COMS 4721: Machine Learning for Data Science

Lecture 7, 2/7/2017

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

CLASSIFICATION

Terminology and Notation

Input: As with regression, in a *classification problem* we start with measurements $x_1, \ldots, x_n$ in an input space $\mathcal{X}$. (Again think $\mathcal{X} = \mathbb{R}^d$)

Output: The *discrete* output space $\mathcal{Y}$ is composed of K possible *classes*:

- $\mathcal{Y} = \{-1, +1\}$ or $\{0, 1\}$ is called binary classification.
- $\mathcal{Y} = \{1, \ldots, K\}$ is called multiclass classification

Instead of a real-valued response, classification assigns x to a category.

- Regression: For pair (x, y), y is the response of x.
- Classification: For pair (x, y), y is the class of x.

Defining a classifier

Classification uses a function f (called a *classifier*) to map input x to class y.

$$y = f(x) \;:\; f \text{ takes in } x \in \mathcal{X} \text{ and declares its class to be } y \in \mathcal{Y}$$

As with regression, the problem is two-fold:

- Define the classifier f and its parameters.
- Learn the classification rule using a training set of "labeled data."

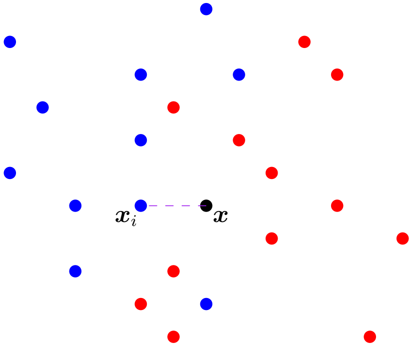
Nearest neighbor classifiers

Nearest neighbor (NN) classifier

Given data $(x_1, y_1), \ldots, (x_n, y_n)$, construct classifier $\hat{f}(x) \to y$ as follows:

For an input x not in the training data,

1. Let x_i be the point among $x_1, x_2, \ldots, x_n$ that is "closest" to x.
2. Return its label y_i.



DISTANCES

Question: How should we measure distance between points?

The default distance for data in $\mathbb{R}^d$ is the Euclidean one:

$$\|u - v\|_2 = \left(\sum_{i=1}^d (u_i - v_i)^2\right)^{\frac{1}{2}} \quad \text{(line-of-sight distance)}$$

But there are other options that may sometimes be better:

- ℓ_p for $p \in [1, \infty]$: $\|u - v\|_p = \left(\sum_{i=1}^d |u_i - v_i|^p\right)^{\frac{1}{p}}$.
- Edit distance (for strings): How many add/delete/substitutions are required to transform one string to the other.
- Correlation distance (for signal): Measures how correlated two vectors are for signal detection.

Example: OCR with NN classifier

- **Handwritten digits data**: grayscale 28×28 images, treated as vectors in $\mathbb{R}^{784}$, with labels indicating the digit they represent.

- Split into training set $\mathcal{S}$ (60K points) and testing set $\mathcal{T}$ (10K points).
- **Training error**: $\mathrm{err}(\hat{f}, \mathcal{S}) = 0 \;\leftarrow$ declare its class to be its own class!
  Test error: $\mathrm{err}(\hat{f}, \mathcal{T}) = 0.0309 \;\leftarrow$ using ℓ_2 distance
- Examples of mistakes: (left) test point, (right) nearest neighbor in $\mathcal{S}$:

- **Observation**: First mistake might have been avoided by looking at three nearest neighbors (whose labels are '8', '2', '2') …

test point three nearest neighbors

k-NEAREST NEIGHBORS CLASSIFIER

Given data $(x_1, y_1), \ldots, (x_n, y_n)$, construct the k-NN classifier as follows:

For a new input x,

1. Return the k points closest to x, indexed as $x_{i_1}, \ldots, x_{i_k}$.
2. Return the majority-vote of $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$.

(Break ties in both steps arbitrarily.)

