18-661 Introduction to Machine Learning

Naïve Bayes

Spring 2025

ECE – Carnegie Mellon University

Announcements

- HW 2 is due on February 21. Gradescope will ask you to mark your answers to each question in your submission; please make sure you do this (and leave enough time before the deadline to complete it).
- First mini-exam is on Feb 10th in class. Topics include everything until the lecture on Jan 29th.
- You are allowed to bring 1 one-sided handwritten US-letter-sized cheat sheet. No electronic devices are permitted. Calculators are allowed but will not be necessary.
- Remember that Piazza is always available for questions about the course, homework assignments, etc.

Outline

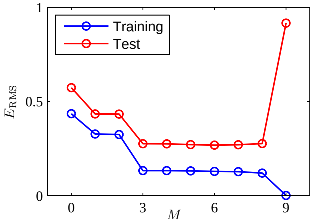
Review of Overfitting and Hyperparameter Tuning

Detecting Overfitting

Plot model complexity versus objective function:

- X axis: model complexity, e.g., M
- Y axis: error, e.g., RSS, RMS (square root of RSS), 0-1 loss

Compute the objective on a training (used to train the model) and test (used to evaluate the model) dataset.



As a model increases in complexity:

- Training error keeps reducing
- Test error may first reduce but eventually increase

Preventing Overfitting

- Try to use more training data (but collecting more data may not be feasible).
- Reduce the number of features (but it's not obvious which or how many features to remove).
- Regularize to encourage "simpler" models with smaller weights.
    - Choose the parameters to not just minimize risk, but avoid being large.

      $$\text{Ridge regression: } \min \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \frac{1}{2}\lambda\|w\|_2^2$$

    - Need to choose the regularization parameter λ...

Tuning with a Validation Dataset

Training data are used to learn $f(\cdot)$ (our model).
N samples/instances: $\mathcal{D}^{\text{TRAIN}} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_\mathsf{N}, y_\mathsf{N})\}$

Test data are used to assess the prediction error.

- M samples/instances: $\mathcal{D}^{\text{TEST}} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_\mathsf{M}, y_\mathsf{M})\}$
- They are used for assessing how well $f(\cdot)$ will do in predicting an unseen $\boldsymbol{x} \notin \mathcal{D}^{\text{TRAIN}}$

Validation data are used to optimize hyperparameter(s).
L samples/instances: $\mathcal{D}^{\text{VAL}} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_\mathsf{L}, y_\mathsf{L})\}$

Training data, validation and test data should not overlap!

Recipe

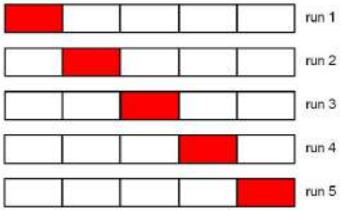
- For each possible value of the hyperparameter (say $\lambda = 1, 3, \cdots, 100$)
  - Train a model using $\mathcal{D}^{\text{TRAIN}}$
  - Evaluate the performance of the model on $\mathcal{D}^{\text{VAL}}$
- Choose the model with the best performance on $\mathcal{D}^{\text{VAL}}$
- Evaluate this model on $\mathcal{D}^{\text{TEST}}$ to get the final prediction error

Cross-validation

What if we do not have validation data?

- We split the training data into S equal parts.
- We use each part in turn as a validation dataset and use the others as a training dataset.
- We choose the hyperparameter such that the model performs the best over S trials (based on average, variance, etc.)
- Re-train with this hyperparameter on the entire training dataset.

Figure 1: S = 5: 5-fold cross validation



Special case: when S = N, this will be leave-one-out.

Review of the Bias-Variance Trade-off

Empirical Risk Minimization

So far, we have been doing empirical risk minimization (ERM)
For linear regression, $h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$, and we use squared loss ℓ.

$$R^{\text{EMP}}[h(\boldsymbol{x})] = \frac{1}{N} \sum_n \ell(h(\boldsymbol{x}_n), y_n)$$

$$R[h(\boldsymbol{x})] = \int_{\boldsymbol{x},y} \ell(h(\boldsymbol{x}), y) p(\boldsymbol{x}, y) d\boldsymbol{x} dy$$

- Limited Data: We don't know $p(\boldsymbol{x}, y)$, so we must hope that we have enough training data that the empirical risk approximates the real risk. Otherwise, we will overfit to the training data.
- Limited Function Class: The function $h(\boldsymbol{x})$ is restricted to a limited class (e.g. linear functions), which does not allow us to perfectly fit y, even if we had infinitely many training data points.

