i Pr[Z = z_i]$$

In general, random variables need not be discrete. They need not even be continuous. The following definition generalizes the definitions of expectation for both discrete and continuous random variables. The interested reader is referred to the course references for more information.

Definition 11. *For general non-negative random variables Z,*

$$Exp[Z] = \sup_{Y \leq_{a.s.} Z} Exp[Y]$$

where Y is simple and is almost surely less than or equal to Z.

Definition 12. *The variance of a random variable Z measures the average deviation from the expectation.*

$$Var(Z) = Exp\left[(Z - Exp[Z])^2\right]$$

Sometimes it will be more useful to use the identity

$$\mathrm{Var}(Z) = \mathrm{Exp}\left[Z^2\right] - (\mathrm{Exp}[Z])^2$$

Understanding $\ell(h; S)$ as an estimator of risk

Now we can discuss loss estimation more rigorously. Let the random variable $Z_i = \ell(h; (x_i, y_i))$ where (x_i, y_i) is the ith example from the random test set $S \sim \mathcal{D}$. Then,

$$\mathrm{Exp}[\ell(h; S)] = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Exp}[\ell(h; (x_i, y_i))] = \ell(h; \mathcal{D})$$

since $\mathrm{Exp}[\ell(h; (x_i, y_i))] = \ell(h; \mathcal{D})$ for all $(x_i, y_i) \in S$. Therefore, when $S \sim \mathcal{D}^m$, empirical risk is an unbiased estimator for risk.

Toward understanding its variance, define $Z = \ell(h; S) = \frac{1}{m} \sum_{i=1}^{m} Z_i$. Since Z_i is independent from Z_j for all $i \neq j$, $Var(\sum_{i=1}^{m} Z_i) = \sum_{i=1}^{m} \mathrm{Var}(Z_i)$. Therefore,

$$\begin{aligned} \mathrm{Var}(Z) &= \mathrm{Var}(\frac{1}{m} \sum_{i=1}^{m} Z_i) \\ &= \frac{1}{m^2} \sum_{i=1}^{m} \mathrm{Var}(Z_i) \\ &= \frac{1}{m} \mathrm{Var}(Z_1) \end{aligned}$$

Note that as more examples are sampled, the variance becomes tighter.

Theorem 13 (Markov's inequality)**.** *For any random variable $X \geq 0$, for any $\epsilon > 0$, $Pr[X \geq \epsilon] \leq \frac{Exp[X]}{\epsilon}$.*

Theorem 14 (Chebyshev's inequality)**.** *For any random variable $X \geq 0$ such that $Var(X)$ exists, for any $\epsilon > 0$,*

$$Pr[|X - Exp[X]| \geq \epsilon] \leq \frac{Var(X)}{\epsilon^2}$$

Chebyshev's inequality and our observation about $\mathrm{Var}(Z)$ immediately imply the following.

Theorem 15. *Let $Z_1, \ldots, Z_m$ be i.i.d. random variables with $\sigma^2 = Var(Z_1) < \infty$ and $\mu = Exp[Z]$. Let $Z = \frac{1}{m}\sum_{i=1}^{m} Z_i$. Then, for every $\epsilon > 0$,*

$$Pr[|Z - \mu| > \epsilon] \leq \frac{\sigma^2}{m\epsilon^2}$$

One can interpret ϵ as being the accuracy parameter, m the sample size, and $\delta = \frac{\sigma^2}{m\epsilon^2}$ as being the level of confidence.

Corollary 16. *Let $Z_1, \ldots, Z_m$ be i.i.d. random variables with $\sigma^2 = Var(Z_1) < \infty$ and $\mu = Exp[Z]$. Let $Z = \frac{1}{m}\sum_{i=1}^{m} Z_i$. Then, for every $\delta > 0$,*

$$Pr\left[|Z - \mu| > \sqrt{\frac{\sigma^2}{m\delta}}\right] \leq \delta$$

Understanding the tradeoff between these three parameters will be a recurring theme throughout this course. The following table summarizes the interaction between them.

m	ϵ	δ
m	ϵ	$\frac{\sigma^2}{m\epsilon^2}$
m	$\sqrt{\frac{\sigma^2}{m\delta}}$	δ
$\frac{\sigma^2}{\delta\epsilon^2}$	ϵ	δ