ORIE 4741: Learning with Big Messy Data

Is Learning Feasible?

Professor Udell
Operations Research and Information Engineering
Cornell

November 11, 2016
Recap: Perceptron

- a simple learning algorithm that works for linearly separable data
- themes we’ll see again: linear functions, iterative updates
- how we plotted the data: axes = \( \mathcal{X} \), color = \( \mathcal{Y} \)
- vector \( w \in \mathbb{R}^d \) defines linear decision boundary
- simplify algorithm with feature transformation to handle offset
- proof of convergence: induction, Cauchy-Schwartz, linear algebra
Desiderata

- good fit: does algorithm fit the data set well?
- robustness: does algorithm degrade gracefully with corruptions in data?
- stability: how much do predictions change if the data changes slightly?
Note on notation

we’ll use $[[\text{statement}]]$ to mean 1 if the statement is true, and false otherwise.

examples:

- $[[1 < 0]] = 0$
- $[[17 = 17]] = 1$
Pocket learning algorithm

like perceptron, but

- at each iteration, record the error (eg, number of misclassifications)

\[ \sum_{i=1}^{n} \left[ [y_i \neq \text{sign}(w^\top x_i)] \right] \]

- keep track of the \( w(t) \) that achieved the best error so far
Pocket learning algorithm

like perceptron, but

- at each iteration, record the error (e.g., number of misclassifications)

\[
\sum_{i=1}^{n} \left[ y_i \neq \text{sign}(w^\top x_i) \right]
\]

- keep track of the \( w(t) \) that achieved the best error so far

properties:

- no convergence guarantees
- don’t know when to stop
- takes a long time to compute error
- can perform well even on non-separable data
Generalization of supervised learning

- unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
- training examples $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- hypothesis set $\mathcal{H}$
- learning algorithm $\mathcal{A}$
- final hypothesis $g : \mathcal{X} \rightarrow \mathcal{Y}$

how well will our classifier do on new data?
Generalization of supervised learning

how well will our classifier do on new data?
Generalization of supervised learning

how well will our classifier do on new data?

- worst case: terrible
Generalization of supervised learning

how well will our classifier do on **new** data?

- worst case: terrible
- average case? usually?

probability to the rescue...
Simplest case: generalizing from a mean

exit polling

- sample $n$ voters leaving polling places\(^1\)
- for each voter $i$, define the Boolean random variable $z_i = \begin{cases} 
1 & \text{if voter } i \text{ voted for Clinton} \\
0 & \text{otherwise} 
\end{cases}$

- sample mean: $\nu = \frac{1}{n} \sum_{i=1}^{n} z_i$
- true mean: $\mu = \mathbb{E}_{i \sim \text{US electorate}} z_i$ is Clinton’s expected vote share

\(^1\)and suppose no one votes by mail or early or . . . , so these are iid samples from the US electorate
**Simplest case: generalizing from a mean**

exit polling

- sample \( n \) voters leaving polling places\(^1\)
- for each voter \( i \), define the Boolean random variable
  
  \[ z_i = \begin{cases} 
  1 & \text{if voter } i \text{ voted for Clinton} \\
  0 & \text{otherwise} 
  \end{cases} \]

- sample mean: \( \nu = \frac{1}{n} \sum_{i=1}^{n} z_i \)
- true mean: \( \mu = \mathbb{E}_{i \sim \text{US electorate}} z_i \) is Clinton’s expected vote share

what does the sample mean \( \nu \) tell us about the true mean \( \mu \)?

\(^1\)and suppose no one votes by mail or early or . . . , so these are iid samples from the US electorate
Hoeffding inequality

Theorem (Hoeffding Inequality)

Let $z_i \in \{0, 1\}, i = 1, \ldots, n$, be independent Boolean random variables with mean $\mathbb{E} z_i = \mu$. Define the sample mean $\nu = \frac{1}{n} \sum_{i=1}^{n} z_i$. Then for any $\epsilon > 0$,

$$\mathbb{P}[|\nu - \mu| > \epsilon] \leq 2 \exp \left(-2\epsilon^2 n\right).$$

an example of a concentration inequality
Hoeffding inequality

Theorem (Hoeffding Inequality)

