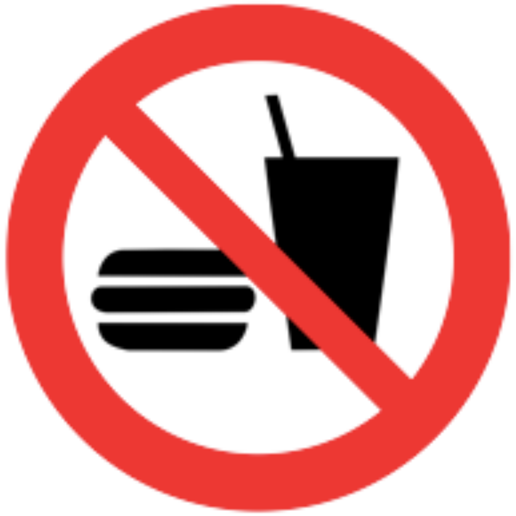
CS490DSC Data Science Capstone
Model Selection

Jean Honorio
Purdue University



Classification problems

- There are two parts to any classification task

1) Estimation: how to select the best classifier out of a particular set (for instance, linear classifiers)
2) Model selection: how to select the best set of classifiers (for instance, decision stumps, linear classifiers, 1-nearest neighbor)

- Both of these selections have to be made based on training data

- In order to grasp the concepts in this lecture better, we will introduce a very simple classifier: decision stump

Decision stump

- Consider a dataset of 6 samples, each with a single continuous attribute/feature ($x = x_1$) and class label (y)

x_1	y
0	+1
4	-1
-2	+1
1	+1
-3	-1
2	-1

- We would like to find a threshold β, and then classify all samples with attribute value x_1 above β as +1, and attribute value x_1 below β as -1 (or viceversa)

Decision stump

- Lets sort with respect to x

x_1	y
0	+1
4	-1
-2	+1
1	+1
-3	-1
2	-1

sort

x_1	y
-3	-1
-2	+1
0	+1
1	+1
2	-1
4	-1

- Lets use the classifier:

$$f(x) = \text{sign}(x_1 - \beta) = \begin{cases} +1, & \text{if } x_1 > \beta \\ -1, & \text{if } x_1 \le \beta \end{cases}$$

- How to find the threshold β ? Try all midpoints of x_1

Decision stump

- Lets use the classifier:

$$f(x) = \text{sign}(x_1 - \beta) = \begin{cases} +1, & \text{if } x_1 > \beta \\ -1, & \text{if } x_1 \le \beta \end{cases}$$

- Count the number of mistakes for all thresholds β

x_1	y	f(x)				
		β=-2.5	β=-1	β=0.5	β=1.5	β=3
-3	-1	-1	-1	-1	-1	-1
-2	+1	+1	-1	-1	-1	-1
0	+1	+1	+1	-1	-1	-1
1	+1	+1	+1	+1	-1	-1
2	-1	+1	+1	+1	+1	-1
4	-1	+1	+1	+1	+1	+1
# mistakes		2	3	4	5	4

Decision stump

- Lets use the classifier:

$$f(x) = \text{sign}(\beta - x_1) = \begin{cases} +1, & \text{if } x_1 < \beta \\ -1, & \text{if } x_1 \geq \beta \end{cases}$$

- Count the number of mistakes for all thresholds β

x_1	y	f(x)				
		β=-2.5	β=-1	β=0.5	β=1.5	β=3
-3	-1	+1	+1	+1	+1	+1
-2	+1	-1	+1	+1	+1	+1
0	+1	-1	-1	+1	+1	+1
1	+1	-1	-1	-1	+1	+1
2	-1	-1	-1	-1	-1	+1
4	-1	-1	-1	-1	-1	-1
# mistakes		4	3	2	1	2

Decision stump

- Thus our best decision stump classifier:

$$f(x) = \text{sign}(1.5 - x_1) = \begin{cases} +1, & \text{if } x_1 < 1.5 \\ -1, & \text{if } x_1 \geq 1.5 \end{cases}$$

- Remember that we consider all classifiers of the form:

$$f(x) = \text{sign}(x_1 - \beta) = \begin{cases} +1, & \text{if } x_1 > \beta \\ -1, & \text{if } x_1 \leq \beta \end{cases}$$

$$f(x) = \text{sign}(\beta - x_1) = \begin{cases} +1, & \text{if } x_1 < \beta \\ -1, & \text{if } x_1 \geq \beta \end{cases}$$

for any real value β

- Although these are simple classifiers, the set of decision stump classifiers is uncountable (there are as "many" as real values)

VC dimension

- The Vapnik-Chervonenkis (VC) dimension allows us to understand the complexity of a model class (a set of classifiers) without having to "count" how many classifiers there are, for instance:
  - the set of decision stump classifiers
  - the set of linear classifiers
  - the set of 1-nearest neighbor classifiers

- Instead we count the number of ways in which a dataset can be classified.

