Nearest neighbors classifiers

James McInerney
Adapted from slides by Daniel Hsu

Sept 11, 2017

Housekeeping

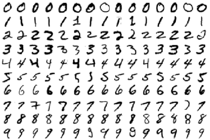
- We received 167 HW0 submissions on Gradescope before midnight Sept 10th.
- From a random sample, most look well done and are using the question assignment mechanism on Gradescope correctly.
- Example assignment submission.
- I received a few late submission emails. This one time only, I will turn the Gradescope submission page on again after class until 8pm EST.
- We will go over the trickiest parts of the homework at the beginning of the talk this Wednesday. Therefore I have pushed the first office hours to Wednesday.
- Here are the dates of the two exams for this course:
  - Exam 1: Wednesday October 18th, 2017
  - Exam 2: Monday December 11th, 2017

Example: OCR for digits

1. Classify images of handwritten digits by the actual digits they represent.
2. Classification problem: $\mathcal{Y} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (a discrete set).

Nearest neighbor (NN) classifier

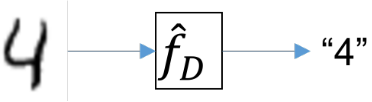
Given: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$



Predictor: $\hat{f}_D : \mathcal{X} \to \mathcal{Y}$

On input $\boldsymbol{x}$,

1. Find the point $\boldsymbol{x}_i$ among $\{\boldsymbol{x}_i\}_{i=1}^n$ that is “closest” to $\boldsymbol{x}$ (the *nearest neighbor*).
2. Return y_i.

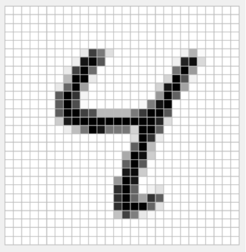


How to measure distance?

A default choice for distance between points in $\mathbb{R}^d$ is the *Euclidean distance* (also called ℓ_2 distance):

$$\|\boldsymbol{u} - \boldsymbol{v}\|_2 \ := \ \sqrt{\sum_{i=1}^{d} (u_i - v_i)^2}$$

(where $\boldsymbol{u} = (u_1, u_2, \ldots, u_d)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_d)$).



Grayscale 28×28 pixel images.

Treat as *vectors* (of 784 real-valued *features*) that live in $\mathbb{R}^{784}$.

Example: OCR for digits with NN classifier

- Classify images of handwritten digits by the digits they depict.

Example: OCR for digits with NN classifier

- Classify images of handwritten digits by the digits they depict.

- $\mathcal{X} = \mathbb{R}^{784}$, $\mathcal{Y} = \{0, 1, \ldots, 9\}$.

Example: OCR for digits with NN classifier

- Classify images of handwritten digits by the digits they depict.

- $\mathcal{X} = \mathbb{R}^{784}$, $\mathcal{Y} = \{0, 1, \ldots, 9\}$.

- **Given**: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y}$.

Example: OCR for digits with NN classifier

- Classify images of handwritten digits by the digits they depict.

- $\mathcal{X} = \mathbb{R}^{784}$, $\mathcal{Y} = \{0, 1, \ldots, 9\}$.
- **Given**: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y}$.
- Construct NN classifier $\hat{f}_D$ using D.

Example: OCR for digits with NN classifier

- Classify images of handwritten digits by the digits they depict.

- $\mathcal{X} = \mathbb{R}^{784}$, $\mathcal{Y} = \{0, 1, \ldots, 9\}$.
- **Given**: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y}$.
- Construct NN classifier $\hat{f}_D$ using D.
- **Question**: Is this classifier any good?

Error rate

- *Error rate* of classifier f on a set of labeled examples D:

$$\mathrm{err}_D(f) \ := \ \frac{\#\text{ of } (\boldsymbol{x}, y) \in D \text{ such that } f(\boldsymbol{x}) \neq y}{|D|}$$

(i.e., the fraction of D on which f disagrees with paired label).

Error rate

- *Error rate* of classifier f on a set of labeled examples D:

$$\mathrm{err}_D(f) \;:=\; \frac{\#\text{ of } (\boldsymbol{x}, y) \in D \text{ such that } f(\boldsymbol{x}) \neq y}{|D|}$$

  (i.e., the fraction of D on which f disagrees with paired label).

- Sometimes, we'll write this as $\mathrm{err}(f, D)$.