Bias-Variance Trade-off: Intuition

- High Bias: Model is not rich enough to fit the training dataset and achieve low training loss
- High Variance: If the training dataset changes slightly, the model changes a lot
- Regularization helps find a middle ground

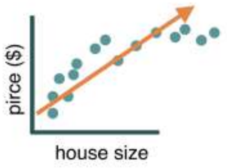


Figure 2: High Bias

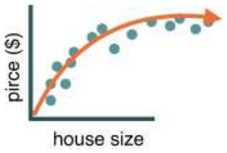


Figure 3: Just Right

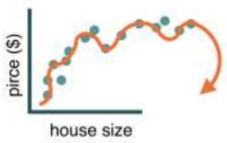


Figure 4: High Variance

As we regularize more (weight λ increases)...bias increases but variance decreases (hence, the trade-off)

Three Components of Average Risk

The average risk (with quadratic loss) can be decomposed as:

$$\mathbb{E}_{\mathcal{D}} R[h_{\mathcal{D}}(\boldsymbol{x})] = \underbrace{\int_{\mathcal{D}} \int_{\boldsymbol{x}} \int_{y} [h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x})]^2 p(\boldsymbol{x}, y) d\boldsymbol{x} dy \; P(\mathcal{D}) d\mathcal{D}}_{\text{VARIANCE: error due to training dataset}}$$

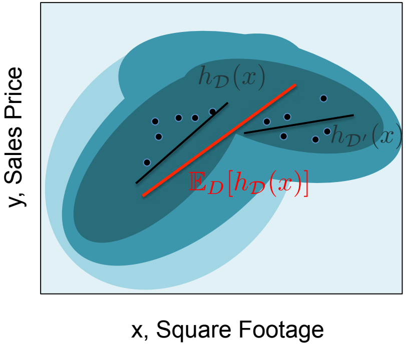
$$+ \underbrace{\int_{\boldsymbol{x}} \int_{y} [\mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{y}[y|\boldsymbol{x}]]^2 p(\boldsymbol{x}, y) d\boldsymbol{x} dy}_{\text{BIAS}^2\text{: error due to the model approximation}}$$

$$+ \underbrace{\int_{\boldsymbol{x}} \int_{y} [\mathbb{E}_{y}[y|\boldsymbol{x}] - y]^2 p(\boldsymbol{x}, y) d\boldsymbol{x} dy}_{\text{NOISE: error due to randomness of } y}$$

Here we define: $h_{\mathcal{D}}(\boldsymbol{x})$ as the output of the model trained on $\mathcal{D}$, $\mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x})$ as the expectation of the model over all datasets $\mathcal{D}$, and $\mathbb{E}_{y}[y|\boldsymbol{x}]$ as the expected value of y conditioned on $\boldsymbol{x}$.

Bias-Variance Trade-off: Illustration

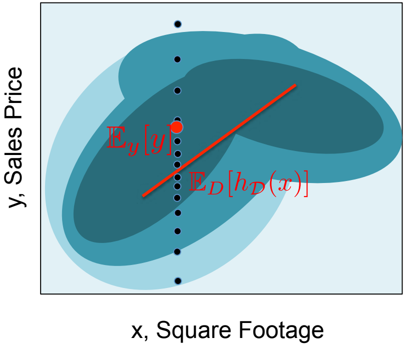
- Joint distribution of square footage x and house sales price y
- Each training dataset $\mathcal{D}$ yields a new linear model $h_{\mathcal{D}}(\boldsymbol{x})$
- Average of such models over infinitely many datasets sampled from the joint distribution is $\mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x})$



Bias-Variance Trade-off: Illustration

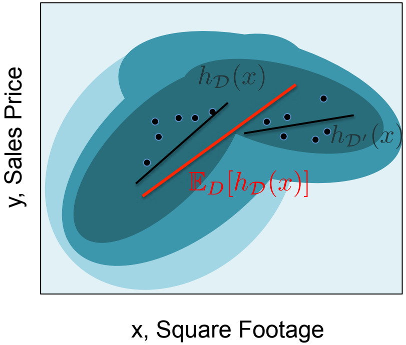
- If our model class was rich enough (eg. a high-degree polynomial), then $\mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x})$ should perfectly match $\mathbb{E}_y[y|\boldsymbol{x}]$
- Restricting to simpler models (eg. linear) results in a bias

$$\underbrace{\int_{\boldsymbol{x}}\int_y [\mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_y[y|\boldsymbol{x}]]^2 p(\boldsymbol{x}, y) d\boldsymbol{x} dy}_{\text{BIAS}^2\text{: error due to the model approximation}}$$



Bias-Variance Trade-off: Illustration

- Average of such models over infinitely many datasets sampled from the joint distribution is $\mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x})$
- Variance term captures how much individual models differ from the average $\underbrace{\int_{\mathcal{D}} \int_{\boldsymbol{x}} \int_{y} [h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}} h_{\mathcal{D}}(\boldsymbol{x})]^2 p(\boldsymbol{x}, y) d\boldsymbol{x} dy \; P(\mathcal{D}) d\mathcal{D}}_{\text{VARIANCE: error due to training dataset}}$

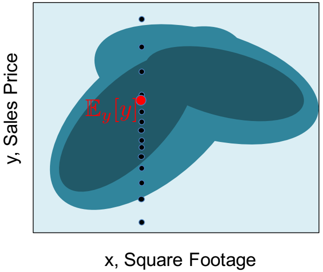


Bias-Variance Trade-off: Illustration

- For a given $\boldsymbol{x}$, we have a conditional distribution $p(y|\boldsymbol{x})$: the Bayesian optimal prediction of the label value for a given $\boldsymbol{x}$ is $\mathbb{E}_y[y|\boldsymbol{x}]$;
- The noise term measures the inherent variance in labels y

$$\underbrace{\int_{\boldsymbol{x}} \int_y [\mathbb{E}_y[y|\boldsymbol{x}] - y]^2 p(\boldsymbol{x}, y) d\boldsymbol{x} dy}_{\text{NOISE: error due to randomness of } y}$$