Let $z_i \in \{0, 1\}, i = 1, \ldots, n$, be independent Boolean random variables with mean $E z_i = \mu$. Define the sample mean $\nu = \frac{1}{n} \sum_{i=1}^{n} z_i$. Then for any $\epsilon > 0$,

$$\Pr[|\nu - \mu| > \epsilon] \leq 2 \exp\left(-2\epsilon^2 n\right).$$

an example of a concentration inequality

- $\mu$ can’t be much higher than $\nu$
- $\mu$ can’t be much lower than $\nu$
- more samples $n$ improve estimate exponentially quickly
Back to the learning problem

fix a hypothesis \( h : \mathcal{X} \to \mathcal{Y} \). take

\[
z_i = \begin{cases} 
1 & y_i = h(x_i) \\
0 & \text{otherwise}
\end{cases}
\]

\[= z_i = [[y_i = h(x_i)]]\]
Back to the learning problem

fix a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$. take

$$z_i = \begin{cases} 1 & y_i = h(x_i) \\ 0 & \text{otherwise} \end{cases}$$

$$= z_i = [y_i = h(x_i)]$$

**example.** build a model of voting behavior:

- $y_i = f(x_i)$ is 1 if voter $i$ voted for Clinton, 0 otherwise
- $h(x_i)$ is our guess of how the voter will vote, using hypothesis $h$
- $z_i$ is 1 if we guess correctly for voter $i$, 0 otherwise
- $z_i$ depends on $x_i$, $y_i$, and $h$
Adding in probability

make our model probabilistic:

- fix a probability distribution $P(x, y)$
- sample $(x_i, y_i)$ iid from $P(x, y)$
- form data set $\mathcal{D}$ by sampling:
  - for $i = 1, \ldots, n$
    - sample $(x_i, y_i) \sim P(x, y)$
  - set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
Adding in probability

make our model probabilistic:

- fix a probability distribution $P(x, y)$
- sample $(x_i, y_i)$ iid from $P(x, y)$
- form data set $\mathcal{D}$ by sampling:
  - for $i = 1, \ldots, n$
    - sample $(x_i, y_i) \sim P(x, y)$
  - set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

special case. $y = f(x)$ is deterministic conditioned on $x$:

$$
P(y|x) = \begin{cases} 
1 & y = f(x) \\
0 & \text{otherwise}
\end{cases}
$$
Hoeffding for the noisy learning problem

- fix a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$.
- consider $(x_i, y_i)$ as samples of random variables drawn from $P(x, y)$.
- take $z_i = \left[ y_i = h(x_i) \right]$.
- $z_i$ are iid (since $(x_i, y_i)$ are iid, and $h$ is fixed).
- $\mathbb{E}z = \mathbb{E}_{(x,y) \sim P(x,y)}[\left[ y = h(x) \right]]$

so we can apply Hoeffding!

\[
\mathbb{P}\left[ \left| \frac{1}{n} \sum_{i=1}^{n} z_i - \mathbb{E}z \right| > \varepsilon \right] \leq 2 \exp \left( -2\varepsilon^2 n \right)
\]

Q: what is the probability over?
A: other data sets $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn iid according to $P(x, y)$
In-sample and out-of-sample error

let’s make some new notation:

▶ in-sample error.

\[ E_{\text{in}}(h) = \text{fraction of } D \text{ where } y_i \text{ and } h(x_i) \text{ disagree} \]
\[ = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[y_i \neq h(x_i)] \]

▶ out-of-sample error.

\[ E_{\text{out}}(h) = \text{probability that } y \text{ and } h(x) \text{ disagree} \]
\[ = \mathbb{P}_{(x,y) \sim P(x,y)} [y \neq h(x)] \]
Hoeffding for the noisy learning problem

- fix a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$.
- consider $(x_i, y_i)$ as samples of random variables drawn from $P(x, y)$
- take $z_i = [[y_i = h(x_i)]]$
- $z_i$ are iid (since $(x_i, y_i)$ are iid, and $h$ is fixed)
- $Ez = \mathbb{E}_{(x,y) \sim P(x,y)}[[y = h(x)]]$

apply Hoeffding:

$$\mathbb{P}[\left| E_{\text{in}}(h) - E_{\text{out}}(h) \right| > \varepsilon] \leq 2 \exp \left( -2\varepsilon^2 n \right)$$
Does Hoeffding work for our learned model?

two scenarios:

1. Without looking at any data, we pick a model \( h : \mathcal{X} \rightarrow \mathcal{Y} \) to predict who will vote for Clinton. Then we sample data \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), and set \( z_i = [[y_i = h(x_i)]] \). Is the sample mean \( \frac{1}{n} \sum_{i=1}^{n} z_i \) a good estimate for the expected performance \( \mathbb{E}z \) of our model?