Error rate

- *Error rate* of classifier f on a set of labeled examples D:

$$\mathrm{err}_D(f) \ := \ \frac{\#\text{ of } (\boldsymbol{x}, y) \in D \text{ such that } f(\boldsymbol{x}) \neq y}{|D|}$$

  (i.e., the fraction of D on which f disagrees with paired label).

- Sometimes, we'll write this as $\mathrm{err}(f, D)$.

- **Question**: What is $\mathrm{err}_D(\hat{f}_D)$?

A better way to evaluate the classifier

- Split the labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.

A better way to evaluate the classifier

- Split the labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.
- Only use *training data* S to construct NN classifier $\hat{f}_S$.

A better way to evaluate the classifier

- Split the labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.
- Only use *training data* S to construct NN classifier $\hat{f}_S$.
  - Training error rate of $\hat{f}_S$: $\text{err}_S(\hat{f}_S) = 0\%$.

A better way to evaluate the classifier

- Split the labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.
- Only use *training data* S to construct NN classifier $\hat{f}_S$.
  - Training error rate of $\hat{f}_S$: $\mathrm{err}_S(\hat{f}_S) = 0\%$.
- Use *test data* T to evaluate accuracy of $\hat{f}_S$.

A better way to evaluate the classifier

- Split the labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.
- Only use *training data* S to construct NN classifier $\hat{f}_S$.
  - Training error rate of $\hat{f}_S$: $\mathrm{err}_S(\hat{f}_S) = 0\%$.
- Use *test data* T to evaluate accuracy of $\hat{f}_S$.
  - Test error rate of $\hat{f}_S$: $\mathrm{err}_T(\hat{f}_S) = 3.09\%$.

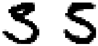
A better way to evaluate the classifier

- Split the labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into two sets (randomly).
  - *Training data* S.
  - *Test data* T.
- Only use *training data* S to construct NN classifier $\hat{f}_S$.
  - Training error rate of $\hat{f}_S$: $\mathrm{err}_S(\hat{f}_S) = 0\%$.
- Use *test data* T to evaluate accuracy of $\hat{f}_S$.
  - Test error rate of $\hat{f}_S$: $\mathrm{err}_T(\hat{f}_S) = 3.09\%$.

Is this good?

Diagnostics

- Some mistakes made by the NN classifier
  (test point in T, nearest neighbor in S):


- First mistake (correct label is "2") could've been avoided by looking at the *three* nearest neighbors (whose labels are "8", "2", and "2").


test point three nearest neighbors

k-nearest neighbors classifier

Given: labeled examples $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$
Predictor: $\hat{f}_{D,k} \colon \mathcal{X} \to \mathcal{Y}$:

On input $\boldsymbol{x}$,

1. Find the k points $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \dots, \boldsymbol{x}_{i_k}$ among $\{\boldsymbol{x}_i\}_{i=1}^n$ "closest" to $\boldsymbol{x}$ (the k nearest neighbors).
2. Return the plurality of $y_{i_1}, y_{i_2}, \dots, y_{i_k}$.

(Break ties in both steps arbitrarily.)

Effect of k

- Smaller k: smaller training error rate.
- Larger k: higher training error rate, but predictions are more "stable" due to voting.

OCR digits classification

k	1	3	5	7	9
Test error rate	0.0309	0.0295	0.0312	0.0306	0.0341

Choosing k

The hold-out set approach

1. Pick a subset $V \subset S$ (*hold-out set*, a.k.a. *validation set*).
2. For each $k \in \{1, 3, 5, \ldots\}$:
   - Construct k-NN classifier $\hat{f}_{S \setminus V, k}$ using $S \setminus V$.
   - Compute error rate of $\hat{f}_{S \setminus V, k}$ on V ("hold-out error rate").
3. Pick the k that gives the smallest hold-out error rate.

