Machine Learning Crash Course

Introduction to Classification

Dr. Henrich Matzinger and Dr. Greg Mayer
Machine Learning and AI Crash Course
December 2019

School of Mathematics
Georgia Institute of Technology, Atlanta, GA

Overview

Objective

Introduce a classification algorithm that we will use for text classification later in our crash course

Topics

1. classification rules
2. training data
3. best classification rule on training data

Catching Fish

Suppose:

- A boat on the sea catches only salmon and tuna
- A robot on the boat classifies each fish by its size: small, medium, or large
- The robot cannot tell whether fish are salmon or tuna

Given a size, how can we best predict the type of fish?

A Probability Model

Notation

- sizes: 1 = small, 2 = medium, 3 = large
- X denotes size of fish
- Y denotes the **class**, or type of of fish

Assumptions

- there is an underling **probability model**, or "probability distribution" or in statistical parlance "the population distribution"
- we are given a **joint probability table**

	$X = 1$	X=2	$X = 3$
tuna	$P(Y = \text{tuna}, X = 1)$	$P(Y = \text{tuna}, X = 2)$	$P(Y = \text{tuna}, X = 3)$
salmon	$P(Y = \text{salmon}, X = 1)$	$P(Y = \text{salmon}, X = 2)$	$P(Y = \text{salmon}, X = 3)$

Joint Probability Table Example

- Suppose the underlying probability model is:

	1	2	3
`tuna`	0.1	0.2	0.3
`salmon`	0.2	0.1	0.1

(1)

  For example, $P(Y = salmon, X = 1) = 20\%$, meaning that in the waters in which we are fishing, 20% of the fish are salmon of size 1.
- Similarly, $P(Y = tuna, X = 3) = 0.3$, meaning that 30% of the fish in the water are $tuna$ of size 3

What if we wanted to predict the type of fish given its size?

A Decision Rule

- Our **decision rule** can be based on choosing the class which has highest probability given its size
- Suppose our robot classifies a fish as being size 2. This fish is then twice as likely to be a tuna:

$$P(Y = tuna \,|\, X = 2) = \frac{0.2}{0.3} = \frac{2}{3}$$

and

$$P(Y = salmon \,|\, X = 2) = \frac{0.1}{0.3} = \frac{1}{3}$$

So, if we catch a fish of size 2, our decision rule states that we classify that fish as tuna.

Bayse Classifier

- In fancy machine learning parlance, classification rules are **classifiers**
- The set $\{1, 2, 3\}$ would be the **feature space**
- $tuna$ and $salmon$ are the **classes**
- The best rule is denoted by g^* and is called a *Bayes classifier*

Formally the Bayes classifier can be defined by:

$$g^*(x) = tuna \quad \texttt{if and only if} \quad \frac{\texttt{P}(\texttt{Y} = \texttt{tuna} | \texttt{X} = \texttt{x})}{\texttt{P}(\texttt{Y} = \texttt{salmon} | \texttt{X} = \texttt{x})} \geq 1 \qquad (2)$$

Prior Probability

Let

- $\pi_{\texttt{tuna}}$ = probability that a given fish from the water is a tuna
- $\pi_{\texttt{salmon}}$ = probability that a given fish from the water is a salmon

Or,

$$\pi_{\texttt{tuna}} := P(Y = \texttt{tuna})$$
$$\pi_{\texttt{salmon}} := P(Y = \texttt{salmon})$$

Another way to rewrite the equations leading to the Bayse classifier, is obtained by using Bayse theorem and is:

$$g^*(x) = tuna \;\; \texttt{if and only if} \;\; \frac{\pi_{\texttt{tuna}} \cdot \texttt{P}(\texttt{X} = \texttt{x} | \texttt{Y} = \texttt{tuna})}{\pi_{\texttt{salmon}} \cdot \texttt{P}(\texttt{X} = \texttt{x} | \texttt{Y} = \texttt{salmon})} \geq 1 \quad (3)$$

Best Decision Rule

In the present case, the best classifier is given by

$$g^*(1) = salmon, g^*(2) = tuna, g^*(3) = tuna$$

Our optimal decision rule is represented in our table as bold entries:

	1	2	3
`tuna`	0.1	**0.2**	**0.3**
`salmon`	**0.2**	0.1	0.1

(4)

Probability of Missclassification

Our optimal decision being represented by the green entries:

	1	2	3
`tuna`	0.1	0.2	0.3
`salmon`	0.2	0.1	0.1

(5)

The **misclassifiction probability** is the sum of the red entries:

$$\begin{aligned} \texttt{misclassification probability of } g^* &= P(g^*(X) \neq Y) \\ &= 0.1 + 0.1 + 0.1 \\ &= 0.3 \end{aligned}$$

This means that in the long run, our robot miscalssifies 30% of the fish.