2. We sample the data \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), and use it to develop a model \( g : \mathcal{X} \rightarrow \mathcal{Y} \) to predict who will vote for Clinton. Set \( z'_i = [[y_i = g(x_i)]] \). Is the sample mean \( \frac{1}{n} \sum_{i=1}^{n} z'_i \) a good estimate for the expected performance \( \mathbb{E}z' \) of our model?

Q: does Hoeffding apply to the first? the second?
Does Hoeffding work for our learned model?

two scenarios:

1. Without looking at any data, we pick a model $h : \mathcal{X} \rightarrow \mathcal{Y}$ to predict who will vote for Clinton. Then we sample data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and set $z_i = \left[ [y_i = h(x_i)] \right]$. Is the sample mean $\frac{1}{n} \sum_{i=1}^{n} z_i$ a good estimate for the expected performance $\mathbb{E}z$ of our model?

2. We sample the data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and use it to develop a model $g : \mathcal{X} \rightarrow \mathcal{Y}$ to predict who will vote for Clinton. Set $z'_i = \left[ [y_i = g(x_i)] \right]$. Is the sample mean $\frac{1}{n} \sum_{i=1}^{n} z'_i$ a good estimate for the expected performance $\mathbb{E}z'$ of our model?

Q: does Hoeffding apply to the first? the second?
A: Hoeffding applies to first, not to second. Extreme case for second scenario: model memorizes the data.
Does Hoeffding work for our learned model?

two scenarios:

1. Without looking at any data, we pick a model $h : \mathcal{X} \to \mathcal{Y}$ to predict who will vote for Clinton. Then we sample data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and set $z_i = [y_i = h(x_i)]$. Is the sample mean $\frac{1}{n} \sum_{i=1}^{n} z_i$ a good estimate for the expected performance $\mathbb{E}z$ of our model?

2. We sample the data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and use it to develop a model $g : \mathcal{X} \to \mathcal{Y}$ to predict who will vote for Clinton. Set $z'_i = [y_i = g(x_i)]$. Is the sample mean $\frac{1}{n} \sum_{i=1}^{n} z'_i$ a good estimate for the expected performance $\mathbb{E}z'$ of our model?

Q: why doesn’t Hoeffding apply to the second scenario?
Does Hoeffding work for our learned model?

two scenarios:

1. Without looking at any data, we pick a model \( h : \mathcal{X} \rightarrow \mathcal{Y} \) to predict who will vote for Clinton. Then we sample data \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), and set \( z_i = \left[ y_i = h(x_i) \right] \).

   Is the sample mean \( \frac{1}{n} \sum_{i=1}^{n} z_i \) a good estimate for the expected performance \( \mathbb{E}z \) of our model?

2. We sample the data \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), and use it to develop a model \( g : \mathcal{X} \rightarrow \mathcal{Y} \) to predict who will vote for Clinton. Set \( z'_i = \left[ y_i = g(x_i) \right] \).

   Is the sample mean \( \frac{1}{n} \sum_{i=1}^{n} z'_i \) a good estimate for the expected performance \( \mathbb{E}z' \) of our model?

**Q:** why doesn’t Hoeffding apply to the second scenario?

**A:** \( z_i \) are not independent: they depend on \( g \), which depends on the whole data set \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \).
Rescuing Hoeffding: the union bound

- let’s suppose $\mathcal{H}$ is finite, with $m$ hypotheses in it
- the hypothesis $g$ is one of those $m$ hypotheses
- so if (given a data set $\mathcal{D}$)
  \[ |E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon, \]
  then \[ |E_{\text{in}}(h) - E_{\text{out}}(h)| > \varepsilon \]
  must be true for some $h \in \mathcal{H}$
- ($g$ depends on the data set; we might choose different $h$s for different data sets)
Rescuing Hoeffding: the union bound

- let’s suppose $\mathcal{H}$ is finite, with $m$ hypotheses in it
- the hypothesis $g$ is one of those $m$ hypotheses
- so if (given a data set $D$)

$$|E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon,$$

then $|E_{\text{in}}(h) - E_{\text{out}}(h)| > \varepsilon$ must be true for some $h \in \mathcal{H}$
- ($g$ depends on the data set; we might choose different $h$s for different data sets)

SO

$$\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon] \leq \sum_{h \in \mathcal{H}} \mathbb{P}[|E_{\text{in}}(h) - E_{\text{out}}(h)| > \varepsilon]$$

$$\leq \sum_{h \in \mathcal{H}} 2 \exp \left( -2\varepsilon^2 n \right)$$

$$= 2m \exp \left( -2\varepsilon^2 n \right)$$
Hoeffding for learning

we just proved that our learning algorithm generalizes!

