STAT 538 Lecture 1

January 3, 2012

Statistical Prediction

©Marina Meilă

mmp@stat.washington.edu

This course has a major theme: **supervised learning** (that will also be called **prediction**) and a minor theme, **optimization**. For a brief overview of *statistical learning* that defines its various subfields and paradigms see Lecture 0 of STAT 535 at `www.stat.washington.edu/courses/stat535/fall11/handouts.html`. In brief, *unsupervised learning*, the focus of STAT535, is concerned with estimating [features of] $P(X)$ from a sample, while *supervisesd learning* is concerned with $P(Y|X)$, i.e predicting a variable given others.

1 Prediction problems by the type of output

In supervised learning, the problem is *predicting* the value of an **output** (or **response** – typically in regression, or **label** – typically in classification) variable Y from the values of some observed variables called **inputs** (or **predictors, features, attributes**) $(X_1, X_2, \ldots X_n) = X$. Typically we will consider that the input $X \in \mathbb{R}^n$.

Prediction problems are classified by the type of response $Y \in \mathcal{Y}$:

- *regression*: $Y \in \mathbb{R}$
- *binary classification*: $Y \in \{-1, +1\}$
- *multiway classification*: $Y \in \{y_1, \ldots y_m\}$ a finite set
- *ranking*: $Y \in \mathbb{S}_p$ the set of permutations of p objects
- *structured prediction* $Y \in \Omega_V$ the state space of a graphical model over a set of [discrete] variables V

Example 1 Regression. *Y is the proportion of high-school students who go to college from a given school in given year. X are school attributes like class size, amount of funding, curriculum (note that they aren't all naturally real valued), median income per family, and other inputs like the state of the economy, etc. Note also that $Y \in [0,1]$ here.*

Economic forecasts are another example of regression. Note that in this problem as well as in the previous, the Y in the previous period, if observed, could be used as a predictor variable for the next Y. This is typical of structured prediction problems.

Weather prediction is typically a regression problem, as winds, rainfall, temperatures are continuous-valued variables.

Predicting the box office totals of a movie. What should the inputs be?

Example 2 (Anomaly) detection. *is a binary classification problem. $Y \in \{\text{normal}, \text{abnormal}\}$. For instance, Y could be "HIV positive" vs "HIV negative" (which could be abbreviated as "+", "-") and the inputs X are concentration of various reagents and lymph cells in the blood.*

Anomaly detection is a problem also in artificial systems, as any device may be functioning normally or not. There are also more general detection problems, where the object detected is of scientific interest rather than an "alarm": detecting Gamma-ray bursts in astronomy, detecting meteorites in Antarctica (a robot collects rocks lying on the ice and determines if the rock is terrestrial or meteorite). More recently, detecting faces/cars/people in images or video streams has become automated.

Example 3 Multiway classification. *Handwritten digit classification: $Y \in \{0, 1, \ldots 9\}$ and X=black/white 64× 64 image of the digit.*

OCR (Optical character recognition). The task is to recognized printed characters automatically. X is again a B/W digital image, $Y \in \{a-z, A-Z, 0-9, "." , "," , \ldots\}$, or another character set (e.g. Chinese).

Example 4 Structured prediction. *Speech recognition. X is a segment of digitally recorded speech, Y is the word corresponding to it. Note that it is not trivial to* segment speech, *i.e to separate the speech segment that* *corresponds to a given word. These segments have different lengths too (and the length varies even when the same word is spoken).*

The classification problem is to associate to each segment X of speech the c9orresponding word. But one notices that the words are not indepedent of other neighboring words. In fact, people speak in sentences, so it is natural to recognize each word in dependence from the others. Thus, one imposes a graphical model structure on the words corresponding to an utterance $X^1, X^2, \ldots X^m$. For instance, the labels $Y^{1:m}$ could form a chain $Y^1 - Y^2 - \ldots Y^m$. *Other more complex graphical models structures can be used too.*

2 Predictors

A **predictor** is a [deterministic] function that associates to an input x a corresponding $\hat{y} = f(x)$. A predictor is a kind of model (not yet a statistical model, though), hence when we talk about the set of possible predictors for a problem we call it the **model class** $\mathcal{F}$.