- This term has nothing to do with our prediction $h_{\mathcal{D}}(\boldsymbol{x})$



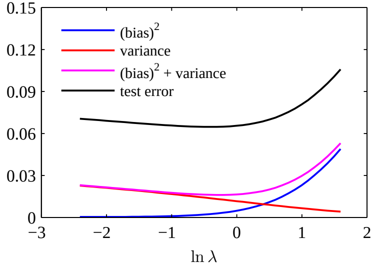
Bias-Variance Tradeoff

Error decomposes into 3 terms

$$\mathbb{E}_{\mathcal{D}} R[h_{\mathcal{D}}(\boldsymbol{x})] = \text{VARIANCE} + \text{BIAS}^2 + \text{NOISE}$$

where the first and the second term are inherently in conflict in terms of choosing what kind of $h(\boldsymbol{x})$ we should use (unless we have an infinite amount of data).

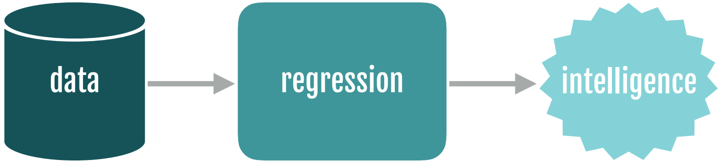
If we can compute all terms analytically, they will look like this

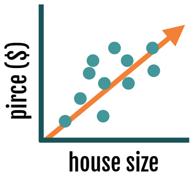


Classification using Naive Bayes

Task 1: Regression (so far we studied this)

How much should you sell your house for?

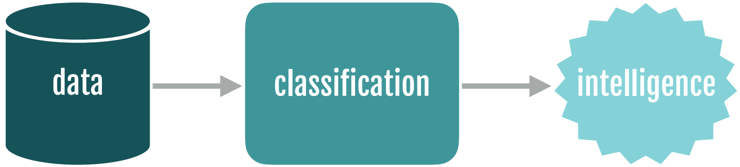


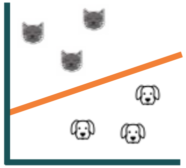


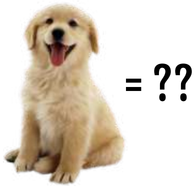
input: houses & features **learn**: $x \to y$ relationship **predict**: y (*continuous*)

Task 2: Classification (next topic)

Cat or dog?

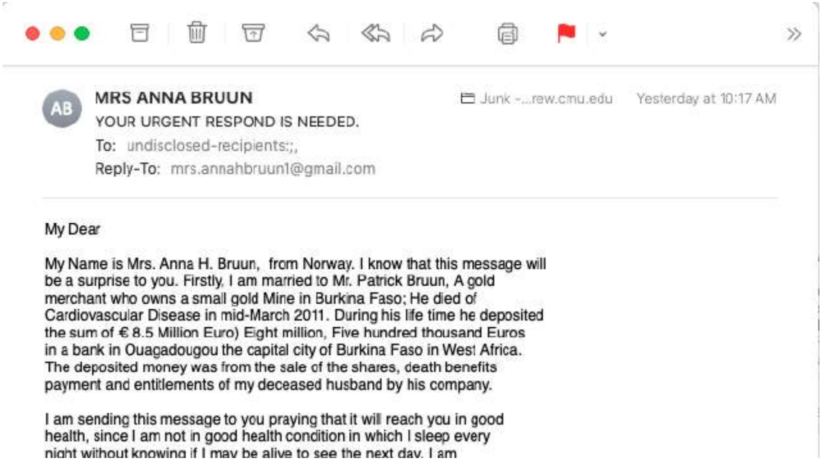






input: cats and dogs **learn**: $x \to y$ relationship **predict**: y (*categorical*)

Spam Classification: A Daily Battle


MRS ANNA BRUUN
YOUR URGENT RESPOND IS NEEDED.
To: undisclosed-recipients:;,
Reply-To: mrs.annahbruun1@gmail.com

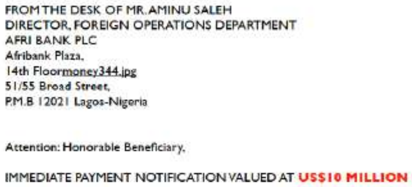
Junk -...rew.cmu.edu
Yesterday at 10:17 AM

My Dear

My Name is Mrs. Anna H. Bruun, from Norway. I know that this message will be a surprise to you. Firstly, I am married to Mr. Patrick Bruun, A gold merchant who owns a small gold Mine in Burkina Faso; He died of Cardiovascular Disease in mid-March 2011. During his life time he deposited the sum of € 8.5 Million Euro) Eight million, Five hundred thousand Euros in a bank in Ouagadougou the capital city of Burkina Faso in West Africa. The deposited money was from the sale of the shares, death benefits payment and entitlements of my deceased husband by his company.

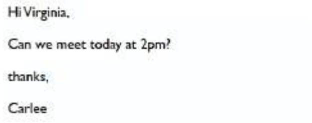
I am sending this message to you praying that it will reach you in good health, since I am not in good health condition in which I sleep every night without knowing if I may be alive to see the next day. I am

How to Tell Spam from Ham Emails?


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


Hi Virginia,

Can we meet today at 2pm?

thanks,

Carlee

How Might We Create Features?