Other distance functions

- **Lp norm**

$$\text{dist}(u, v) \;=\; ||x_1^p + x_2^p + \cdots + x_d^p||^{\frac{1}{p}}$$

Other distance functions

- **Lp norm**

$$\text{dist}(u, v) \;=\; ||x_1^p + x_2^p + \cdots + x_d^p||^{\frac{1}{p}}$$

OCR digits classification

Distance	ℓ_2	ℓ_3
Test error rate	3.09%	2.83%

Other distance functions

- **Lp norm**

$$\text{dist}(u, v) \;=\; ||x_1^p + x_2^p + \cdots + x_d^p||^{\frac{1}{p}}$$

OCR digits classification

Distance	ℓ_2	ℓ_3
Test error rate	3.09%	2.83%

- **Manhattan distance**

$\text{dist}(u, v) \;=\;$ distance on grid between u and v on strict horizontal/vertical path

Other distance functions

- **Lp norm**

$$\text{dist}(u, v) = ||x_1^p + x_2^p + \cdots + x_d^p||^{\frac{1}{p}}$$

OCR digits classification

Distance	ℓ_2	ℓ_3
Test error rate	3.09%	2.83%

- **Manhattan distance**

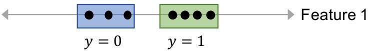
$\text{dist}(u, v) =$ distance on grid between u and v on strict horizontal/vertical path

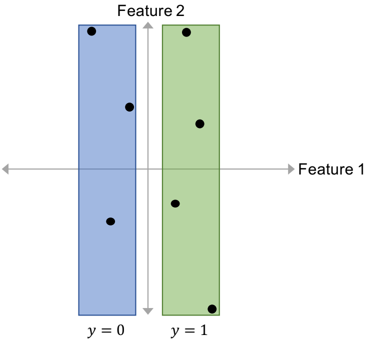
- **String edit distance**

$\text{dist}(u, v) = \#$ insertions/deletions/mutations needed to change u to v

Bad features

Caution: nearest neighbor classifier can be broken by bad/noisy features!





Questions of interest

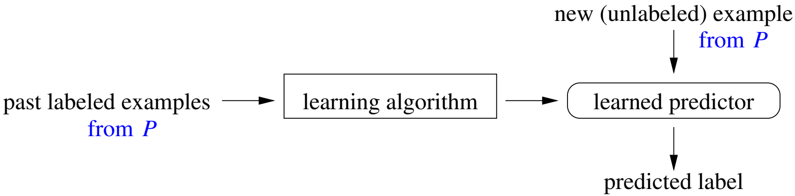
1. How good is the classifier learned using NN *on your problem*?
2. Is NN a good learning method *in general*?

Statistical learning theory

Basic assumption (main idea):
labeled examples $\{(\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ come from same source as future examples.

Statistical learning theory

Basic assumption (main idea):
labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ come from same source as future examples.



More formally:
$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is an *i.i.d. sample* from a probability distribution P over $\mathcal{X} \times \mathcal{Y}$.

Prediction error rate

- Define the *(true) error rate* of a classifier $f\colon \mathcal{X} \to \mathcal{Y}$ w.r.t. P to be

$$\mathrm{err}_P(f) \ := \ P(f(\boldsymbol{X}) \neq Y)$$

  where $(\boldsymbol{X}, Y)$ is a pair of random variables with joint distribution P (i.e., $(\boldsymbol{X}, Y) \sim P$).

Prediction error rate

- Define the *(true) error rate* of a classifier $f \colon \mathcal{X} \to \mathcal{Y}$ w.r.t. P to be

$$\mathrm{err}_P(f) \ := \ P(f(\boldsymbol{X}) \neq Y)$$

  where $(\boldsymbol{X}, Y)$ is a pair of random variables with joint distribution P (i.e., $(\boldsymbol{X}, Y) \sim P$).

- Let $\hat{f}_S$ be classifier trained using labeled examples S.

Prediction error rate

- Define the *(true) error rate* of a classifier $f\colon \mathcal{X} \to \mathcal{Y}$ w.r.t. P to be

$$\mathrm{err}_P(f) \ := \ P(f(\boldsymbol{X}) \neq Y)$$

  where $(\boldsymbol{X}, Y)$ is a pair of random variables with joint distribution P (i.e., $(\boldsymbol{X}, Y) \sim P$).

- Let $\hat{f}_S$ be classifier trained using labeled examples S.

- True error rate of $\hat{f}_S$ is

$$\mathrm{err}_P(\hat{f}_S) \ := \ P(\hat{f}_S(\boldsymbol{X}) \neq Y)\,.$$

Prediction error rate

- Define the *(true) error rate* of a classifier $f\colon \mathcal{X} \to \mathcal{Y}$ w.r.t. P to be

$$\mathrm{err}_P(f) \; := \; P(f(\boldsymbol{X}) \neq Y)$$

where $(\boldsymbol{X}, Y)$ is a pair of random variables with joint distribution P (i.e., $(\boldsymbol{X}, Y) \sim P$).