We choose the "best" predictor in $\mathcal{F}$ for a particular task based on a **smaple** or (**training set**) of **labeled data** $\mathcal{D} = \{(x^1, y^1), (x^2, y^2), (x^N, y^N)\}$. The pairs (x^i, y^i) in the sample are called **examples**. In a binary classification problem we talk about **negative**, respectively **positive** examples referring to examples labeled $+$, respectively $-$. N is the **sample size**.

One can classify prediction methods by the type of predictor (i.e. by the type of model class $\mathcal{F}$).

Example 5 The linear predictor

$$f(x) \;=\; \beta^T x \tag{1}$$

where $Y \in \mathbb{R}$, $X \in \mathbb{R}^n$ *and* $\beta \in \mathbb{R}^n$ *is a* **vector of parameters**. *The model class is* $\mathcal{F} = \{\beta \in \mathbb{R}^n\}$ *the set of all linear functions over* $\mathbb{R}^n$.

The linear predictor can be used for [binary] classification as well

$$f(x) \;=\; \mathrm{sgn} \beta^T x \tag{2}$$

The above classifier is closely related to **logistic** [**linear**] **regression**, *where we model*

$$\frac{P(Y=1|X)}{P(Y=-1|X)} = \beta^T X \tag{3}$$

Example 6 The Nearest-Neigbor predictor *This is a non-parametric predictor well suited for multiway classification. The label of a point x is assigned as follows: (1) we find the example x^i that is nearest to x (in Euclidean distance), (2) we assign x the label y^i.*

There are many possible extensions to the nearest neighbor classifier; for instance, one could find the K nearest neighbors of x, and set $f(x)$ to be the label that appears most frequently among the K neighbors. Ties are possible; hence, for binary classification, it is practical to make K an odd number.

Example 7 Classification and regression trees. *A* **classification tree** *or (***decision tree***) is built recursively by splitting the data with hyperplanes parallel to the coordinate axes. At each split, the goal is to separate $+$ examples from $-$ examples as well as possible. Eventually, all the regions will be "pure", i.e. will contain examples from one class only. Classification trees can be used in multiway classification as well (there we try to create a pure region on at least one side of the split) and even with regression (there we try to obtain regions where the outputs are nearly the same).*

Decision regions For a classifier, the function $f(x)$ takes only a finite set of values. The region in X space where f takes value y is called the **decision region** associated to y. $D_y = \{x \in \mathbb{R}^n,\ f(x) = y\} = f^{-1}(y)$. The boundaries separating the decision regions are called **decision boundaries**. For a binary classifier, we have two decision regions, one associated to the value $+1$, the other associated to -1. It is assumed by convention that, in this case, $f(x) = 0$ on the decision boundary.

Sometime, classifiers are named after their decision boundaries: e.g. *linear classifier*, *quadratic classifier.*

Exercises Show that (i) the linear classifier in (2) has a linear decision boundary; (ii) decision boundary or the nearest neighbor classifier is a polygonal line; (iii) the generative classifier defined in (7) below has a quadratic decision boundary.