Intuition

- Q: How might a human solve this problem?
- A: Simple strategy would be to look for keywords that we often associate with spam

Spam

- We expect to see words like "money", "free", "bank account"

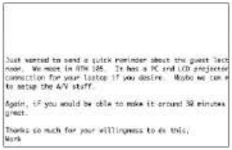
Ham

- Expect to see fewer spam words, more personalization (e.g., name)

Simple Strategy: Count the Words

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

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



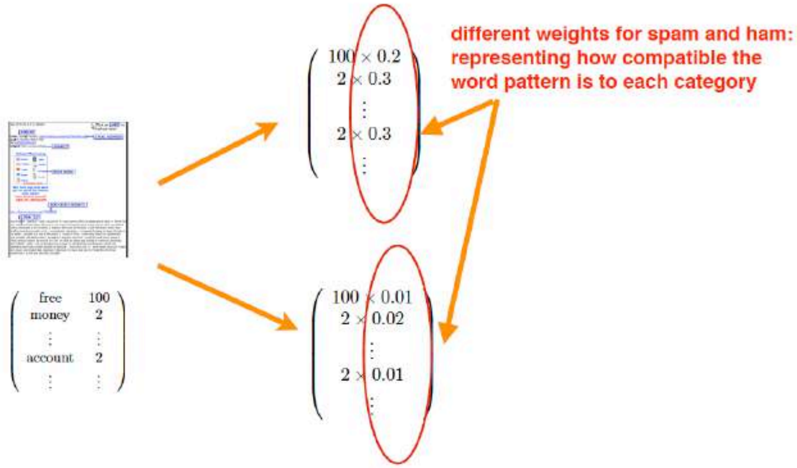
Just wanted to send a quick reminder about the guest lect
room. We meet in RTH 105. It has a PC and LCD projector
connection for your laptop if you desire. Maybe we can m
to setup the A/V stuff.

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


free 100
money 2
account 2
100 × 0.2
2 × 0.3
2 × 0.3
100 × 0.01
2 × 0.02
2 × 0.01
different weights for spam and ham: representing how compatible the word pattern is to each category

Weighted Sum of Those Telltale Words

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

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

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

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

= 3.2

= 1.03

Our Intuitive Model of Classification

1. **Assign weight to each word**
    - Let s:= spam weights, h:= ham weights
    - Compute compatibility score of **spam**
        - $(\#$ "free" $\times s_{\text{free}}) + (\#$ "account" $\times s_{\text{account}}) + (\#$ "money" $\times s_{\text{money}})$
    - Compute compatibility score of **ham**
        - $(\#$ "free" $\times h_{\text{free}}) + (\#$ "account" $\times h_{\text{account}}) + (\#$ "money" $\times h_{\text{money}})$
2. **Make a decision**
    - if spam score $>$ ham score then spam
    - else ham

Try It Out

Suppose you see the following email:

```
CONGRATULATIONS!! Your email address have won you the
lottery sum of US$2,500,000.00 USD to claim your prize,
contact your office agent (Athur walter) via email
claims2155@yahoo.com.hk or call +44 704 575 1113
```

And our weights for spam and ham are:
spam: [lottery=0.3, prize=0.3, office=0.01, email=0.01, . . .]
ham: [lottery=0.01, prize=0.01, office=0.1, email=0.05, . . .]

Will we predict that the email is spam or ham?

spam = 0.3*1 + 0.3*1 + 0.01*1 + 0.01*2 =0.63
ham = 0.01*1 + 0.01*1 + 0.1*1 + 0.05*2 = 0.22
so we predict spam!

How Do We Get the Weights?

Learn from experience

- get a lot of spams
- get a lot of hams

But what to optimize?

Naïve Bayes Model

Naïve Bayes Model for Identifying Spam

- **Class label**: binary
  - $y = \{$ spam, ham $\}$
- **Features**: word counts in the document (bag-of-words)
  - $\mathbf{x} = \{$(‘free’, 100), (‘lottery’, 5), (‘money’, 10)$\}$
  - Each pair is in the format of (word$_i$, #word$_i$), namely, a unique word in the dictionary, and the number of times it shows up
- **Model**

$$p(\mathbf{x}|spam) = p(\text{‘}free'|spam)^{100} p(\text{‘}lottery'|spam)^{5} p(\text{‘}money'|spam)^{10} \cdots$$

  - Choose the “most likely” option: $p(\mathbf{x}|spam)p(spam)$ vs. $p(\mathbf{x}|ham)p(ham)$

These conditional probabilities are the parameters we need to estimate

Why Is This Naïve?