- Let $\hat{f}_S$ be classifier trained using labeled examples S.
- True error rate of $\hat{f}_S$ is

$$\mathrm{err}_P(\hat{f}_S) \; := \; P(\hat{f}_S(\boldsymbol{X}) \neq Y)\,.$$

- We cannot compute this without knowing P.

Estimating the true error rate

- Suppose $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ (assumed to be an i.i.d. sample from P) is randomly split into S and T, and $\hat{f}_S$ is based only on S.

Estimating the true error rate

- Suppose $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ (assumed to be an i.i.d. sample from P) is randomly split into S and T, and $\hat{f}_S$ is based only on S.
- $\hat{f}_S$ and T are *independent*, and the *test error rate* of $\hat{f}_S$ is an *unbiased* estimate of the true error rate of $\hat{f}_S$.

Estimating the true error rate

- Suppose $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ (assumed to be an i.i.d. sample from P) is randomly split into S and T, and $\hat{f}_S$ is based only on S.
- $\hat{f}_S$ and T are *independent*, and the *test error rate* of $\hat{f}_S$ is an *unbiased* estimate of the true error rate of $\hat{f}_S$.
- If $|T| = m$, then the test error rate $\mathrm{err}_T(\hat{f}_S)$ of $\hat{f}_S$ (conditional on S) is a *binomial random variable* (scaled by $1/m$):

$$m \cdot \mathrm{err}_T(\hat{f}_S) \mid S \;\sim\; \mathrm{Bin}(m, \mathrm{err}_P(\hat{f}_S))\,.$$

Estimating the true error rate

- Suppose $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ (assumed to be an i.i.d. sample from P) is randomly split into S and T, and $\hat{f}_S$ is based only on S.
- $\hat{f}_S$ and T are *independent*, and the *test error rate* of $\hat{f}_S$ is an *unbiased* estimate of the true error rate of $\hat{f}_S$.
- If $|T| = m$, then the test error rate $\mathrm{err}_T(\hat{f}_S)$ of $\hat{f}_S$ (conditional on S) is a *binomial random variable* (scaled by $1/m$):

$$m \cdot \mathrm{err}_T(\hat{f}_S) \mid S \;\sim\; \mathrm{Bin}(m, \mathrm{err}_P(\hat{f}_S))\,.$$

- The expected value of $\mathrm{err}_T(\hat{f}_S)$ is $\mathrm{err}_P(\hat{f}_S)$.
  (This means that $\mathrm{err}_T(\hat{f}_S)$ is an *unbiased estimator* of $\mathrm{err}_P(\hat{f}_S)$.)

Limits of prediction

- Binary classification: $\mathcal{Y} = \{0, 1\}$.

Limits of prediction

- Binary classification: $\mathcal{Y} = \{0, 1\}$.
- Probability distribution P over $\mathcal{X} \times \{0, 1\}$; let $(\boldsymbol{X}, Y) \sim P$.

Limits of prediction

- Binary classification: $\mathcal{Y} = \{0, 1\}$.
- Probability distribution P over $\mathcal{X} \times \{0, 1\}$; let $(\boldsymbol{X}, Y) \sim P$.
- Think of P as being comprised of two parts.
  1. Marginal distribution μ of $\boldsymbol{X}$ (a distribution over $\mathcal{X}$).
  2. Conditional distribution of Y given $\boldsymbol{X} = \boldsymbol{x}$, for each $\boldsymbol{x} \in \mathcal{X}$:

$$\eta(\boldsymbol{x}) \; := \; P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \, .$$

Limits of prediction

- Binary classification: $\mathcal{Y} = \{0, 1\}$.
- Probability distribution P over $\mathcal{X} \times \{0, 1\}$; let $(\boldsymbol{X}, Y) \sim P$.
- Think of P as being comprised of two parts.
  1. Marginal distribution μ of $\boldsymbol{X}$ (a distribution over $\mathcal{X}$).
  2. Conditional distribution of Y given $\boldsymbol{X} = \boldsymbol{x}$, for each $\boldsymbol{x} \in \mathcal{X}$:

$$\eta(\boldsymbol{x}) \; := \; P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \,.$$

- If $\eta(\boldsymbol{x})$ is 0 or 1 for all $\boldsymbol{x} \in \mathcal{X}$ where $\mu(\boldsymbol{x}) > 0$,
  then optimal error rate is zero (i.e., $\min_f \operatorname{err}_P(f) = 0$).