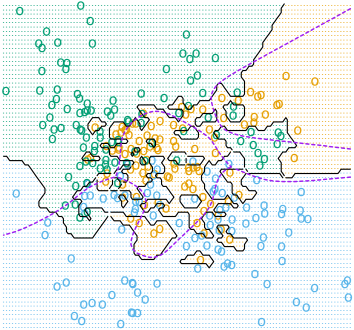
Example: OCR with k-NN classifier

k	1	3	5	7	9
$\mathrm{err}(\hat{f}_k, T)$	0.0309	0.0295	0.0312	0.0306	0.0341

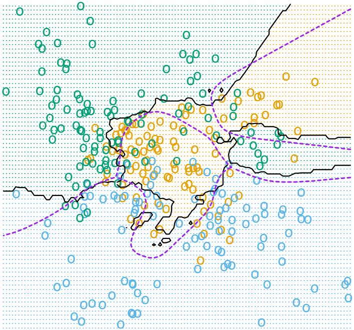
EFFECT OF k

In general:

- Smaller $k \Rightarrow$ smaller training error.
- Larger $k \Rightarrow$ predictions are more "stable" due to voting.



1-NN



15-NN

Purple dotted lines : Can ignore for now.

Black solid lines : k-NN's decision boundaries.

Statistical setting

How do we measure the quality of a classifier?

For any classifier we care about two sides of the same coin:

- Prediction accuracy: $P(f(x) = y)$.
- Prediction error: $\mathrm{err}(f) = P(f(x) \neq y)$.

To calculate these values, we assume there is a distribution $\mathcal{P}$ over the space of labeled examples generating the data

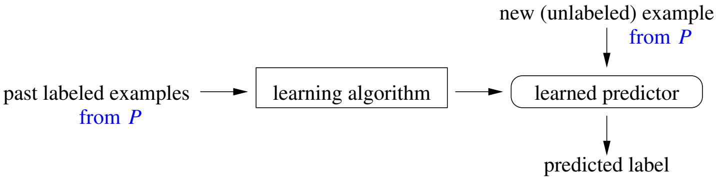
$$(x_i, y_i) \overset{iid}{\sim} \mathcal{P}, \qquad i = 1, \ldots, n.$$

We don't know what $\mathcal{P}$ is, but can still talk about it in abstract terms.

When is there any hope for finding an accurate classifier?

Key assumption: Data $(x_1, y_1), \ldots, (x_n, y_n)$ are i.i.d. random labeled examples with distribution $\mathcal{P}$.

This assumption allows us to say that the past should look like the future.



Regression makes similar assumptions.

BAYES CLASSIFIERS

Optimal classifiers

Can we talk about what an "optimal" classifier looks like?

Assume that $(X, Y) \overset{iid}{\sim} \mathcal{P}$. (Again, we don't know $\mathcal{P}$)

Some probability equalities with $\mathcal{P}$:

1. The expectation of an indicator of an event is the probability of the event, e.g.,

$$\mathbb{E}_P[\mathbb{1}(Y = 1)] = P(Y = 1), \quad \leftarrow \mathbb{1}(\cdot) = 0 \text{ or } 1 \text{ depending if } \cdot \text{ is true}$$

2. Conditional expectations can be random variables, and their expectations remove the randomness,

$$C = \mathbb{E}[\, A \,|\, B \,] : \quad A \text{ and } B \text{ are } \textit{both} \text{ random, so } C \text{ is random}$$

$$\mathbb{E}[\, C \,] = \mathbb{E}[\, \mathbb{E}[\, A \,|\, B \,] \,] = \mathbb{E}[\, A \,] \quad \text{"tower property" of expectation}$$

OPTIMAL CLASSIFIERS

For any classifier $f \colon \mathcal{X} \to \mathcal{Y}$, its prediction error is

$$P(f(X) \neq Y) = \mathbb{E}[\mathbb{1}(f(X) \neq Y)] = \mathbb{E}[\underbrace{\mathbb{E}[\mathbb{1}(f(X) \neq Y)\,|X]}_{\text{a random variable}}] \qquad (\dagger)$$

Optimal classifiers

For any classifier $f\colon \mathcal{X} \to \mathcal{Y}$, its prediction error is

$$P(f(X) \neq Y) = \mathbb{E}[\mathbb{1}(f(X) \neq Y)] = \mathbb{E}[\underbrace{\mathbb{E}[\mathbb{1}(f(X) \neq Y)\,|X]}_{\text{a random variable}}] \qquad (\dagger)$$