Theorem (Generalization bound for learning)

Let $g$ be a hypothesis chosen from among $m$ different hypotheses. Then

$$\Pr[|E_{in}(g) - E_{out}(g)| > \varepsilon] \leq 2m \exp\left(-2\varepsilon^2 n\right).$$
Hoeffding for learning

we just proved that our learning algorithm generalizes!

Theorem (Generalization bound for learning)

Let \( g \) be a hypothesis chosen from among \( m \) different hypotheses. Then

\[
\mathbb{P}[|E_{in}(g) - E_{out}(g)| > \varepsilon] \leq 2m \exp\left(-2\varepsilon^2 n\right).
\]

Q: do you think this bound is tight?
Hoeffding for learning

we just proved that our learning algorithm generalizes!

Theorem (Generalization bound for learning)

Let \( g \) be a hypothesis chosen from among \( m \) different hypotheses. Then

\[
P[|E_{in}(g) - E_{out}(g)| > \varepsilon] \leq 2m \exp \left( -2\varepsilon^2 n \right).
\]

Q: do you think this bound is tight?
A: no, it can overcount badly
we just proved that our learning algorithm generalizes!

**Theorem (Generalization bound for learning)**

Let $g$ be a hypothesis chosen from among $m$ different hypotheses. Then

$$\Pr[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon] \leq 2m \exp \left(-2\varepsilon^2 n\right).$$

Q: do you think this bound is tight?
A: no, it can overcount badly

**more information.** Look up the Vapnik-Chervoninkis (VC) dimension, e.g., in *Learning from Data*, by Abu-Mostafa, Magdon-Ismail, and Lin.
A tradeoff for learning

- we want $\mathcal{H}$ to be **big** to make $E_{in}$ small
- we want $\mathcal{H}$ to be **small** to ensure $E_{out}$ is close to $E_{in}$
A tradeoff for learning

- we want $\mathcal{H}$ to be **big** to make $E_{in}$ small
- we want $\mathcal{H}$ to be **small** to ensure $E_{out}$ is close to $E_{in}$

what does this tell us about the difficulty of learning complicated functions $f$?
Recap

- we introduced a probabilistic framework for generating data
- we showed that the in-sample error predicts the out-of-sample error for a single hypothesis
- we showed that the in-sample error predicts the out-of-sample error for a learned hypothesis, when $H$ is finite
- we stopped there, because the math gets much more complicated — but indeed, generalization is possible!
- the practical lesson: (especially for complex models), don’t learn and estimate your error on the same data set
You run a hospital. A vendor wants to sell you a new machine learning system for diagnosing patients, and tells you the system works with 99% accuracy.

If she’s right, you might save millions of dollars and thousands of lives. If she’s wrong, you might lose the same. What evidence would you ask for to verify this claim?
You run a hospital. A vendor wants to sell you a new machine learning system for diagnosing patients, and tells you the system works with 99% accuracy.

If she’s right, you might save millions of dollars and thousands of lives. If she’s wrong, you might lose the same. What evidence would you ask for to verify this claim?

Idea: see how well the model works on data it was not trained on.
Cross-validation

some new notation:

- partition data $\mathcal{D}$ into
  - training set $\mathcal{D}_{\text{train}}$ and
  - test set $\mathcal{D}_{\text{test}}$

  so $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \emptyset$, $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}} = \mathcal{D}$

- algorithm $\mathcal{A}$ is **only** allowed to see the training set

- **training error.**

  \[ E_{\text{train}}(g) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{i \in \mathcal{D}_{\text{train}}}[y_i \neq g(x_i)] \]

- **test error.**

  \[ E_{\text{test}}(g) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}}[y_i \neq g(x_i)] \]
the in-sample error and training error do not obey the Hoeffding inequality (without using, e.g., a union bound)

the test error does obey the Hoeffding inequality

\[ \mathbb{P}[|E_{\text{test}}(g) - E_{\text{out}}(g)| > \varepsilon] \leq 2 \exp\left(-2\varepsilon^2|D_{\text{test}}|\right). \]

so we can use the test error to predict generalization!
how to choose train and test sets?

- at random (eg, toss random coin with prob $p$)
- more data in the training set improves the model fit
- more data in test set helps determine how well the model will work out of sample
- usual rule of thumb: put about 20% of data into the test set