Limits of prediction

- Binary classification: $\mathcal{Y} = \{0, 1\}$.
- Probability distribution P over $\mathcal{X} \times \{0, 1\}$; let $(\boldsymbol{X}, Y) \sim P$.
- Think of P as being comprised of two parts.
  1. Marginal distribution μ of $\boldsymbol{X}$ (a distribution over $\mathcal{X}$).
  2. Conditional distribution of Y given $\boldsymbol{X} = \boldsymbol{x}$, for each $\boldsymbol{x} \in \mathcal{X}$:

$$\eta(\boldsymbol{x}) \ := \ P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \, .$$

- If $\eta(\boldsymbol{x})$ is 0 or 1 for all $\boldsymbol{x} \in \mathcal{X}$ where $\mu(\boldsymbol{x}) > 0$,
  then optimal error rate is zero (i.e., $\min_f \operatorname{err}_P(f) = 0$).
- Otherwise it is non-zero.

Bayes optimality

- What is the classifier with smallest true error rate?

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

(Do you see why?)

Bayes optimality

- What is the classifier with smallest true error rate?

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

(Do you see why?)

- $f^\star$ is called the *Bayes (optimal) classifier*, and

$$\mathrm{err}_P(f^\star) \;=\; \min_f \mathrm{err}_P(f) \;=\; \mathbb{E}\Big[\min\big\{\eta(\boldsymbol{X}),\, 1-\eta(\boldsymbol{X})\big\}\Big]$$

which is called the *Bayes (optimal) error rate*.

Bayes optimality

- What is the classifier with smallest true error rate?

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

  (Do you see why?)

- $f^\star$ is called the *Bayes (optimal) classifier*, and

$$\mathrm{err}_P(f^\star) = \min_f \mathrm{err}_P(f) = \mathbb{E}\Big[\min\big\{\eta(\boldsymbol{X}),\, 1 - \eta(\boldsymbol{X})\big\}\Big]$$

  which is called the *Bayes (optimal) error rate*.

Question:
How far from optimal is the classifier produced by the NN learning method?

Consistency of k-NN

We say that a learning algorithm A is **consistent** if

$$\lim_{n\to\infty} \mathbb{E}\Big[\mathrm{err}_P(\hat{f}_n)\Big] \;=\; \mathrm{err}(f^\star)\,,$$

where $\hat{f}_n$ is the classifier learned using A on an i.i.d. sample of size n.

Theorem (e.g., Cover and Hart 1967)

Assume η is continuous. Then:

- 1-*NN is consistent if* $\min_f \mathrm{err}_P(f) = 0$.
- k-*NN is consistent, provided that* $k := k_n$ *is chosen as an increasing but sublinear function of* n*:*

$$\lim_{n\to\infty} k_n \;=\; \infty\,, \qquad \lim_{n\to\infty} \frac{k_n}{n} \;=\; 0\,.$$

Key takeaways

1. k-NN learning procedure; role of k, distance functions, features.
2. Training and test error rates.
3. Framework of statistical learning theory; estimating the “true” error rate; Bayes optimality; high-level idea of consistency.

Recap: definitions

- denote our classifier as f

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D
- what is $\mathrm{err}_D(f)$?

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D
- what is $\mathrm{err}_D(f)$? $\frac{\#\text{ of }(\boldsymbol{x}, y) \in D\text{ such that }f(\boldsymbol{x}) \neq y}{|D|}$

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D
- what is $\mathrm{err}_D(f)$? $\frac{\#\text{ of }(\boldsymbol{x}, y) \in D\text{ such that }f(\boldsymbol{x}) \neq y}{|D|}$
- what does $P(\mathbf{X}, Y)$ mean?

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D
- what is $\mathrm{err}_D(f)$? $\frac{\#\text{ of } (\boldsymbol{x}, y) \in D \text{ such that } f(\boldsymbol{x}) \neq y}{|D|}$
- what does $P(\mathbf{X}, Y)$ mean?
- what is $\mathrm{err}_P(f)$?

