CSCI567 Machine Learning (Fall 2016)

Dr. Yan Liu

yanliu.cs@usc.edu

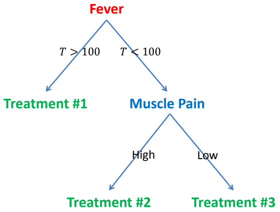
September 13, 2016

Outline

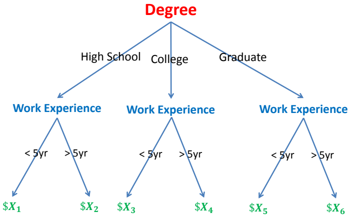
1. Decision tree
   - Examples
   - Algorithm
2. Naive Bayes

Many decisions are tree structures

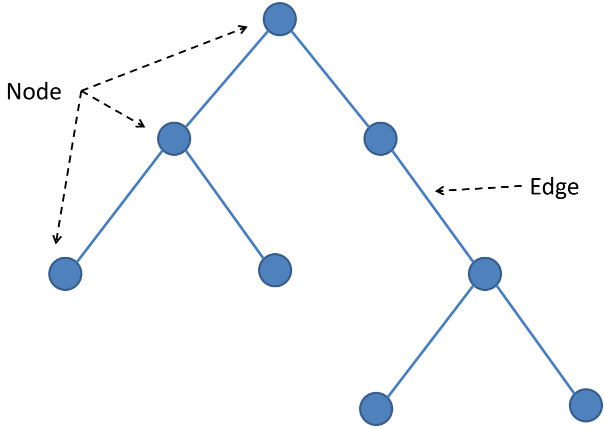
Medical treatment



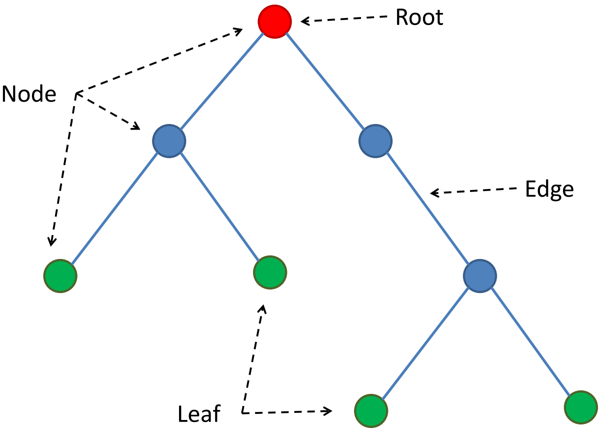
Salary in a company



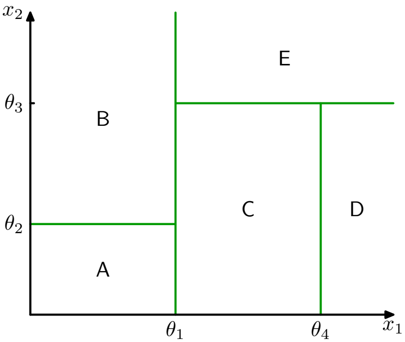
What is a Tree?

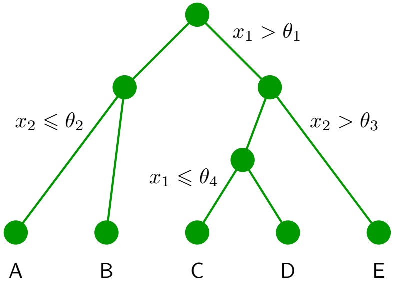


Special Names for Nodes in a Tree



A tree partitions the feature space

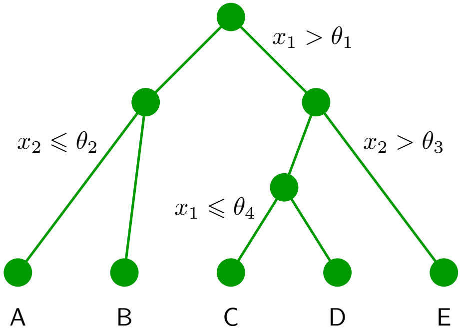




Learning a tree model

Three things to learn:

1. The structure of the tree.
2. The threshold values (θ_i).
3. The values for the leafs $(A, B, \ldots)$.



A tree model for deciding where to eat

Choosing a restaurant

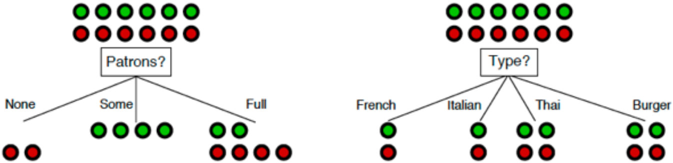
(Example from Russell & Norvig, AIMA)

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Classification of examples is positive (T) or negative (F)

First decision: at the root of the tree

Which attribute to split?



Patrons? is a better choice—gives **information** about the classification

Idea: use information gain to choose
which attribute to split

How to measure information gain?

Idea:

Gaining information reduces uncertainty

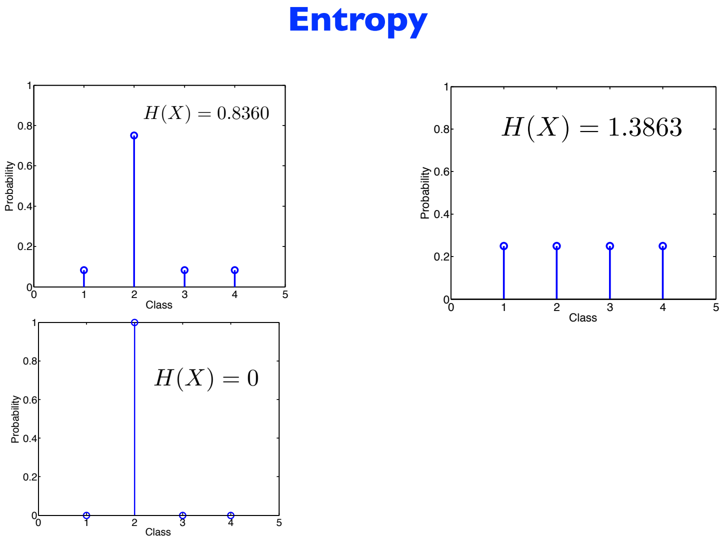
the base can be 2 , though it is not essential (if the base is 2, the unit of the entropy is called "bit")

Use to entropy to measure uncertainty

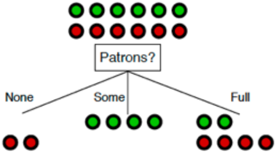
If a random variable X has K different values, a_1, a_2, ...a_K, it is entropy is given by

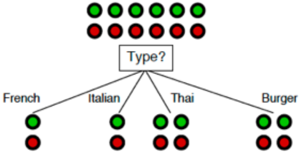
$$H[X] = -\sum_{k=1}^{K} P(X = a_k) \log P(X = a_k)$$

Examples of computing entropy



Which attribute to split?





Patrons? is a better choice—gives **information** about the classification

Patron vs. Type?

By choosing Patron, we end up with a partition (3 branches) with smaller entropy, ie, smaller uncertainty (0.45 bit)

By choosing Type, we end up with uncertainty of 1 bit.

Thus, we choose Patron over Type.

Uncertainty if we go with "Patron"

For "None" branch

$$-\left(\frac{0}{0+2}\log\frac{0}{0+2}+\frac{2}{0+2}\log\frac{2}{0+2}\right)=0$$

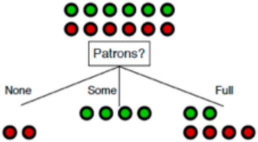
For "Some" branch

$$-\left(\frac{4}{4+0}\log\frac{4}{4+0}+\frac{4}{4+0}\log\frac{4}{4+0}\right)=0$$