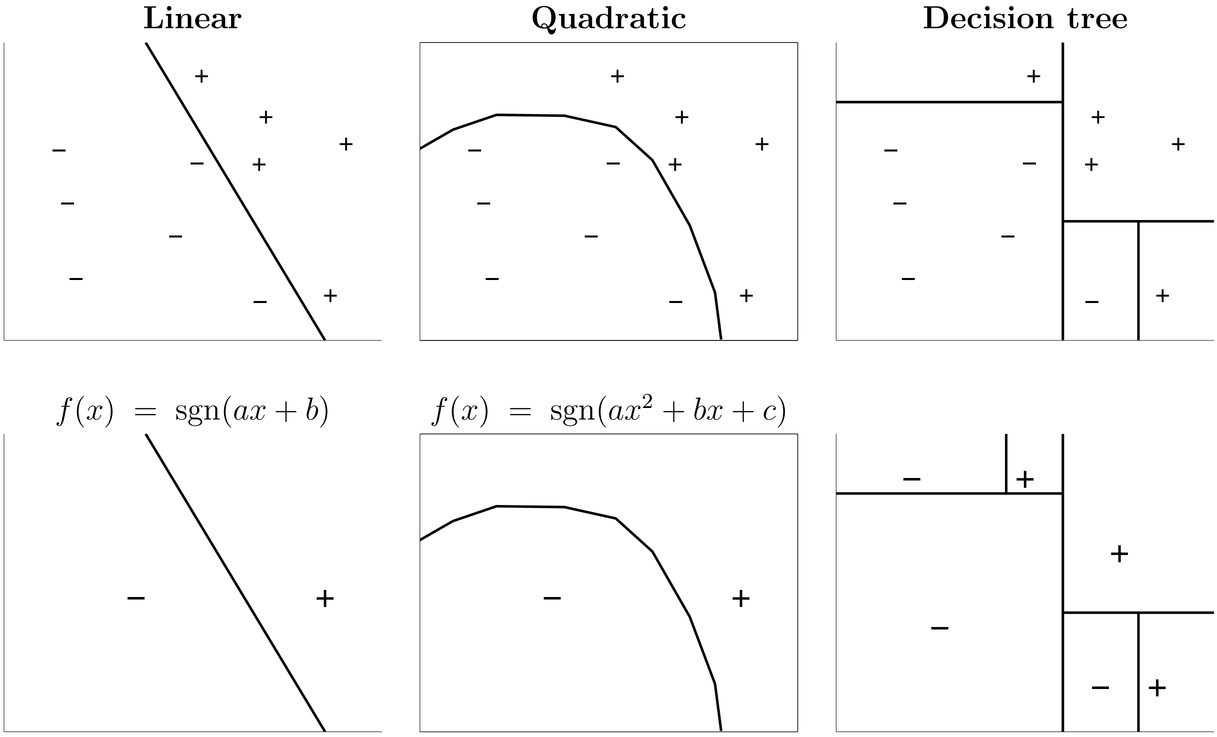

Linear
Quadratic
Decision tree
+
−
$f(x) \ = \ \mathrm{sgn}(ax+b)$
$f(x) \ = \ \mathrm{sgn}(ax^2+bx+c)$

Figure 1: Examples of classifiers and their decision regions

2.1 Generative models for classification

One way to define a classifier is to assume that each class is generated by a distribution $g_y(X) = P(X|Y = y)$. If we know the distributions g_y and the class probabilities $P(Y = y)$, we can derive the *posterior probability* distribution of Y for a given x. This is

$$P(Y = y|X) \;=\; \frac{P(Y = y)g_y(X)}{\sum_{y'} P(Y = y')g_{y'}(X)} \;=\; \frac{P(Y = y)g_y(X)}{P(X)} \tag{4}$$

The "best guess" for $Y(X)$ is

$$f(X) = \text{argmax}_y P(Y = y|x) = \text{argmax}_y P(Y = y)g_y(x) \tag{5}$$

The classification method above amounts to a likelihood ratio test for Y. The functions $g_y(x)$ are known as **generative models** for the classes y. Therefore, the resulting classifier is called a **generative classifier**. In contrast, a classifier defined directly in terms of $f(x)$, like the linear, quadratic, decision tree is called a **discriminative classifier**. In practice, we may not know the functions $g_y(x)$, in which case we estimate them from the sample $\mathcal{D}$.