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D
- what is $\mathrm{err}_D(f)$? $\frac{\#\text{ of } (\boldsymbol{x}, y) \in D \text{ such that } f(\boldsymbol{x}) \neq y}{|D|}$
- what does $P(\mathbf{X}, Y)$ mean?
- what is $\mathrm{err}_P(f)$? $\mathbb{E}_P[\mathbb{I}[f(\mathbf{X}) \neq Y]]$

Recap: definitions

- denote our classifier as f
- what does $f(x)$ mean?
- denote the dataset as D
- what is $\mathrm{err}_D(f)$? $\frac{\#\text{ of } (\boldsymbol{x}, y) \in D \text{ such that } f(\boldsymbol{x}) \neq y}{|D|}$
- what does $P(\mathbf{X}, Y)$ mean?
- what is $\mathrm{err}_P(f)$? $\mathbb{E}_P[\mathbb{I}[f(\mathbf{X}) \neq Y]] = P(f(\mathbf{X}) \neq Y)$

Recap: unbiased estimator

- what is an estimator?

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of?

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of?
  $\mathrm{err}_P(f)$

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of? $\mathrm{err}_P(f)$
- what is an *unbiased* estimator?

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of? $\mathrm{err}_P(f)$
- what is an *unbiased* estimator? $\mathbb{E}[\mathrm{err}_D(f)] = \mathrm{err}_P(f)$

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of? $\mathrm{err}_P(f)$
- what is an *unbiased* estimator? $\mathbb{E}[\mathrm{err}_D(f)] = \mathrm{err}_P(f)$
- what on earth does that mean?

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of? $\mathrm{err}_P(f)$
- what is an *unbiased* estimator? $\mathbb{E}[\mathrm{err}_D(f)] = \mathrm{err}_P(f)$
- what on earth does that mean? if we repeatedly draw datasets of size n, $D \sim_n P(\mathbf{X}, Y)$ and calculate $\mathrm{err}_D(f)$, the average will converge to $\mathrm{err}_P(f)$

Recap: unbiased estimator

- what is an estimator?
- using definitions from previous slide, what is $\mathrm{err}_D(f)$ an *estimator* of? $\mathrm{err}_P(f)$
- what is an *unbiased* estimator? $\mathbb{E}[\mathrm{err}_D(f)] = \mathrm{err}_P(f)$
- what on earth does that mean? if we repeatedly draw datasets of size n, $D \sim_n P(\mathbf{X}, Y)$ and calculate $\mathrm{err}_D(f)$, the average will converge to $\mathrm{err}_P(f)$
- why should we care?

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples
- what is the property $\lim_{n\to\infty} \mathbb{E}\Big[\operatorname{err}_P(\hat{f}_n)\Big] = \operatorname{err}(f^\star)$?

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples
- what is the property $\lim_{n\to\infty} \mathbb{E}\left[\mathrm{err}_P(\hat{f}_n)\right] = \mathrm{err}(f^\star)$? consistency

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples
- what is the property $\lim_{n\to\infty} \mathbb{E}\left[\text{err}_P(\hat{f}_n)\right] = \text{err}(f^\star)$? consistency
- what is the ideal classifier $f^\star$ also known as?

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples
- what is the property $\lim_{n\to\infty} \mathbb{E}\Big[\mathrm{err}_P(\hat{f}_n)\Big] = \mathrm{err}(f^\star)$? consistency
- what is the ideal classifier $f^\star$ also known as? the Bayes optimal classifier

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples
- what is the property $\lim_{n\to\infty} \mathbb{E}\Big[\mathrm{err}_P(\hat{f}_n)\Big] = \mathrm{err}(f^\star)$? consistency
- what is the ideal classifier $f^\star$ also known as? the Bayes optimal classifier
- how can we define $f^\star$?

Recap: consistent algorithm

- let $\hat{f}_n$ mean a classifier trained using n examples
- what is the property $\lim_{n\to\infty} \mathbb{E}\left[\operatorname{err}_P(\hat{f}_n)\right] = \operatorname{err}(f^\star)$? consistency
- what is the ideal classifier $f^\star$ also known as? the Bayes optimal classifier
- how can we define $f^\star$?
  $\operatorname{err}_P(f^\star) = \min_f \operatorname{err}_P(f) = \mathbb{E}\left[\min\{\eta(\boldsymbol{X}),\, 1 - \eta(\boldsymbol{X})\}\right]$