- Strong assumption of conditional independence:

$$p(\text{word}_i, \text{word}_j | y) = p(\text{word}_i | y) p(\text{word}_j | y)$$

- Previous example:

$$p(\mathbf{x}|spam) = p(`free'|spam)^{100} p(`lottery'|spam)^{5} p(`money'|spam)^{10} \cdots$$

- Independence across different words as well as multiple occurrences of the same word
- This assumption makes estimation much easier (as we'll see)

Naïve Bayes Classification Rule

For any document $\mathbf{x}$, we want to compare $p(\text{spam}|\mathbf{x})$ and $p(\text{ham}|\mathbf{x})$

Recall that by Bayes rule we have:

$$p(\text{spam}|\mathbf{x}) = \frac{p(\mathbf{x}|\text{spam})p(\text{spam})}{p(\mathbf{x})}$$

$$p(\text{ham}|\mathbf{x}) = \frac{p(\mathbf{x}|\text{ham})p(\text{ham})}{p(\mathbf{x})}$$

Denominators are the same, and easier to compute logarithms, so we compare the spam and ham probabilities:

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

Classifier in Linear Form

$$\begin{aligned}\log[p(\mathbf{x}|\text{spam})p(\text{spam})] &= \log\left[\prod_i p(\text{word}_i|\text{spam})^{\#\text{word}_i} p(\text{spam})\right] \\ &= \sum_i (\#\text{word}_i) \log p(\text{word}_i|\text{spam}) + \log p(\text{spam})\end{aligned}$$

Similarly, we have

$$\log[p(\mathbf{x}|\text{ham})p(\text{ham})] = \sum_i (\#\text{word}_i) \log p(\text{word}_i|\text{ham}) + \log p(\text{ham})$$

We're back to the idea of comparing weighted sums of word occurrences!

$\log p(spam)$ *and* $\log p(ham)$ *are called "priors" (in our initial example we did not include them but they are important!)*

Parameter Estimation

Formal Definition of Naïve Bayes

Given a random vector $\mathbf{X} \in \mathbb{R}^{\mathrm{K}}$ and a dependent variable $Y \in [\mathrm{C}]$, the Naïve Bayes model defines the joint distribution

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}, Y = c) &= P(Y = c)P(\mathbf{X} = \mathbf{x} | Y = c) \\ &= P(Y = c) \prod_{k=1}^{\mathrm{K}} P(\text{word}_k | Y = c)^{x_k} \\ &= \pi_c \prod_{k=1}^{\mathrm{K}} \theta_{ck}^{x_k} \end{aligned}$$

where $\pi_c = P(Y = c)$ is the prior probability of class c, x_k is the number of occurences of the kth word, and $\theta_{ck} = P(\text{word}_k | Y = c)$ is the weight of the kth word for the cth class.

Learning Problem?

Training data

$$\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{\mathsf{N}} \to \mathcal{D} = \{(\{x_{nk}\}_{k=1}^{\mathsf{K}}, y_n)\}_{n=1}^{\mathsf{N}}$$

Goal

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

$$\sum_c \pi_c = 1$$

and

$$\sum_k \theta_{ck} = \sum_k P(\text{word}_k | Y = c) = 1$$

as well as: all π_c, $\theta_{ck} \geq 0$.

Our Hammer: Maximum Likelihood Estimation

Likelihood of the training data

$$\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N} \rightarrow \mathcal{D} = \{(\{x_{nk}\}_{k=1}^{K}, y_n)\}_{n=1}^{N}$$

$$L = P(\mathcal{D}) = \prod_{n=1}^{N} \pi_{y_n} P(\mathbf{x}_n | y_n)$$

Log-Likelihood of the training data

$$\mathcal{L} = \log P(\mathcal{D}) = \log \prod_{n=1}^{N} \pi_{y_n} P(\mathbf{x}_n | y_n)$$

Our Hammer: Maximum Likelihood Estimation

Log-Likelihood of the training data

$$\begin{aligned}
\mathcal{L} = \log P(\mathcal{D}) &= \log \prod_{n=1}^{\mathrm{N}} \pi_{y_n} P(\mathbf{x}_n | y_n) \\
&= \log \prod_{n=1}^{\mathrm{N}} \left(\pi_{y_n} \prod_k \theta_{y_n k}^{x_{nk}} \right) \\
&= \sum_n \left(\log \pi_{y_n} + \sum_k x_{nk} \log \theta_{y_n k} \right) \\
&= \sum_n \log \pi_{y_n} + \sum_{n,k} x_{nk} \log \theta_{y_n k}
\end{aligned}$$

Optimize it!

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

Separating the Optimization Variables

Note the separation of parameters in the likelihood

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

this 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

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

The latter implies $\{\theta_{ck}\}$ and $\{\theta_{c'k}\}$ for $c \neq c'$ 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 $\{\theta_{ck}, k = 1, 2, \cdots, \mathrm{K}\}$

We want to maximize

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

Intuition

- Again similar to rolling a dice: each side of the dice shows up with a probability of θ_{ck} (total K sides)
- And we have total $\sum_{n:y_n=c,k} x_{nk}$ trials (times a word shows up in class c).

Solution

$$\theta^*_{ck} = \frac{\#\text{of times word } k \text{ shows up in data points labeled as } c}{\#\text{total trials for data points labeled as } c}$$

Back to Detecting Spam Emails

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

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

- Estimate the weights, e.g., $p(\text{funny\_word}|\text{ham})$

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

Example: Spam Classification

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	1	2	3	Ham
Email 4	1	2	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\theta_{spam,free} = Pr(free|spam)$$
$$\theta_{spam,bank} = Pr(bank|spam)$$
$$\theta_{spam,meet} = Pr(meet|spam)$$
$$\theta_{spam,time} = Pr(time|spam)$$
$$\pi_{spam} = Pr(spam)$$