For "Full" branch

$$-\left(\frac{2}{2+4}\log\frac{2}{2+4}+\frac{4}{2+4}\log\frac{4}{2+4}\right)\approx 0.9$$

For choosing "Patrons"



weighted average of each branch: this quantity is called conditional entropy

$$\frac{2}{12}*0+\frac{4}{12}*0+\frac{6}{12}*0.9=0.45$$

Conditional entropy

Definition. Given two random variables X and Y

$$H[Y|X] = \sum_k P(X = a_k) H[Y|X = a_k]$$

In our example

X: the attribute to be split

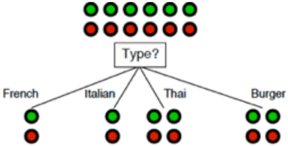
Y: Wait or not

When H[Y] is fixed, we need only to compare conditional entropy

Relation to information gain

$$\mathrm{GAIN} = H[Y] - H[Y|X]$$

Conditional entropy for Type



For "French" branch

$$-\left(\frac{1}{1+1}\log\frac{1}{1+1}+\frac{1}{1+1}\log\frac{1}{1+1}\right)=1$$

For "Italian" branch

$$-\left(\frac{1}{1+1}\log\frac{1}{1+1}+\frac{1}{1+1}\log\frac{1}{1+1}\right)=1$$

For "Thai" and "Burger" branches

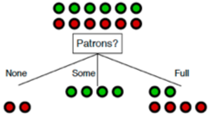
$$-\left(\frac{2}{2+2}\log\frac{2}{2+2}+\frac{2}{2+2}\log\frac{2}{2+2}\right)=1$$

For choosing "Type"

weighted average of each branch:

$$\frac{2}{12}*1+\frac{2}{12}*1+\frac{4}{12}*1+\frac{4}{12}*1=1$$

next split?

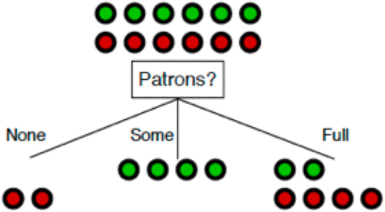


We will look only at the 6 instances with Patrons == Full

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Classification of examples is positive (T) or negative (F)

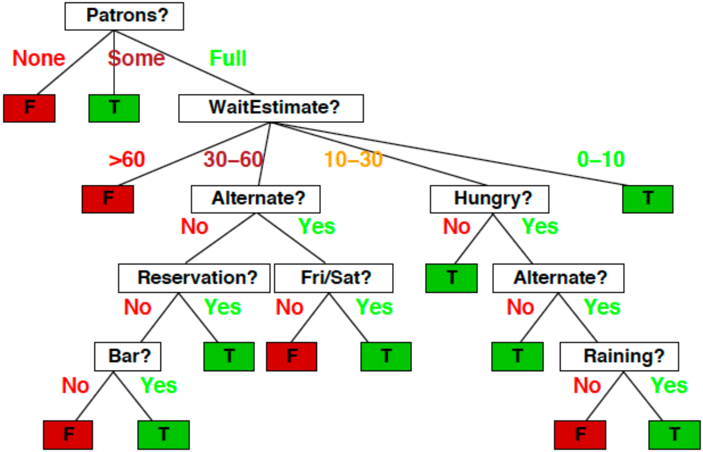
Do we split on "Non" or "Some"?



No, we do not

The decision is deterministic, as seen from the training data

Greedily we build the tree and get this



How deep should we continue to split?

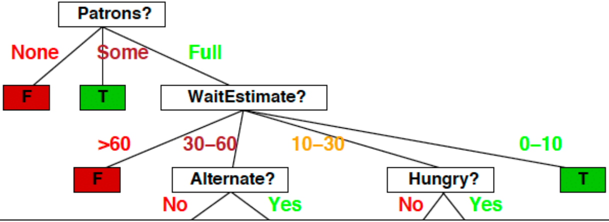
We should be very careful about this

Eventually, we can get all training examples right. But is that what we want?

