Machine Learning (CSE 446): Limits of Learning

Noah Smith

University of Washington
nasmith@cs.washington.edu

October 4, 2017

The Bayes Optimal Classifier

$$f^{(\mathsf{BO})}(x) = \operatorname*{argmax}_{y} \mathcal{D}(x, y)$$

The Bayes Optimal Classifier

$$f^{(\mathsf{BO})}(x) = \operatorname*{argmax}_y \mathcal{D}(x, y)$$

Theorem: The Bayes optimal classifier achieves minimal zero/one error ($\ell(y, \hat{y}) = [\![y \neq \hat{y}]\!]$) of any deterministic classifier.

Proof

Consider (deterministic) f' that claims to be better than $f^{(\mathsf{BO})}$ and x such that $f^{(\mathsf{BO})}(x) \neq f'(x)$.

Proof

Consider (deterministic) f' that claims to be better than $f^{(\mathsf{BO})}$ and x such that $f^{(\mathsf{BO})}(x) \neq f'(x)$.

Probability that f' makes an error on this input: $\left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f'(x))$.

Proof

Consider (deterministic) f' that claims to be better than $f^{(\mathrm{BO})}$ and x such that $f^{(\mathrm{BO})}(x) \neq f'(x)$.

Probability that f' makes an error on this input: $\left(\sum_y \mathcal{D}(x, y)\right) - \mathcal{D}(x, f'(x))$.

Probability that $f^{(\mathrm{BO})}$ makes an error on this input: $\left(\sum_y \mathcal{D}(x, y)\right) - \mathcal{D}(x, f^{(\mathrm{BO})}(x))$.

Proof

Consider (deterministic) f' that claims to be better than $f^{(\mathsf{BO})}$ and x such that $f^{(\mathsf{BO})}(x) \neq f'(x)$.

Probability that f' makes an error on this input: $\left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f'(x))$.

Probability that $f^{(\mathsf{BO})}$ makes an error on this input: $\left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f^{(\mathsf{BO})}(x))$.

By definition,

$$\mathcal{D}(x, f^{(\mathsf{BO})}(x)) = \max_y \mathcal{D}(x,y) \geq \mathcal{D}(x, f'(x))$$

$$\Rightarrow \left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f^{(\mathsf{BO})}(x)) \leq \left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f'(x))$$

Proof

Consider (deterministic) f' that claims to be better than $f^{(\mathsf{BO})}$ and x such that $f^{(\mathsf{BO})}(x) \neq f'(x)$.

Probability that f' makes an error on this input: $\left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f'(x))$.

Probability that $f^{(\mathsf{BO})}$ makes an error on this input: $\left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f^{(\mathsf{BO})}(x))$.

By definition,

$$\mathcal{D}(x, f^{(\mathsf{BO})}(x)) = \max_y \mathcal{D}(x,y) \geq \mathcal{D}(x, f'(x))$$

$$\Rightarrow \left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f^{(\mathsf{BO})}(x)) \leq \left(\sum_y \mathcal{D}(x,y)\right) - \mathcal{D}(x, f'(x))$$

This must hold for all x. Hence f' is no better than $f^{(\mathsf{BO})}$.

One Limit of Learning

You cannot do better than $\epsilon(f^{\mathsf{BO}})$.

Unavoidable Error

- Noise in the features (we don't want to "fit" the noise!)
- Insufficient information in the available features (e.g., incomplete data)
- No single correct label (e.g., inconsistencies in the data-generating process)

These have nothing to do with your choice of learning algorithm.

An Exercise

Following Daume (2017), chapter 2.

Class A

Class B

An Exercise

Following Daume (2017), chapter 2.

Test




Inductive Bias

Just as *you* had a tendency to focus on a certain type of function f, machine learning algorithms correspond to classes of functions ($\mathcal{F}$) and preferences within the class.

Inductive Bias

Just as *you* had a tendency to focus on a certain type of function f, machine learning algorithms correspond to classes of functions ($\mathcal{F}$) and preferences within the class.

E.g., shallow decision trees: "use a small number of features."

General Recipe

The cardinal rule of machine learning: **Don't touch your test data.**

If you follow that rule, this recipe will give you accurate information:

1. Split data into training, development, and test sets.
2. For different hyperparameter settings:
   2.1 Train on the training data using those hyperparameter values.
   2.2 Evaluate loss on development data.
3. Choose the hyperparameter setting whose model achieved the lowest development data loss.
   Optionally, retrain on the training and development data together.
4. Evaluate that model on test data.