VC Dimension of decision stump

- Lets take the sorted dataset we used before
- Consider decision stump classifiers with all values of β that would lead to different ways of classifying the samples

$$f(x) = \operatorname{sign}(x_1 - \beta) = \begin{cases} +1, & \text{if } x_1 > \beta \\ -1, & \text{if } x_1 \le \beta \end{cases}$$

$$f(x) = \operatorname{sign}(\beta - x_1) = \begin{cases} +1, & \text{if } x_1 < \beta \\ -1, & \text{if } x_1 \ge \beta \end{cases}$$

x_1	f(x)					
	β=-2.5	β=-1	β=0.5	β=1.5	β=3	β=∞
-3	-1	-1	-1	-1	-1	-1
-2	+1	-1	-1	-1	-1	-1
0	+1	+1	-1	-1	-1	-1
1	+1	+1	+1	-1	-1	-1
2	+1	+1	+1	+1	-1	-1
4	+1	+1	+1	+1	+1	-1

x_1	f(x)					
	β=-2.5	β=-1	β=0.5	β=1.5	β=3	β=∞
-3	+1	+1	+1	+1	+1	+1
-2	-1	+1	+1	+1	+1	+1
0	-1	-1	+1	+1	+1	+1
1	-1	-1	-1	+1	+1	+1
2	-1	-1	-1	-1	+1	+1
4	-1	-1	-1	-1	-1	+1

- We highlight (in blue) one way of classifying the 6 samples
- We have 12 different ways of classifying the 6 samples

VC dimension of decision stump

- In general, the set of decision stump classifiers lead to 2n different ways of classifying n samples
  - We classify the n samples as -1's followed by +1's
  - We also classify the n samples as +1's followed by -1's

VC dimension of decision stump

- In general, the set of decision stump classifiers lead to 2n different ways of classifying n samples
  - We classify the n samples as -1's followed by +1's
  - We also classify the n samples as +1's followed by -1's

- More complex classifiers would lead to more than 2n different ways of classifying n samples
- The most complex classifiers would lead to 2^n different ways of classifying n samples
  - There are 2^n different vectors of size n with each entry being either +1 or -1

VC dimension of decision stump

- In general, the set of decision stump classifiers lead to 2n different ways of classifying n samples
  - We classify the n samples as -1's followed by +1's
  - We also classify the n samples as +1's followed by -1's

- More complex classifiers would lead to more than 2n different ways of classifying n samples
- The most complex classifiers would lead to 2^n different ways of classifying n samples
  - There are 2^n different vectors of size n with each entry being either +1 or -1

- More complex classifiers are not always better, as we will see later

VC dimension

- The Vapnik-Chervonenkis (VC) dimension is the maximum number of samples n that can be classified in any possible way (that is, 2^n ways) by a model class (a set of classifiers)

VC dimension of decision stump

- The Vapnik-Chervonenkis (VC) dimension is the maximum number of samples n that can be classified in any possible way (that is, 2^n ways) by a model class (a set of classifiers)

- Recall that decision stump classifiers lead to 2n different ways of classifying n samples
- Find the maximum n for which $2n = 2^n$
- The VC dimension is VC = 2

n	2n	2^n
1	2	2
2	4	4
3	6	8

VC dimension of decision stump

- The Vapnik-Chervonenkis (VC) dimension is the maximum number of samples n that can be classified in any possible way (that is, 2^n ways) by a model class (a set of classifiers)

- Recall that decision stump classifiers lead to 2n different ways of classifying n samples
- Find the maximum n for which $2n = 2^n$
- The VC dimension is VC = 2

n	2n	2^n
1	2	2
2	4	4
3	6	8

- For more intuition, see the 2^n ways of classifying n samples

n=1

+1	-1

n=2

+1	+1	-1	-1
+1	-1	+1	-1

n=3

+1	+1	+1	+1	-1	-1	-1	-1
+1	+1	-1	-1	+1	+1	-1	-1
+1	-1	+1	-1	+1	-1	+1	-1

2 ways (2^3-2*3 = 2) of classifying (in red) are not -1's followed by +1's, neither +1's followed by -1's

VC dimension

- The Vapnik-Chervonenkis (VC) dimension is the maximum number of samples n that can be classified in any possible way (that is, 2^n ways) by a model class (a set of classifiers)