The maximum depth of the tree is a hyperparameter and should not be tuned by training data — this is to prevent overfitting (we will discuss later)

Control the size of the tree

We would prune to have a smaller one



If we stop here, not all training sample would be classified correctly.

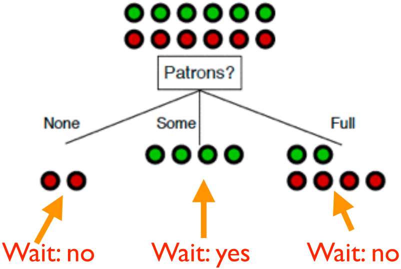
More importantly, how do we classify a new instance?

We label the leaves of this smaller tree with the majority of training samples' labels

Example

Example

We stop after the root (first node)



Splitting and Stopping Criteria

For every leaf m, define the node impurity $Q(m)$ as:

Misclassification error	$\frac{1}{N_m}\sum_{i\in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk}$.
Gini Index	$\sum_{k\neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$.
Cross-entropy	$-\sum_{k=1}^{K} \hat{p}_{mk}\log\hat{p}_{mk}$.

The **Misclassification Error** is less sensitive to changes in class probability:

$\Rightarrow$ Use Gini Index or Cross-entropy for growing T_0,

$\Rightarrow$ Use Misclassification Error for pruning T_0 and finding T.

Summary of learning trees

Other ideas in learning trees

- There are other ways of splitting attributes, such as Gini index.
- There are other fast ways of learning tree models.
- There are approaches of learning an ensemble of tree models (more on this later)

Advantages of using trees

- The models are transparent: easily interpretable by human (as long as the tree is not too big)
- It is parametric thus compact: unlike NNC, we do not have to carry our training instances around

Outline

A daily battle

Great news: I will be rich!

FROM THE DESK OF MR. AMINU SALEH
DIRECTOR, FOREIGN OPERATIONS DEPARTMENT
AFRI BANK PLC
Afribank Plaza,
14th Floormoney344.jpg
51/55 Broad Street,
P.M.B 12021 Lagos-Nigeria

Attention: Honorable Beneficiary,

IMMEDIATE PAYMENT NOTIFICATION

It is my modest obligation to write you th owed payment through our most respected
financial institution (AFRI BANK PLC). I a tions Department, AFRI Bank Plc, NIGERIA.
The British Government, in conjunction w NITED NATIONS ORGANIZATION on
foreign payment matters, has empowered tion, to handle all foreign payments and
release them to their appropriate benefici deral Reserve Bank.

To facilitate the process of this transactio tion below:

1) Your full Name and Address:
2) Phones, Fax and Mobile No. :
3) Profession, Age and Marital Status:
4) Copy of any valid form of your Identification:

How to tell spam from ham?

FROM THE DESK OF MR. AMINU SALEH
DIRECTOR, FOREIGN OPERATIONS DEPARTMENT
AFRI BANK PLC
Afribank Plaza,
14th Floormoney344.jpg
51/55 Broad Street,
P.M.B 12021 Lagos-Nigeria

Attention: Honorable Beneficiary,

IMMEDIATE PAYMENT NOTIFICATION VALUED AT **US$10 MILLION**

Dear Dr.Sha,

I just would like to remind you of your scheduled presentation for CS597, Monday October 13, 12pm at OHE122.

If there is anything that you would need, please do not hesitate to contact me.

sincerely,

Christian Siagian

Intuition

How human solves the problem?

Spam emails

concentrated use of a lot of words like "money", "free", "bank account", "viagara"

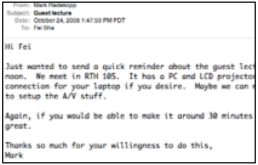
Ham emails

word usage pattern is more spread out

Simple strategy: count the words

Bag-of-word representation
of documents (and textual data)

$$\begin{pmatrix} \text{free} & 100 \\ \text{money} & 2 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$



From: Mark Hadekopp
Subject: Guest lecture
Date: October 24, 2008 1:47:59 PM PDT
To: Fei Sha

Hi Fei

Just wanted to send a quick reminder about the guest lect
noon. We meet in RTH 105. It has a PC and LCD projector
connection for your laptop if you desire. Maybe we can
to setup the A/V stuff.