Example 8 *Assume $Y = \pm 1$, $g_y(x) = N(x, \pm\mu, \sigma^2 I)$, i.e each class is generated by a Normal distribution with the same spherical covariance matrix, but with a different mean. Let $P(Y = 1) = p \in (0, 1)$. Then, the posterior probability of Y is*

$$P(Y = 1|x) \;\propto\; pe^{-||x-\mu||^2/(2\sigma^2)} \qquad P(Y = -1|x) \;\propto\; (1-p)e^{-||x+\mu||^2/(2\sigma^2)} \tag{6}$$

and $f(x) = 1$ iff $\ln P(Y = 1|x)/P(Y = -1|x) \geq 0$, i.e iff

$$\ln\frac{p}{1-p} - \frac{1}{2\sigma^2}[||x^2|| - 2\mu^T x + ||\mu||^2 - ||x^2|| - (2\mu)^T x - ||\mu||^2] \;=\; \frac{-2\mu^T}{\sigma^2}x + \ln\frac{p}{1-p} \geq 0 \tag{7}$$

Hence, the classifier $f(x)$ turns out to be a linear classifier. The decision boundary is perpendicular to the segment connecting the centers $\mu, -\mu$. This classifier is known as **Fisher's Linear Discriminant**. *[***Exercises*** Show that if the generative models are normal with different variances, then we obtain a quadratic classifier. What happens if the models g_y have the same variance, but it is a full covariance matrix Σ?]*

3 Loss functions

The **loss function** represents the cost of error in a prediction problem. We dentote it by L, where $L(y, f(x))$ is the cost of predicting $f(x)$ when the actual outcome is y[1].

The **Least squares** (or **quadratic**) loss function is given by

$$L(y, f(x)) \;=\; (y - f(x))^2 \tag{8}$$

This loss is commonly associated with regression problems. For classification, a natural loss function is **misclassification error** (also called **0-1 loss**)

$$L(y, f(x)) \;=\; \nVdash_{y \neq f(x)} \;=\; \begin{cases} 1 & \text{if}\, y \neq f(x) \\ 0 & \text{if}\, y = f(x) \end{cases} \tag{9}$$

Sometimes different errors have different costs. For instance, classifying a HIV+ patient as negative (**a false negative** error) incurs a much higher cost than classifying a normal patient as HIV+ (**false positive** error). This is expressed by **asymmetric misclaassification costs**. For instance, assume that a false positive has cost one and a false negative has cost 100. We can express this in the matrix

$f(x)$:	$+$	$-$
true :$+$	0	100
$-$	1	0

In general, when there are p classes, the matrix $L = [L_{kl}]$ defines the loss, with L_{kl} being the cost of misclassifying as l an example whose true class is k.

The objective of prediction is to minimize the expected cost on future data, $\min_{f \in \mathcal{F}} E_{P(X,Y)}[L(Y, f(X)]$; we denote the expected cost by $L(f)$. In particular, for the misclassification error, $L_{01}(f)$ is the probability of making an error on future data. [Prove this.] The following is a simple rewrite of this fact

$$L_{01}(f) \;=\; P[Yf(X) < 0] \;=\; E_{P_{XY}}[1_{[Yf(X)<0]}] \tag{10}$$

This objective cannot be optimized directly, because we don't know the data distribution P_{XY}. Therefore, in training classifiers, one uses the **empirical** data distribution given by the sample $\mathcal{D}$.

[1] Note that sometimes it is posible that the loss depends on x directly. Then we would write it as $L(y, \hat{y}, x)$ where $\hat{y} = f(x)$.

The **empirical loss** or **empirical error** or **training error** is the average loss on $\mathcal{D}$, i.e

$$\hat{L}(f) \;=\; \frac{1}{N}\sum_{i=1}^{N} 1_{[y^i f(x^i)<0]} \tag{11}$$

How small can the expected loss be? It is clear that $\min_{f\in\mathcal{F}} L(f) \;\geq\; \min_f L(f) \;=\; L^*$. The cost L^* is the absolute minimum loss for the given P_{XY} and it is called the **Bayes loss**. This loss is usually not zero.

Exercise What is the Bayes loss if (1) $P(Y|X) \;\sim\; N((\beta^*)^T X, \sigma^2 I)$ and the loss is L_{LS}; (2) $P(X|Y=\pm 1) \;\sim\; N(\mu_\pm, \sigma^2 I)$ and the loss is L_{01} (for simplicity, assume $X \in \mathbb{R}, \mu_{pm} = \pm 1$, $\sigma = 1$); (3) give a formula for the Bayes loss if we know $P(X|Y=\pm 1), P(Y)$, $Y \in \{\pm 1\}$ and the loss is L_{01}. (4) Give an example of a situation when the Bayes loss is 0.