For each $x \in \mathcal{X}$,

$$\mathbb{E}[\mathbb{1}(f(X) \neq Y)\,|\,X = x] = \sum_{y \in \mathcal{Y}} P(Y = y\,|\,X = x) \cdot \mathbb{1}(f(x) \neq y), \qquad (\ddagger)$$

OPTIMAL CLASSIFIERS

For any classifier $f \colon \mathcal{X} \to \mathcal{Y}$, its prediction error is

$$P(f(X) \neq Y) = \mathbb{E}[\mathbb{1}(f(X) \neq Y)] = \mathbb{E}[\underbrace{\mathbb{E}[\mathbb{1}(f(X) \neq Y)\,|X]}_{\text{a random variable}}] \qquad (\dagger)$$

For each $x \in \mathcal{X}$,

$$\mathbb{E}[\mathbb{1}(f(X) \neq Y)\,|\,X = x] = \sum_{y \in \mathcal{Y}} P(Y = y\,|\,X = x) \cdot \mathbb{1}(f(x) \neq y), \qquad (\ddagger)$$

The above quantity ($\ddagger$) is minimized for this particular $x \in \mathcal{X}$ when

$$f(x) = \arg\max_{y \in \mathcal{Y}} \; P(Y = y | X = x). \qquad (\star)$$

The classifier f with property ($\star$) for all $x \in \mathcal{X}$ is called the *Bayes classifier*, and it has the smallest prediction error ($\dagger$) among *all classifiers*.

The Bayes classifier

Under the assumption $(X, Y) \overset{iid}{\sim} \mathcal{P}$, the optimal classifier is

$$f^\star(x) := \arg\max_{y \in \mathcal{Y}} \ P(Y = y | X = x).$$

From Bayes rule we equivalently have

$$f^\star(x) = \arg\max_{y \in \mathcal{Y}} \ \underbrace{P(Y = y)}_{\textit{class prior}} \times \ \underbrace{P(X = x | Y = y)}_{\textit{data likelihood} \,|\, \textit{class}}.$$

- $P(Y = y)$ is called the *class prior*.
- $P(X = x | Y = y)$ is called the *class conditional distribution* of X.
- In practice we don't know either of these, so we approximate them.

Aside: If X is a continuous-valued random variable, replace $P(X = x | Y = y)$ with *class conditional density* $p(x | Y = y)$.

Example: Gaussian class conditional densities

Suppose $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \{0, 1\}$, and the distribution $\mathcal{P}$ of (X, Y) is as follows.

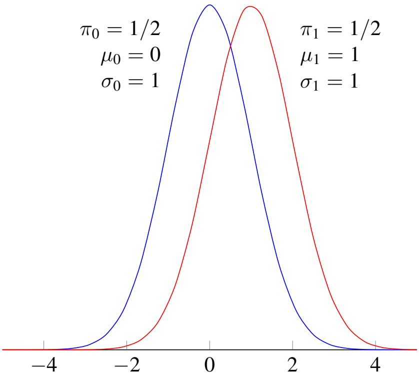
- **Class prior**: $P(Y = y) = \pi_y, \quad y \in \{0, 1\}$.
- **Class conditional density** for class $y \in \{0, 1\}$: $p_y(x) = N(x|\mu_y, \sigma_y^2)$.
- **Bayes classifier**:

$$
\begin{aligned}
f^\star(x) &= \operatorname*{argmax}_{y \in \{0,1\}} \; p(X = x|Y = y)P(Y = y) \\
&= \begin{cases} 1 & \text{if } \dfrac{\pi_1}{\sigma_1} \exp\left[-\dfrac{(x-\mu_1)^2}{2\sigma_1^2}\right] > \dfrac{\pi_0}{\sigma_0} \exp\left[-\dfrac{(x-\mu_0)^2}{2\sigma_0^2}\right] \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

This type of classifier is called a *generative* model.