Example: Spam Classification

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	1	2	3	Ham
Email 4	1	2	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\begin{aligned}
\theta_{spam,free} &= Pr(free|spam) = (5+4)/(5+3+1+1+4+2+1+1) = 9/18 \\
\theta_{spam,bank} &= Pr(bank|spam) = (3+2)/18 = 5/18 \\
\theta_{spam,meet} &= Pr(meet|spam) = 2/18 \\
\theta_{spam,time} &= Pr(time|spam) = 2/18 \\
\pi_{spam} &= Pr(spam) = 2/4
\end{aligned}$$

Example: Spam Classification

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	1	2	3	Ham
Email 4	1	2	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\theta_{ham,free} = Pr(free|ham)$$
$$\theta_{ham,bank} = Pr(bank|ham)$$
$$\theta_{ham,meet} = Pr(meet|ham)$$
$$\theta_{ham,time} = Pr(time|ham)$$
$$\pi_{ham} = Pr(ham)$$

Example: Spam Classification

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	1	2	3	Ham
Email 4	1	2	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\theta_{ham,free} = Pr(free|ham) = 3/16$$
$$\theta_{ham,bank} = Pr(bank|ham) = 3/16$$
$$\theta_{ham,meet} = Pr(meet|ham) = 5/16$$
$$\theta_{ham,time} = Pr(time|ham) = 5/16$$
$$\pi_{ham} = Pr(ham) = 2/4$$

Classification Rule

Given an unlabeled point $\mathbf{x} = \{x_k, k = 1, 2, \cdots, \mathrm{K}\}$, how to label it?

$$\begin{aligned} y^* &= \arg\max_{c \in [\mathrm{C}]} P(y = c|\mathbf{x}) \\ &= \arg\max_{c \in [\mathrm{C}]} P(y = c)P(\mathbf{x}|y = c) \\ &= \arg\max_c [\log \pi_c + \sum_k x_k \log \theta_{ck}] \end{aligned}$$

Choose class c that maximizes the log-likelihood of an observed email

Example: Spam Classification

$$\begin{aligned} \theta_{spam,free} &= Pr(free|spam) = 9/18 & \theta_{ham,free} &= Pr(free|ham) = 3/16 \\ \theta_{spam,bank} &= Pr(bank|spam) = 5/18 & \theta_{ham,bank} &= Pr(bank|ham) = 3/16 \\ \theta_{spam,meet} &= Pr(meet|spam) = 2/18 & \theta_{ham,meet} &= Pr(meet|ham) = 5/16 \\ \theta_{spam,time} &= Pr(time|spam) = 2/18 & \theta_{ham,time} &= Pr(time|ham) = 5/16 \\ \pi_{spam} &= Pr(spam) = 2/4 & \pi_{ham} &= Pr(ham) = 2/4 \end{aligned}$$

We observe a new email with the word counts (free, bank, meet, time) = (1,3,4,2). Should it be classified as spam or ham?

$$\begin{aligned} \log \Pr(spam|\mathbf{x}) &\propto \log\left(\Pr(spam) \cdot \Pr(\mathbf{x}|spam)\right) \\ &= \log\left(\frac{2}{4} \cdot \left(\frac{9}{18}\right)\left(\frac{5}{18}\right)^3\left(\frac{2}{18}\right)^4\left(\frac{2}{18}\right)^2\right) \\ &= -7.99 \end{aligned}$$

Example: Spam Classification

$$\begin{aligned} \theta_{spam,free} &= Pr(free|spam) = 9/18 & \theta_{ham,free} &= Pr(free|ham) = 3/16 \\ \theta_{spam,bank} &= Pr(bank|spam) = 5/18 & \theta_{ham,bank} &= Pr(bank|ham) = 3/16 \\ \theta_{spam,meet} &= Pr(meet|spam) = 2/18 & \theta_{ham,meet} &= Pr(meet|ham) = 5/16 \\ \theta_{spam,time} &= Pr(time|spam) = 2/18 & \theta_{ham,time} &= Pr(time|ham) = 5/16 \\ \pi_{spam} &= Pr(spam) = 2/4 & \pi_{ham} &= Pr(ham) = 2/4 \end{aligned}$$

We observe a new email with the word counts (free, bank, meet, time) = (1,3,4,2). Should it be classified as spam or ham?

$$\begin{aligned} \log \Pr(ham|\mathbf{x}) &\propto \log \Pr(ham) \cdot \Pr(\mathbf{x}|ham) \\ &= \log \left(\frac{2}{4} \cdot \left(\frac{3}{16}\right) \left(\frac{3}{16}\right)^3 \left(\frac{5}{16}\right)^4 \left(\frac{5}{16}\right)^2 \right) \\ &= -6.2399 \end{aligned}$$

Example: Spam Classification

We observe a new email with the word counts (free, bank, meet, time) = (1,3,4,2). Should it be classified as spam or ham?

$$\begin{aligned}
\log \Pr(\textit{spam}|\mathbf{x}) &\propto \log \Pr(\textit{spam}) \cdot \Pr(\mathbf{x}|\textit{spam}) \\
&= \log \left(\frac{2}{4} \cdot \left(\frac{9}{18}\right) \left(\frac{5}{18}\right)^3 \left(\frac{2}{18}\right)^4 \left(\frac{2}{18}\right)^2 \right) \\
&= -7.99
\end{aligned}$$

$$\begin{aligned}
\log \Pr(\textit{ham}|\mathbf{x}) &\propto \log \Pr(\textit{ham}) \cdot \Pr(\mathbf{x}|\textit{ham}) \\
&= \log \left(\frac{2}{4} \cdot \left(\frac{3}{16}\right) \left(\frac{3}{16}\right)^3 \left(\frac{5}{16}\right)^4 \left(\frac{5}{16}\right)^2 \right) \\
&= -6.2399
\end{aligned}$$