Again, if you would be able to make it around 30 minutes
great.

Thanks so much for your willingness to do this,
Mark

$$\begin{pmatrix} \text{free} & 1 \\ \text{money} & 1 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

Weighted sum of those telltale words

$$\begin{pmatrix} \text{free} & 100 \\ \text{money} & 2 \\ \vdots & \vdots \\ \text{account} & 2 \\ \vdots & \vdots \end{pmatrix}$$

$$\begin{pmatrix} 100 \times 0.2 \\ 2 \times 0.3 \\ \vdots \\ 2 \times 0.3 \\ \vdots \end{pmatrix}$$

different weights for spam and ham: representing how compatible the word usage pattern is to different category

= 3.2

$$\begin{pmatrix} 100 \times 0.01 \\ 2 \times 0.02 \\ \vdots \\ 2 \times 0.01 \\ \vdots \end{pmatrix}$$

= 1.03

Our intuitive model of classification

Assign weight to each word

Compute compatibility score to "spam"

of "free" x a_{free} + # of "account" x $a_{account}$ + # of "money" x a_{money}

Compute compatibility score to "ham":

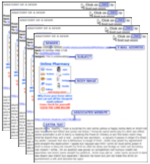
of "free" x b_{free} + # of "account" x $b_{account}$ + # of "money" x b_{money}

Make a decision:

if spam score > ham score then spam

else ham

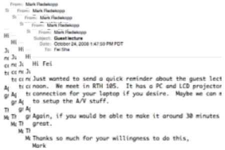
How we get the weights?



Learning from experience

get a lot of spams

get a lot of hams



But what to optimize?

A probabilistic modeling perspective

Naive Bayes model for identifying spams

Class label: binary

y = {spam, ham}

Features: word counts in the document (Bag-of-word)

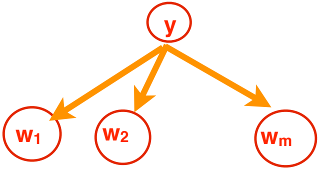
Ex: x = {('free', 100), ('lottery', 10), ('money', 10), , ('identification', 1)...}

Each pair is in the format of
(w_i, #w_i), namely, a unique word in the dictionary,
and the number of times it shows up

Naive Bayes model for identifying spams

$$
\begin{aligned}
p(x|y) &= p(w_1|y)^{\#w_1} p(w_2|y)^{\#w_2} \cdots p(w_m|y)^{\#w_m} \\
&= \prod_i p(w_i|y)^{\#w_i}
\end{aligned}
$$

These conditional probabilities are model parameters



Spam writer's vocabulary

Features: word counts in the document

Ex: x = {('free', 100), ('identification', 2), ('lottery', 10), ('money', 10), ...}

Model: Naive Bayes (NB)

$$\begin{aligned} p(x|\text{spam}) = p(\text{'free'}|\text{spam})^{100} p(\text{'identification'}|\text{spam})^{2} \\ p(\text{'lottery'}|\text{spam})^{10} p(\text{'money'}|\text{spam})^{10} \cdots \\ \neq p(x|\text{ham}) \end{aligned}$$

Parameters to be estimated:
p('free'|spam), p('free'|ham),etc

Naive Bayes

Why the name "naive"?

Strong assumption of conditional independence:

$$p(w_i, w_j | y) = p(w_i | y) p(w_j | y)$$

How to estimate model parameters?

Use maximum likelihood estimation (soon)

Does this correspond to our intuitive model of classification?

Yes. It does!