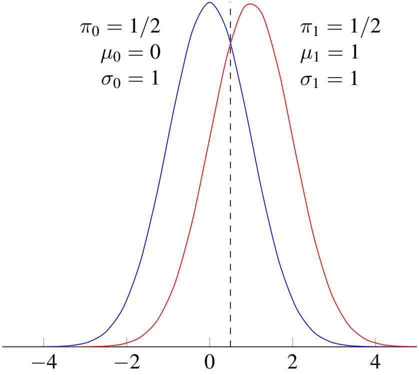
- **Generative model**: Model x and y with distributions.
- **Discriminative model**: Plug x into a distribution on y (used thus far).

EXAMPLE: GAUSSIAN CLASS CONDITIONAL DENSITIES



$1/2$ of x's from $N(0, 1) \rightarrow y = 0$
$1/2$ of x's from $N(1, 1) \rightarrow y = 1$

Example: Gaussian class conditional densities

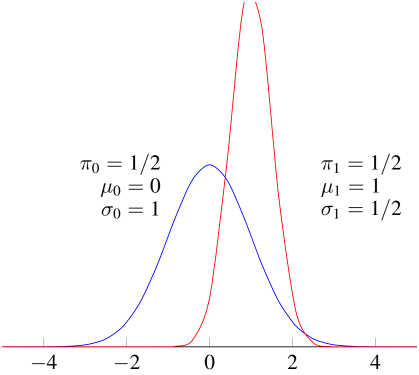


$1/2$ of x's from $N(0,1) \to y = 0$
$1/2$ of x's from $N(1,1) \to y = 1$

Bayes classifier:

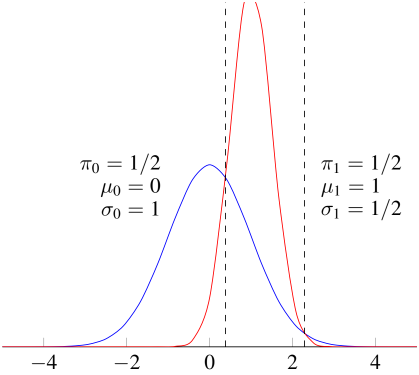
$$f^\star(x) = \begin{cases} 1 & \text{if } x > 1/2; \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE: GAUSSIAN CLASS CONDITIONAL DENSITIES



$1/2$ of x's from $\mathcal{N}(0, 1) \to y = 0$
$1/2$ of x's from $\mathcal{N}(1, 1/4) \to y = 1$

Example: Gaussian class conditional densities



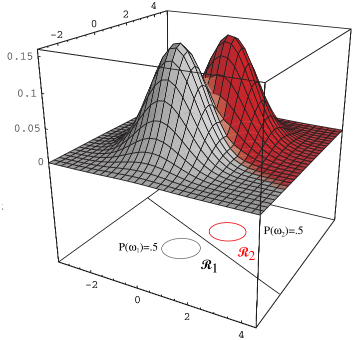
$1/2$ of x's from $\mathcal{N}(0, 1) \to y = 0$
$1/2$ of x's from $\mathcal{N}(1, 1/4) \to y = 1$

Bayes classifier:

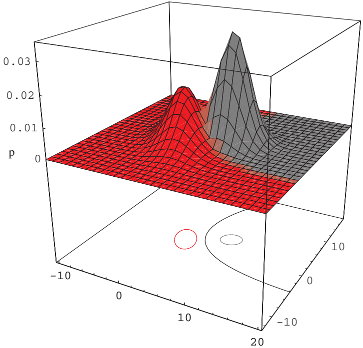
$$f^\star(x) = \begin{cases} 1 & \text{if } x \in [0.38, 2.29]; \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE: MULTIVARIATE GAUSSIANS

Data: $\mathcal{X} = \mathbb{R}^2$, Label: $\mathcal{Y} = \{0, 1\}$
Class conditional densities are Gaussians in $\mathbb{R}^2$ with covariance Σ_0 and Σ_1.