ANSWER: Ham

Missing Features: Some Words Never Occur in Ham Emails

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

In this training phase, we can just use all available values and ignore missing values

$$\begin{aligned}
\theta_{spam,free} &= Pr(free|spam) = 9/18 & \theta_{ham,free} &= Pr(free|ham) = 3/13 \\
\theta_{spam,bank} &= Pr(bank|spam) = 5/18 & \theta_{ham,bank} &= Pr(bank|ham) = 0/13 \\
\theta_{spam,meet} &= Pr(meet|spam) = 2/18 & \theta_{ham,meet} &= Pr(meet|ham) = 5/13 \\
\theta_{spam,time} &= Pr(time|spam) = 2/18 & \theta_{ham,time} &= Pr(time|ham) = 5/13 \\
\pi_{spam} &= Pr(spam) = 2/4 & \pi_{ham} &= Pr(ham) = 2/4
\end{aligned}$$

Missing Features: Test Phase

$$\begin{aligned}
\theta_{spam,free} &= Pr(free|spam) = 9/18 & \theta_{ham,free} &= Pr(free|ham) = 3/13 \\
\theta_{spam,bank} &= Pr(bank|spam) = 5/18 & \theta_{ham,bank} &= Pr(bank|ham) = 0/13 \\
\theta_{spam,meet} &= Pr(meet|spam) = 2/18 & \theta_{ham,meet} &= Pr(meet|ham) = 5/13 \\
\theta_{spam,time} &= Pr(time|spam) = 2/18 & \theta_{ham,time} &= Pr(time|ham) = 5/13 \\
\pi_{spam} &= Pr(spam) = 2/4 & \pi_{ham} &= Pr(ham) = 2/4
\end{aligned}$$

New email with the word counts (free, bank, meet, time) = (1,3,4,2),

$$\begin{aligned}
\log \Pr(ham|\mathbf{x}) &\propto \log \Pr(ham) \cdot \Pr(\mathbf{x}|ham) \\
&= \log\left(\frac{2}{4} \cdot \left(\frac{3}{13}\right)\left(\frac{0}{13}\right)^3\left(\frac{5}{13}\right)^4\left(\frac{5}{13}\right)^2\right) \\
&= -\infty
\end{aligned}$$

Problem: The email is ALWAYS classified as spam. Just because an event has not happened yet, doesn't mean that it won't ever happen.

Dealing with Missing Features: Solution 1

Remove the features that take zero values for one or more classes

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

Remove the 'bank' column

We can then use the same procedure as before to learn the parameters, and then use these parameters to classify new emails

But then we are wasting a lot of useful data..

Dealing with Missing Features: Solution 2

Use Laplacian smoothing: Pretend you've seen each word 1 extra time for each class (spam and ham). This is called a 'pseudo-count'.

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

$$\begin{aligned}
\theta_{ham,free} &= Pr(free|ham) = (3+1)/(13+4) \\
\theta_{ham,bank} &= Pr(bank|ham) = (0+1)/(13+4) \\
\theta_{ham,meet} &= Pr(meet|ham) = (5+1)/(13+4) \\
\theta_{ham,time} &= Pr(time|ham) = (5+1)/(13+4) \\
\pi_{ham} &= Pr(ham) = 2/4
\end{aligned}$$

Dealing with Missing Features: Solution 2

More generally, pretend you've seen each word $\alpha \geq 1$ extra times.

	free	bank	meet	time	y
Email 1	5	3	1	1	Spam
Email 2	4	2	1	1	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

$$p(\text{funny\_word}|\text{spam}) = \frac{\#\text{of funny\_word in spam emails} + \alpha}{\#\text{of words in spam emails} + \alpha \times \#\text{of unique words}}$$

Laplacian Smoothing: History and Effect on ML Estimate

History: What is the prob. that the sun will rise tomorrow?

- Given a large sample of days with the rising sun, we still can not be completely sure that the sun will still rise tomorrow

$$\Pr(\text{sun rising tomorrow}|\text{it rose } t \text{ times}) = \frac{t+1}{t+2}$$

Effect on the ML estimate

- Laplace smoothing biases the ML estimate
- Equivalent to performing MAP estimation with a Dirichlet (multi-variate Beta) prior
- As training data size grows, the effect of Laplacian smoothing disappears

Example: Perform Laplacian Smoothing

	free	bank	meet	time	y
Email 1	5	3	1	0	Spam
Email 2	4	2	1	0	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\begin{aligned}
\theta_{spam,free} &= Pr(free|spam) \\
\theta_{spam,bank} &= Pr(bank|spam) \\
\theta_{spam,meet} &= Pr(meet|spam) \\
\theta_{spam,time} &= Pr(time|spam) \\
\pi_{spam} &= Pr(spam)
\end{aligned}$$