Let us consider the Bayes optimal classifier under this assumed probabilistic distribution

$$\begin{aligned} p(x|y) &= p(w_1|y)^{\#w_1} p(w_2|y)^{\#w_2} \cdots p(w_m|y)^{\#w_m} \\ &= \prod_i p(w_i|y)^{\#w_i} \end{aligned}$$

Naive Bayes classification rule

For any document x, we need to compute

$$p(\text{spam}|x) \quad \text{and} \quad p(\text{ham}|x)$$

Naive Bayes classification rule

For any document x, we need to compute

$$p(\text{spam}|x) \quad \text{and} \quad p(\text{ham}|x)$$

Using Bayes rule, this gives rise to

$$p(\text{spam}|x) = \frac{p(x|\text{spam})p(\text{spam})}{p(x)}, \quad p(\text{ham}|x) = \frac{p(x|\text{ham})p(\text{ham})}{p(x)}$$

Naive Bayes classification rule

For any document x, we need to compute

$$p(\text{spam}|x) \quad \text{and} \quad p(\text{ham}|x)$$

Using Bayes rule, this gives rise to

$$p(\text{spam}|x) = \frac{p(x|\text{spam})p(\text{spam})}{p(x)}, \quad p(\text{ham}|x) = \frac{p(x|\text{ham})p(\text{ham})}{p(x)}$$

It is convenient to compute the logarithms, so we need only to compare

$$\log[p(x|\text{spam})p(\text{spam})] \quad \text{versus} \quad \log[p(x|\text{ham})p(\text{ham})]$$

as the denominators are the same

Classifier in the linear form of compatibility scores

$$\log[p(x|\text{spam})p(\text{spam})] = \log\left[\prod_i p(w_i|\text{spam})^{\#w_i} p(\text{spam})\right] \tag{1}$$

$$= \sum_i \#w_i \log p(w_i|\text{spam}) + \log p(\text{spam}) \tag{2}$$

Classifier in the linear form of compatibility scores

$$\log[p(x|\textsf{spam})p(\textsf{spam})] = \log\left[\prod_i p(w_i|\textsf{spam})^{\#w_i} p(\textsf{spam})\right] \quad (1)$$

$$= \sum_i \#w_i \log p(w_i|\textsf{spam}) + \log p(\textsf{spam}) \quad (2)$$

Similarly, we have

$$\log[p(x|\textsf{ham})p(\textsf{ham})] = \sum_i \#w_i \log p(w_i|\textsf{ham}) + \log p(\textsf{ham})$$

Namely, we are back to the idea of comparing weighted sum of # of word occurrences!

$\log p(\textit{spam})$ *and* $\log p(\textit{ham})$ *are called "priors" or "bias" (they are not in our intuition but they are crucially needed)*

Mini-summary

What we have shown

By making a probabilistic model (i.e., Naive Bayes), we are able to derive a decision rule that is consistent with our intuition

Our next step is to leverage this link to learn the rule from the data

Formal definition of Naive Bayes

General case

Given a random variable $X \in \mathbb{R}^{\mathsf{D}}$ and a dependent variable $Y \in [\mathsf{C}]$, the Naive Bayes model defines the joint distribution

$$P(X = x, Y = y) = P(Y = y)P(X = x|Y = y) \quad (3)$$

$$= P(Y = y)\prod_{d=1}^{\mathsf{D}} P(X_d = x_d|Y = y) \quad (4)$$

Special case (i.e., our model of spam emails)

Assumptions

- All X_d are categorical variables from the same domain — $x_d \in [\mathsf{K}]$, for example, the index to the unique words in a dictionary.
- $P(X_d = x_d | Y = y)$ depends only on the value of x_d, not d itself, namely, orders are not important (thus, we only need to count).

Simplified definition

$$P(X = x, Y = c) = P(Y = c) \prod_k P(k|Y = c)^{z_k} = \pi_c \prod_k \theta_{ck}^{z_k}$$

where z_k is the number of times k in x.

Note that we only need to enumerate in the product, the index to the x_d's possible values. On the previous slide, however, we enumerate over d as we do not have the assumption there that order is not important.