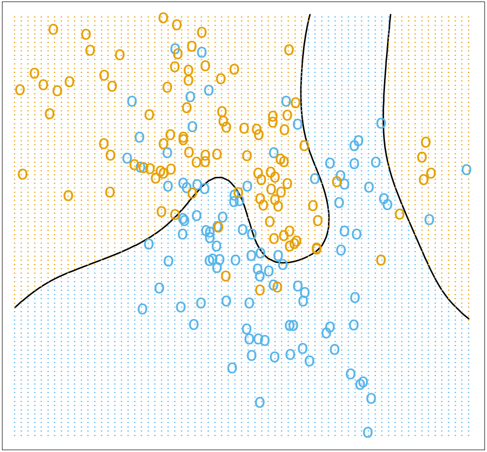


$\Sigma_0 = \Sigma_1$
Bayes classifier:
linear separator



$\Sigma_0 \neq \Sigma_1$
Bayes classifier:
quadratic separator

Bayes classifier in general



In general, the Bayes classifier may be rather complicated!
This one uses more than a single Gaussian for the class-conditional density.

PLUG-IN CLASSIFIERS

Bayes classifier

The Bayes classifier has the smallest prediction error of *all* classifiers.

Problem: We can't construct the Bayes classifier without knowing $\mathcal{P}$.

- What is $P(Y = y|X = x)$, or equiv., $P(X = x|Y = y)$ and $P(Y = y)$?
- All we have are labeled examples drawn from the distribution $\mathcal{P}$.

Plug-in classifiers

Use the available data to approximate $P(Y = y)$ and $P(X = x|Y = y)$.

- Of course, the result may no longer give the best results among all the classifiers we can choose from (e.g., k-NN and those discussed later).

EXAMPLE: GAUSSIAN CLASS CONDITIONAL DENSITIES

Here, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{1, \ldots, K\}$. Estimate Bayes classifier via MLE:

- **Class priors**: The MLE estimate of π_y is $\hat{\pi}_y = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(y_i = y)$.
- **Class conditional density**: Choose $p(x|Y = y) = N(x|\mu_y, \Sigma_y)$. The MLE estimate of (μ_y, Σ_y) is

$$\hat{\mu}_y = \frac{1}{n_y} \sum_{i=1}^{n} \mathbb{1}(y_i = y) x_i,$$

$$\hat{\Sigma}_y = \frac{1}{n_y} \sum_{i=1}^{n} \mathbb{1}(y_i = y)(x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^T.$$

  This is just the empirical mean and covariance of class y.

- **Plug-in classifier**:

$$\hat{f}(x) = \arg\max_{y \in \mathcal{Y}} \ \hat{\pi}_y |\hat{\Sigma}_y|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \hat{\mu}_y)^T \hat{\Sigma}_y^{-1} (x - \hat{\mu}_y)\right\}.$$

Example: Spam Filtering

Representing emails

- **Input**: x, a vector of word counts. For example, if index $\{j \to \text{“car”}\}$ $x[j] = 3$ means that the word “car” occurs three times in the email.
- **Output**: $\mathcal{Y} = \{-1, +1\}$. Map $\{\text{email} \to -1, \text{spam} \to +1\}$

Example dimensions

	george	you	your	hp	free	work	!	our	re	click	remove
spam	0	4	1	0	4	0	5	5	1	3	2
email	1	3	4	1	1	4	0	1	1	0	0

Using a Bayes classifier

$$f(x) = \underset{y \in \{-1,+1\}}{\operatorname{argmax}} \; p(x|Y = y)P(Y = y)$$

Naive Bayes

We have to *define* $p(X = x|Y = y)$.

Simplifying assumption

Naive Bayes is a Bayes classifier that makes the assumption

$$p(X = x|Y = y) = \prod_{j=1}^{d} p_j(x[j]|Y = y),$$

i.e., it treats the dimensions of X as *conditionally independent* given y.

In spam example

- Correlations between words are ignored.
- Can help make it easier to define the distribution.

ESTIMATION

Class prior

The distribution $P(Y = y)$ is again easy to estimate from the training data:

$$P(Y = y) \;=\; \frac{\#\text{observations in class } y}{\#\text{observations}}$$

Class-conditional distributions

For the spam model we define

$$P(X = x|Y = y) \;=\; \prod_j p_j(x[j]|Y = y) \;=\; \prod_j \text{Poisson}(x[j]|\lambda_j^{(y)})$$

We then approximate each $\lambda_j^{(y)}$ from the data. For example, the MLE is

$$\lambda_j^{(y)} = \frac{\#\text{unique uses of word } j \text{ in observations from class } y}{\#\text{observations in class } y}$$