Example 9 *Assume we know the data distribution, i.e. we know that* $P(X|Y = \pm 1) \;\sim\; N(\mu_\pm, \sigma^2 I)$ *and* $P(Y=1) = p$. *We will calculate the Bayes loss explicitly. From Example 8 we know the expression for* $P(Y|X)$. *The loss is* L_{01}*, so the probability of error if we choose* $Y = 1$ *for a given* x *is equal to* $P(Y = -1|x)$ *and this equals the expected* L_{01} *loss. Hence, the best* y *for a given* x *is the one that minimizes* $P(-Y|x)$. *The Bayes loss is equal to* $L_{01}^* = \int_{\mathbb{R}^n} \min_{y=\pm 1} P(y|x)P(x)dx = \int_{\mathbb{R}^n} \min_{y=\pm 1} g_y(x)P(y)dx = p\int_{D_-} g_+(x)dx + (1-p)\int_{D_+} g_+(x)dx$.

Some issues we will study. Now we have most of the elements in place to formulate some questions about our task.

- We could try to find $\hat{f} = \underset{f\in\mathcal{F}}{\operatorname{argmax}} \hat{L}(f)$? This is called the **empirical minimizer**. A first problem is how to find $\hat{f}$. For some problems, in particular for linear regression with least-squares loss, there is an analytical formula for $\hat{f}$.

  $$\hat{\beta} \;=\; (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{12}$$

  with $\mathbf{X}, \mathbf{Y}$ representing respectively the matrix with the inputs x^i as rows, and the vector of corresponding outputs (responses).

  On the other hand, the linear classification problem, even for the "simple" misclassification cost, has no analytic minimizer. One resorts to numerical, often iterative algorithms, for minimizing the misclassification cost, or other costs associated with classification. This why people call estimating a predictor "training".

Thus, one challenge we will take up in this course is to find minima of functions by numerical methods; this is the realm of **optimization**.

- A second question is whether minimizing the empirical loss is a "good" strategy for our stated objective (to have a low expected loss). This is a statistical question, and we will study it. The result is that sometimes it is better not to minimize the empirical loss perfectly. This is called the "bias-variance tradeoff" (explained in the next section). In particular, sometimes one minimizes

$$\hat{L}(f) + \lambda J(f), \quad \lambda \geq 0 \tag{13}$$

where the first term depends on the data, and the second term depends on properties of f alone. This term is called a **regularizer**. One can always cast the above optimization into a statistical estimation problem. The term that depends on the data is called (formally) the (negative) **log-likelihood**, while the term $\lambda J(f)$ is the (negative) **(log)-prior**. In this paradigm, the minimization in (13) represent a MAP (Maximum A-Posteriori Estimation). The "prior" $J(f)$ is typically favoring "simple" functions (more about this later). Forms of regularization have been in use in statistics for a long time, under the name **shrinkage**.

3.1 A comparison of generative and discriminative classifiers

Below is a list of the relative advantages of generative and discriminative classifiers.

Advantages of generative classifiers

- Generative classifiers are statistically motivated
- Generative classifiers are *asymptotically optimal*

Theorem 1 *If the model class G_y in which we are estimating g_y contains the true distributions $P(X|Y = y)$ for every y, and $g_y = P(X|Y), P(Y = y)$ are estimated by Maximum Likelihood then the expected loss of the* *generative classifier f_g given by (5) tends to the Bayes loss when $N \rightarrow \infty$, i.e* $\lim_{N\rightarrow\infty} L_{01}(f_g) \leq \min_{f\in\mathcal{F}} L_{01}(f)$*. Here $\mathcal{F}$ is the class of likelihood ratio classifiers obtainable from g_y's in $\mathcal{G}_\dagger$.*

- The log-likelihood ratio $\ln \frac{P(Y=1|x)}{P(Y=-1|x)}$ is a natural confidence measure for the label at $f_g(x)$. In other words, the further away from 0 the likelihood ratio, the more confident we should be that the chosen y is correct.
- Generative classifiers extend naturally to more than two classes. If the problem changes, and a new class appears, or the class distribution $P(Y)$ changes, updating the classifier is simple and computationally efficient. By contrast, representing/visualizing decision boundaries between more than two classes is tedious.
- Often it is easier to pick a (parametric) model class for g_y than directly for f. Generative models are generally more intuitive than discriminative models.