Learning problem

Training data

$$\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{\mathsf{N}} \rightarrow \mathcal{D} = \{(\{z_{nk}\}_{k=1}^{\mathsf{K}}, y_n)\}_{n=1}^{\mathsf{N}}$$

Goal

Learn $\pi_c, c = 1, 2, \cdots, \mathsf{C}$, and $\theta_{ck}, \forall c \in [\mathsf{C}], k \in [\mathsf{K}]$ under the constraint

$$\sum_c \pi_c = 1$$

and

$$\sum_k \theta_{ck} = \sum_k P(k|Y = c) = 1$$

as well as those quantities should be nonnegative.

Our hammer: maximum likelihood estimation

Log-Likelihood of the training data

$$\mathcal{L} = \log P(\mathcal{D}) = \log \prod_{n=1}^{\mathsf{N}} \pi_{y_n} P(x_n|y_n) \tag{5}$$

$$= \log \prod_{n=1}^{\mathsf{N}} \left(\pi_{y_n} \prod_k \theta_{y_n k}^{z_{nk}} \right) \tag{6}$$

$$= \sum_n \left(\log \pi_{y_n} + \sum_k z_{nk} \log \theta_{y_n k} \right) \tag{7}$$

$$= \sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k} \tag{8}$$

Our hammer: maximum likelihood estimation

Log-Likelihood of the training data

$$\mathcal{L} = \log P(\mathcal{D}) = \log \prod_{n=1}^{\mathsf{N}} \pi_{y_n} P(x_n | y_n) \tag{5}$$

$$= \log \prod_{n=1}^{\mathsf{N}} \left(\pi_{y_n} \prod_k \theta_{y_n k}^{z_{nk}} \right) \tag{6}$$

$$= \sum_n \left(\log \pi_{y_n} + \sum_k z_{nk} \log \theta_{y_n k} \right) \tag{7}$$

$$= \sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k} \tag{8}$$

Optimize it!

$$(\pi_c^*, \theta_{ck}^*) = \arg\max \sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k}$$

Details

Note the separation of parameters in the likelihood

$$\sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k}$$

which implies that $\{\pi_c\}$ and $\{\theta_{ck}\}$ can be estimated separately.

Reorganize terms

$$\sum_n \log \pi_{y_n} = \sum_c \log \pi_c \times (\#\text{of data points labeled as c})$$

and

$$\sum_{n,k} z_{nk} \log \theta_{y_n k} = \sum_c \sum_{n:y_n=c} \sum_k z_{nk} \log \theta_{ck} = \sum_c \sum_{n:y_n=c,k} z_{nk} \log \theta_{ck}$$

The later implies $\{\theta_{ck}, k = 1, 2, \cdots, \mathsf{K}\}$ and $\{\theta_{c'k}, k = 1, 2, \cdots, \mathsf{K}\}$ can be estimated independently.

Estimating $\{\pi_c\}$

We want to maximize

$$\sum_c \log \pi_c \times (\text{\#of data points labeled as c})$$

Intuition

- Similar to roll a dice (or flip a coin): each side of the dice shows up with a probability of π_c (total C sides)
- And we have total N trials of rolling this dice

Solution

$$\pi_c^* = \frac{\text{\#of data points labeled as c}}{\text{N}}$$

Estimating $\{\theta_{ck}, k = 1, 2, \cdots, \mathsf{K}\}$

We want to maximize

$$\sum_{n:y_n=c,k} z_{nk} \log \theta_{ck}$$

Intuition

- Similar to roll a dice with color c: each side of the dice shows up with a probability of θ_{ck} (total K slides)
- And we have total $\sum_{n:y_n=c,k} z_{nk}$ trials.