Design Process for ML Applications

		example
1	real world goal	increase revenue
2	mechanism	show better ads
3	learning problem	will a user who queries q click ad a?
4	data collection	interaction with existing system
5	collected data	query q, ad a, $\pm$click
6	data representation	(q word, a word) pairs
7	select model family	decision trees up to 20
8	select training/dev. data	September
9	train and select hyperparameters	single decision tree
10	make predictions on test set	October
11	evaluate error	zero-one loss ($\pm$click)
12	deploy	\$?

Machine Learning (CSE 446): Geometry and Nearest Neighbors

Noah Smith

University of Washington
nasmith@cs.washington.edu

October 4, 2017

Features

Data derived from `https://archive.ics.uci.edu/ml/datasets/Auto+MPG`

mpg; cylinders; displacement; horsepower; weight; acceleration; year; origin

18.0	8	307.0	130.0	3504.	12.0	70	1
15.0	8	350.0	165.0	3693.	11.5	70	1
18.0	8	318.0	150.0	3436.	11.0	70	1
16.0	8	304.0	150.0	3433.	12.0	70	1
17.0	8	302.0	140.0	3449.	10.5	70	1
15.0	8	429.0	198.0	4341.	10.0	70	1
14.0	8	454.0	220.0	4354.	9.0	70	1
14.0	8	440.0	215.0	4312.	8.5	70	1
14.0	8	455.0	225.0	4425.	10.0	70	1
15.0	8	390.0	190.0	3850.	8.5	70	1
15.0	8	383.0	170.0	3563.	10.0	70	1
14.0	8	340.0	160.0	3609.	8.0	70	1
15.0	8	400.0	150.0	3761.	9.5	70	1
14.0	8	455.0	225.0	3086.	10.0	70	1
24.0	4	113.0	95.00	2372.	15.0	70	3
22.0	6	198.0	95.00	2833.	15.5	70	1
18.0	6	199.0	97.00	2774.	15.5	70	1
21.0	6	200.0	85.00	2587.	16.0	70	1
27.0	4	97.00	88.00	2130.	14.5	70	3
26.0	4	97.00	46.00	1835.	20.5	70	2
25.0	4	110.0	87.00	2672.	17.5	70	2
24.0	4	107.0	90.00	2430.	14.5	70	2

All features are represented as $\mathbb{R}$ values.

Side note: could convert discrete origin feature into three binary features as follows:

$$\begin{aligned} 1/\text{america} &\to (1,0,0) \\ 2/\text{europe} &\to (0,1,0) \\ 3/\text{asia} &\to (0,0,1) \end{aligned}$$

The "1–2–3" values suggest ordinality, which is misleading.

Instance x Becomes Vector $\mathbf{x}$

First example in the data, "Chevrolet Chevelle Malibu," becomes:

$$[8, 307.0, 130.0, 3504, 12.0, 70, 1, 0, 0]$$

"Buick Skylark 320" becomes:

$$[8, 350.0, 165.0, 3693, 11.5, 70, 1, 0, 0]$$

Euclidean Distance

General formula for the Euclidean distance between two d-length vectors:

$$
\begin{aligned}
dist(\mathbf{x}, \mathbf{x}') &= \sqrt{\sum_{j=1}^{d} (\mathbf{x}[j] - \mathbf{x}'[j])^2} \\
&= \left\|\mathbf{x} - \mathbf{x}'\right\|_2
\end{aligned}
$$

Euclidean Distance

General formula for the Euclidean distance between two d-length vectors:

$$dist(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{j=1}^{d} (\mathbf{x}[j] - \mathbf{x}'[j])^2}$$
$$= \|\mathbf{x} - \mathbf{x}'\|_2$$

The distance between the Chevrolet Chevelle Malibu and the Buick Skylark 320:

$$\sqrt{\begin{array}{c} (8-8)^2 + (307-350)^2 + (130-165)^2 + (3504-3693)^2 \\ +(12-11.5)^2 + (70-70)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 \end{array}}$$
$$= \sqrt{1849 + 1225 + 35721 + 0.25}$$
$$\approx 196.965$$

References I

Hal Daume. *A Course in Machine Learning (v0.9)*. Self-published at http://ciml.info/, 2017.