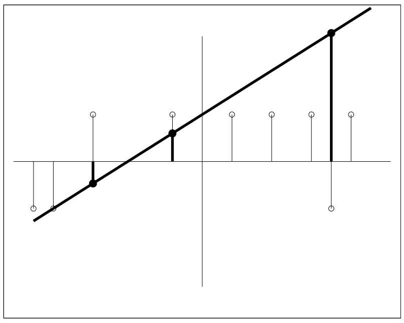
Advantages of discriminative classifiers

- Generative models offer no guarantees if the true g_y aren't in the chosen model class, whereas for many classes of discriminative models there are guarantees.
- Many discriminative models have performance guarantees for any sample size N, while generative models are only guaranteed for large enough N
- Discriminative classifiers offer many more choices (but one must know how to pick the right model)
- *The most important advantage:* Generative models do not use data optimally in the non-asymptotic regime (when $N \ll \infty$). This has been confirmed practically many times, as discriminative classifiers have been very successful for limited sample sizes

Confidence and margin. Sometimes we construct real-valued "classifiers"

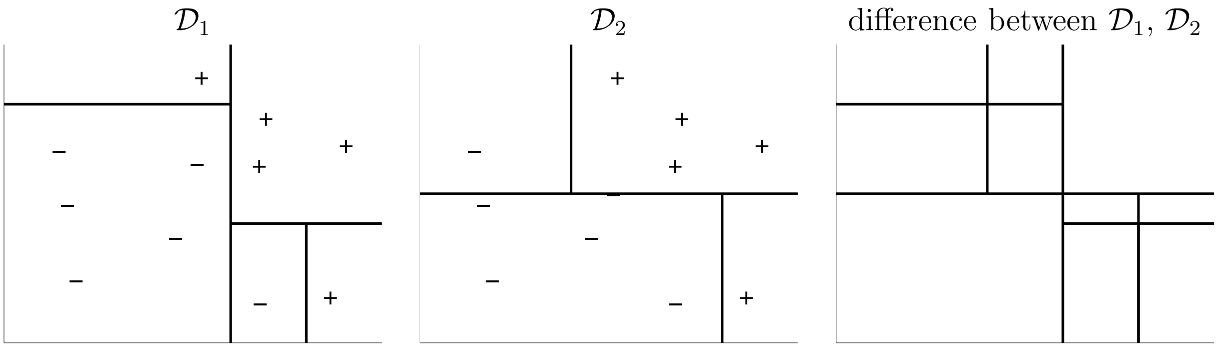
$$f : X \longrightarrow \mathbb{R}$$

Then, the label of a new example is $\hat{y} = \text{sgn} f(x)$ and $|f(x)|$ is the **confidence** of the classification. The **margin** is $yf(x)$ which should be > 0 for correct classification.



3.2 Variance and Bias

For a fixed learning algorithm, if a new data set is sampled ⇒ the learned f will be different. The **variance** measures the sensitivity of f to changes in the training set.



- variance decreases with N
- variance increases with complexity (high for overfitted models)

When the classifiers in $\mathcal{F}$ are too simple, they cannot fit the data well. This is called **bias**. Bias can be

- **deterministic (hard)**: no $f \in \mathcal{F}$ near optimal generalization error (in the case the classes are **separable**, no $f \in \mathcal{F}$ fits the data)
- **stochastic (soft)**: the prior of an $f \in \mathcal{F}$ with near optimal generalization error (i.e that fits the data if classes separable) is very small