Solution

$$\theta^*_{ck} = \frac{\text{\#of side-k shows up in data points labeled as c}}{\text{\#of all slides in data points labeled as c}}$$

Translating back to our problem of detecting spam emails

- Collect a lot of ham and spam emails as training examples
- Estimate the "bias"

$$p(\text{ham}) = \frac{\#\text{of ham emails}}{\#\text{of emails}}, \quad p(\text{spam}) = \frac{\#\text{of spam emails}}{\#\text{of emails}}$$

- Estimate the weights (i.e., $p(\text{dollar}|\text{ham})$ etc)

$$p(\text{funny\_word}|\text{ham}) = \frac{\#\text{of funny\_word in ham emails}}{\#\text{of words in ham emails}} \tag{9}$$

$$p(\text{funny\_word}|\text{spam}) = \frac{\#\text{of funny\_word in spam emails}}{\#\text{of words in spam emails}} \tag{10}$$

Classification rule

Given an unlabeled data point $x = \{z_k, k = 1, 2, \cdots, \mathsf{K}\}$**, label it with**

$$y^* = \arg\max_{c \in [\mathsf{C}]} P(y = c|x) \tag{11}$$

$$= \arg\max_{c \in [\mathsf{C}]} P(y = c) P(x|y = c) \tag{12}$$

$$= \arg\max_c [\log \pi_c + \sum_k z_k \log \theta_{ck}] \tag{13}$$

A short derivation of the maximum likelihood estimation

The steps are similar to the ones in Math Review

To maximize

$$\sum_{n:y_n=c} z_{nk} \log \theta_{ck}$$

We use the Lagrangian multiplier

$$\sum_{n:y_n=c,k} z_{nk} \log \theta_{ck} + \lambda \left(\sum_k \theta_{ck} - 1 \right)$$

Taking derivatives with respect to θ_{ck} and then find the stationary point

$$\sum_{n:y_n=c} \frac{z_{nk}}{\theta_{ck}} + \lambda = 0 \rightarrow \theta_{ck} = -\frac{1}{\lambda} \sum_{n:y_n=c} z_{nk}$$

Apply the constraint that $\sum_k \theta_{ck} = 1$,

$$\theta_{ck} = \frac{\sum_{n:y_n=c} z_{nk}}{\sum_{k'} \sum_{n:y_n=c} z_{nk'}}$$

Summary

You should know or be able to

- What naive Bayes model is
    - write down the joint distribution
    - explain the conditional independence assumption implied by the model
    - explain how this model can be used to distinguish spam from ham emails
- Be able to go through the short derivation for parameter estimation
    - The model illustrated here is called discrete Naive Bayes
    - Your homework asks you to apply the same principle to Gaussian naive Bayes
    - The derivation is very similar – except there you need to estimate Gaussian continuous random variables (instead of estimating discrete random variables like rolling a dice)
- think about another classification task that this model might be useful

To enhance your understanding

write a personalized spam email detector yourself

- Collect from your own email inbox, 500 samples of spam and good emails (the more, the merrier)
- Create a training (400 samples), validation (50 samples) and test dataset (50 samples)
- Estimate Naive Bayes model parameters for distinguishing ham and spam emails
- Apply the model to classify test dataset (you will use validation dataset later)
- Report your results on Discussion forum and post your questions of doing this experiment

This recipe is not 100% bullet-proof. You will discover practical issues. Working on those issues will improve your understanding of the algorithm and its practice.

Moving forward

Examine the classification rule for naive Bayes

$$y^* = \arg\max_c \log \pi_c + \sum_k z_k \log \theta_{ck}$$

For binary classification problem, this is just to determine the label basing on

$$\log \pi_1 + \sum_k z_k \log \theta_{1k} - \left(\log \pi_2 + \sum_k z_k \log \theta_{2k} \right)$$

This is just a linear function of the features $\{z_k\}$

$$w_0 + \sum_k z_k w_k$$

where we "absorb" $w_0 = \log \pi_1 - \log \pi_2$ and $w_k = \log \theta_{1k} - \log \theta_{2k}$.

Naive Bayes is a linear classifier

Fundamentally, what really matters in deciding decision boundary is

$$w_0 + \sum_k z_k w_k$$

This motivates many new methods. One of them is logistic regression, to be discussed in next lecture.