Example: Perform Laplacian Smoothing

	free	bank	meet	time	y
Email 1	5	3	1	0	Spam
Email 2	4	2	1	0	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\begin{aligned}
\theta_{spam,free} &= Pr(free|spam) = (9+1)/(16+4) = 10/20 \\
\theta_{spam,bank} &= Pr(bank|spam) = (5+1)/(16+4) = 6/20 \\
\theta_{spam,meet} &= Pr(meet|spam) = (2+1)/(16+4) = 3/20 \\
\theta_{spam,time} &= Pr(time|spam) = (0+1)/(16+4) = 1/20 \\
\pi_{spam} &= Pr(spam) = 2/4
\end{aligned}$$

Example: Perform Laplacian Smoothing

	free	bank	meet	time	y
Email 1	5	3	1	0	Spam
Email 2	4	2	1	0	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\theta_{ham,free} = Pr(free|ham)$$
$$\theta_{ham,bank} = Pr(bank|ham)$$
$$\theta_{ham,meet} = Pr(meet|ham)$$
$$\theta_{ham,time} = Pr(time|ham)$$
$$\pi_{ham} = Pr(ham)$$

Example: Perform Laplacian Smoothing

	free	bank	meet	time	y
Email 1	5	3	1	0	Spam
Email 2	4	2	1	0	Spam
Email 3	2	0	2	3	Ham
Email 4	1	0	3	2	Ham

Find ML estimates of parameters π_c and θ_{ck}

$$\begin{aligned}
\theta_{ham,free} &= Pr(free|ham) = (3+1)/(13+4) = 4/17 \\
\theta_{ham,bank} &= Pr(bank|ham) = (0+1)/(13+4) = 1/17 \\
\theta_{ham,meet} &= Pr(meet|ham) = (5+1)/(13+4) = 6/17 \\
\theta_{ham,time} &= Pr(time|ham) = (5+1)/(13+4) = 6/17 \\
\pi_{ham} &= Pr(ham) = 2/4
\end{aligned}$$

Summary

You should know

- what a Naïve Bayes model is
- why it is 'naïve'
- how this model can be used to classify spam vs ham emails
- Handle missing features via Laplace smoothing

Intuition: Logistic Regression

Examine the classification rule for naive Bayes

$$y^* = \arg\max_c \left(\log \pi_c + \sum_k x_k \log \theta_{ck} \right)$$

For binary classification, we thus determine the label based on the sign of

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

This is just a linear function of the features (word-counts) $\{x_k\}$

$$w_0 + \sum_k x_k w_k$$

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

Intuition: Logistic Regression

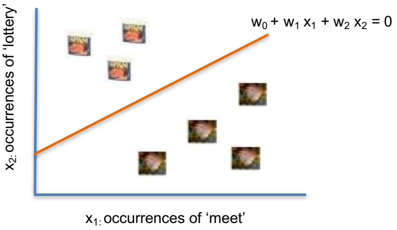
Fundamentally, what really matters is the decision boundary

$$w_0 + \sum_k x_k w_k$$

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

Intuition: Logistic Regression

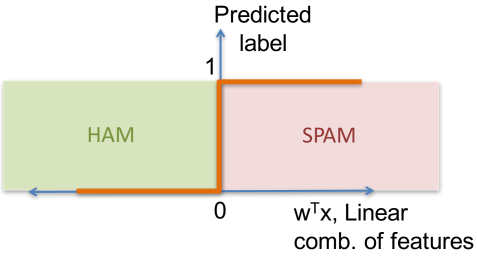
- x_1 = # of times 'meet' appears in an email
- x_2 = # of times 'lottery' appears in an email
- Define feature vector $\mathbf{x} = [1, x_1, x_2]$
- Learn the decision boundary $w_0 + w_1 x_1 + w_2 x_2 = 0$ such that
  - If $\mathbf{w}^\top \mathbf{x} \geq 0$ declare $y = 1$ (spam)
  - If $\mathbf{w}^\top \mathbf{x} < 0$ declare $y = 0$ (ham)


$w_0 + w_1 x_1 + w_2 x_2 = 0$
x_2: occurrences of 'lottery'
x_1: occurrences of 'meet'

Key Idea: If 'meet' appears few times and 'lottery' appears many times than the email is spam

Visualizing a Linear Classifier

- x_1 = # of times 'lottery' appears in an email
- x_2 = # of times 'meet' appears in an email
- Define feature vector $\mathbf{x} = [1, x_1, x_2]$
- Learn the decision boundary $w_0 + w_1 x_1 + w_2 x_2 = 0$ such that
  - If $\mathbf{w}^\top \mathbf{x} \geq 0$ declare $y = 1$ (spam)
  - If $\mathbf{w}^\top \mathbf{x} < 0$ declare $y = 0$ (ham)



$y = 1$ for spam, $y = 0$ for ham

Generative Model v.s. Discriminative Model

The set

$$\{x : P(Y = 1|X = x) = P(Y = 0|X = x)\}$$

is called the *decision boundary*.

In a **generative classifier**, we model the class-conditional densities $P(Y|X = x)$ explicitly. In other words,

$$P(Y = 1|X = x) = \frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x|Y = 1)P(Y = 1) + P(X = x|Y = 0)P(Y = 0)}$$

This means we need to do two separate density estimates, i.e., $P(X|Y)$ and $P(Y)$.

In a **discriminative classifier**, we avoid that and directly model the discriminant function, which is tantamount to modeling the decision boundary.