- The VC dimension of the set of decision stumps is $VC = 2$
- The VC dimension of the set of linear classifiers in d dimensions (R^d) without offset parameter, is $VC = d$
- The VC dimension of the set of linear classifiers in d dimensions (R^d) with offset parameter, is $VC = d + 1$
- The VC dimension of the set of 1-nearest neighbor classifiers is $VC = \infty$

Mean versus expectation

- Consider a Bernoulli random variable X with $p = 0.5$
  - $X = 1$ with probability p
  - $X = 0$ with probability $1 - p$

- The expected value of X is:

$$\begin{aligned} E[X] &= 1 \times P(X = 1) + 0 \times P(X = 0) \\ &= 1 \times p + 0 \times (1 - p) \\ &= p \end{aligned}$$

- Assume we have a dataset of n bits: $x_1, x_2, \ldots, x_n$
- We can compute the mean:

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Mean versus expectation

```
import numpy as np
def example_bernoulli(n):
  z = np.random.randint(0,2,n)
  return 1.0/n * np.sum(z)

>>> example_bernoulli(10)
0.8
>>> example_bernoulli(100)
0.44
>>> example_bernoulli(10000)
0.5138
```

Returns n random integers >= 0 and < 2, each value with equal probability.
In this case (0 or 1) then p = 0.5 in the Bernoulli distribution

Computes average

Training error

- For computational purposes, we consider data to be constant, but data is a random variable!
- There is an unknown data distribution P
- The training set has n samples: $\underline{x}_1, y_1, \ldots, \underline{x}_n, y_n$
  Samples $\underline{x}_i, y_i$ are independent, with probability distribution P
- The training error is:

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}(y_i, f(\underline{x}_i))$$

where f is a classifier and $\text{Loss}(y, y') = \begin{cases} 1, & y \neq y' \\ 0, & \text{o.w.} \end{cases}$

- Given a classifier f and n samples, we can compute the training error $\hat{R}_n(f)$

Test error

- The test error is the expected value of the error
- The training error is an estimate (an average of a finite number of samples) of the expected value
- Intuitively speaking, the test error is the error when using an infinite number of samples
- The test error is:

$$R_P(f) = \int_{\underline{x},y} \mathrm{Loss}(y, f(\underline{x}))\; P(\underline{x}, y)\; d\underline{x}\, dy$$

$$= \mathrm{E}_P[\ \mathrm{Loss}(y, f(\underline{x}))\ ]$$

- Given a classifier f, we cannot compute the test error $R_P(f)$ because the data distribution P is unknown

Training and test error

- While we can only compute the training error $\hat{R}_n(f)$, we are truly interested on the test error $R_P(f)$, because the test error is the true measure of how we will perform on unseen data

- Under-fitting: large training error $\hat{R}_n(f)$ and test error $R_P(f)$

- Over-fitting: small training error $\hat{R}_n(f)$, large test error $R_P(f)$

Generalization

- We cannot compute $R_P(f)$, but we can bound it!
- Consider a model class (a set of classifiers) with Vapnik-Chervonenkis dimension: VC
- Vapnik 1979: Without any knowledge of the data distribution P, with probability at least $1-\delta$ over the choice of the training set, for all classifiers f in the model class:

$$R_P(f) \leq \hat{R}_n(f) + \sqrt{\frac{VC(\log(2n/VC)+1)+\log(4/\delta)}{n}}$$

Generalization

- We cannot compute $R_P(f)$, but we can bound it!
- Consider a model class (a set of classifiers) with Vapnik-Chervonenkis dimension: VC
- Vapnik 1979: Without any knowledge of the data distribution P, with probability at least $1-\delta$ over the choice of the training set, for all classifiers f in the model class:

$$R_P(f) \leq \hat{R}_n(f) + \sqrt{\frac{VC(\log(2n/VC)+1)+\log(4/\delta)}{n}}$$

- For instance, for decision stumps: $VC = 2$, let $\delta = 0.1$, With probability at least $1-\delta = 0.9$:

$$R_P(f) \leq \hat{R}_n(f) + \sqrt{\frac{2(\log n + 1)+\log(40)}{n}}$$

Structural risk minimization

- Choose the model class (for instance, decision stumps versus linear classifiers) with best guarantee of generalization:

$$\hat{R}_n(f) + \sqrt{\frac{VC(\log(2n/VC)+1)+\log(4/\delta)}{n}}$$

Large for simple classifiers,
small for complex classifiers

Small for simple classifiers (small VC),
large for complex classifiers (large VC)

Large for small n,
small for large n