Training Data

Underlying Probability Model Usually Unknown

- on general, underlying probabilities are not exactly known
- but if we can catch some fish, label them manually as salmon or tuna and then estimate the probabilities given in Table 5.
- the data we use to determine a classification rule is called a **training sample**

Estimating Probabilities

Suppose we catch 100 fish and obtain the frequency table

	1	2	3
`tuna`	4	10	45
`salmon`	15	6	20

(6)

Four fish are tuna of size 1, so we make the estimate:

$$\hat{P}(Y = tuna, X = 1) = 0.04$$

Fifteen salmon of size 1, which leads to our estimate:

$$\hat{P}(Y = salmon, X = 1) = 0.15$$

So, we have the estimated probabilities given in the table

	1	2	3
`tuna`	0.04	**0.10**	**0.45**
`salmon`	**0.15**	0.06	0.2

(7)

Best classifiction rule on given data

The classification rule (classifier) which we chose is

$$g(1) = salmon, g(2) = tuna, g(3) = tuna$$

is based on the estimated probabilities given in the table

	1	2	3
`tuna`	0.04	**0.10**	**0.45**
`salmon`	**0.15**	0.06	0.2

(8)

This is the rule that is best at classifying our data

Essentially:

- we do not know true probabilities but we have their estimates.
- we act as if the estimates were the true probabilities and create a best prediction rule from them

Best Classification Rules

Overfitting

- When we have many more options for the feature and not enough data we encounter **overfitting**
- Suppose our robot can identify many more sizes and we collect the frequency table below

	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
`tuna`	0	**1**	0	**1**	**3**	**1**	0	0	**1**	0	0
`salmon`	0	0	**1**	0	0	0	**1**	**2**	0	**3**	**2**

- we can create a rule that classifies fish **in our training data** correctly 100 % of the time
- if we then catch another fish and use such a rule, what problems might we encounter?

Formal Random Variables

Again, same frequency table:

	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
`tuna`	0	**1**	0	**1**	**3**	**1**	0	0	**1**	0	0
`salmon`	0	0	**1**	0	0	0	**1**	**2**	0	**3**	**2**

Consider the classification rule:

$$\texttt{size} \geq 2 \implies \texttt{salmon}$$

Based on our frequency table, this classifies correctly 13 out of 16

- Let X_i be the size of the i-th fish and let Y_i be its class.
- Our training sample is the data $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$
- Let $Z_i = 1$ if ith fish gets classified correctly with our rule and $Z_i = 0$ otherwise. In our case

$$Z_1 + Z_2 + \ldots + Z_{16} = 13$$

Probability of Correct Classification

- Then, we can estimate the probability of correct classification:

$$\hat{P}(g(X) = Y) = \frac{Z_1 + Z_2 + \ldots + Z_n}{n}$$

- This is estimating the parameter p in a binomial, which if n is not very small is quite precise.
- So we should be able to see which classification rule are best by using our estimate $\hat{P}$

Solution to Overfitting

- to avoid overfitting we can use a small number of rules to identify **best classification rule**
- In our data, if we only consider classification rules of the type

$$X \geq \texttt{constant} \implies \texttt{salmon}$$

what value for `constant` leads to least classification error in the training data?

	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
`tuna`	0	**1**	0	**1**	**3**	**1**	0	0	**1**	0	0
`salmon`	0	0	**1**	0	0	0	**1**	**2**	0	**3**	**2**

Justification for Best Classification Rule

Assume that

1. size of tuna is normal with expectation μ_t
2. size of salmon normal with expectation μ_s
3. distributions both have standard deviation 1

Then, the optimal rule:

$$P(x \,|\, tuna)\pi_t > P(x \,|\, salmon)\pi_s \implies tuna$$

is equivalent to

$$\pi_t \cdot \exp(-(x-\mu_t)^2) > \pi_s \cdot \exp(-(x-\mu_s)^2)$$

solving for x

$$x \geq \frac{\mu_s^2 - \mu_t^2}{\mu_t - \mu_s} + \log(\pi_s) - \log(\pi_t) = \texttt{constant}$$

Thus $x > \texttt{constant}$ is a reasonable approach to classifying our data