Generative Classifiers: Part 1

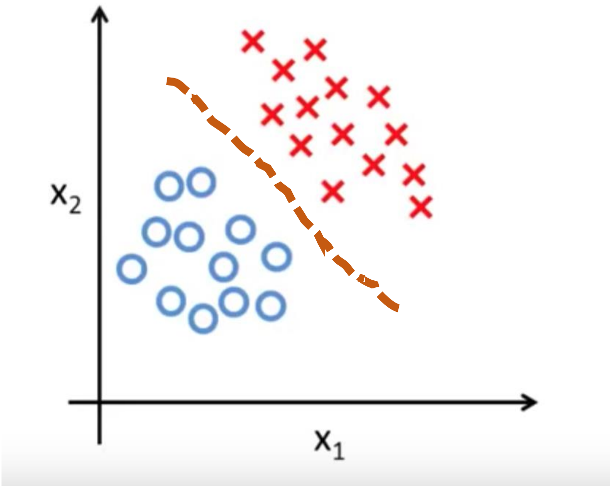
CSC411/2515: Machine Learning and Data Mining, Winter 2018

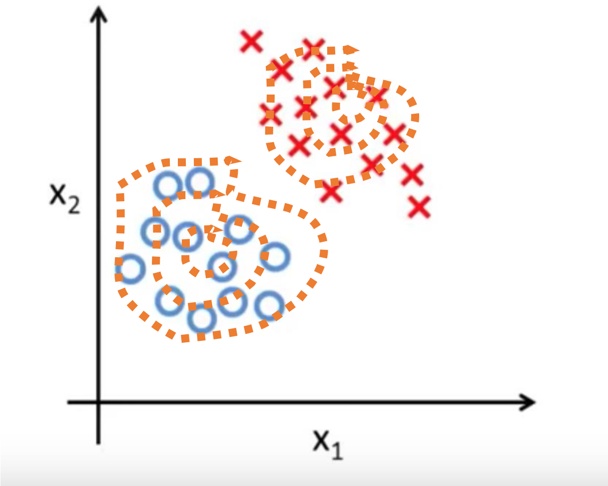
Michael Guerzhoy and Lisa Zhang

This Week

- Discriminative vs Generative Models
- Simple Model: Does the patient have cancer?
  - Discrete test
  - Continuous test
- Naïve Bayes: Spam filtering example
- Continuous features
- Naïve Bayes and Logistic Regression

Discriminative vs Generative Models





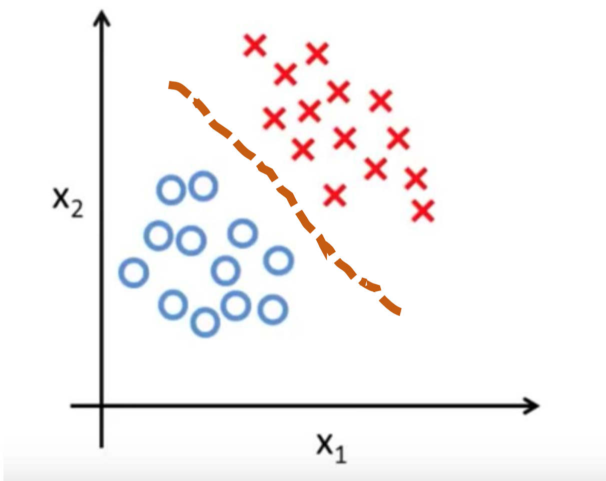
Build a classifier that learns a decision boundary

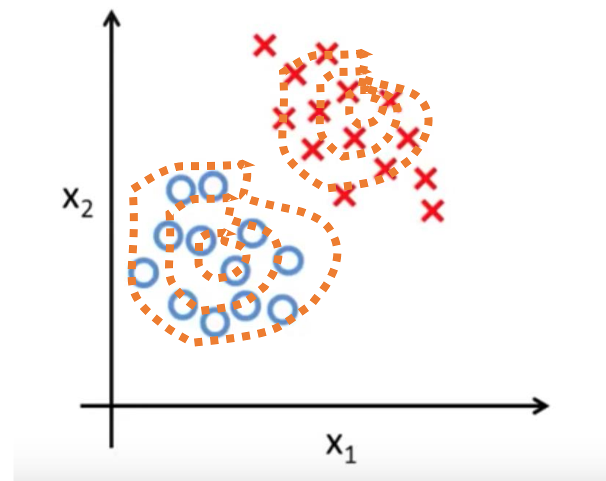
Classify test data by seeing which side of the boundary it falls into. (what are some examples we've seen?)

Build a model for what the data in each class looks like (i.e., the distribution of the data)

Classify test data by comparing it against the models for the data in the two classes.

Discriminative vs Generative Models





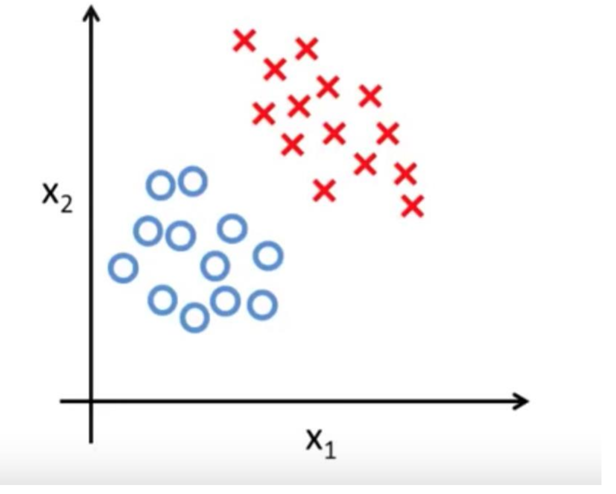
Build a classifier that learns a decision boundary

Models $p(y|x)$ directly

Build a model of what data in each class looks like

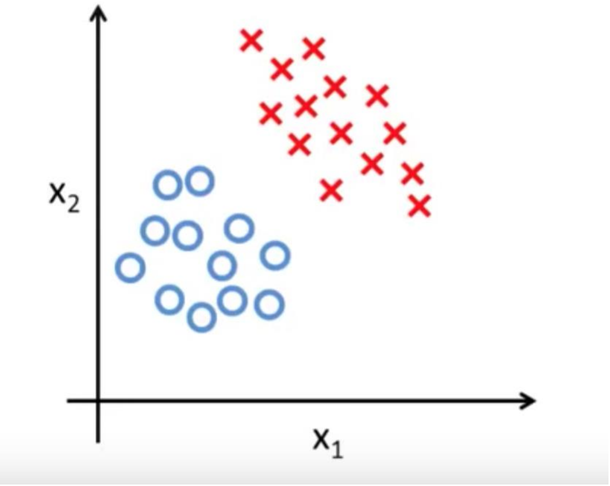
Models $p(x|y)$ for each value of y and $p(y)$ and then use Bayes' rule to find $p(y|x)$

Example of p(x|y)



- $P(x_1, x_2|y = blue) = N(x_1|\mu = 1, \sigma^2 = 1)\ N(x_2|\mu = 1, \sigma^2 = 1)$
  - Product of normal densities for both x_1 and x_2
  - Highest density at (1, 1), lower densities far away from (1,1)
- Wouldn't work for the red crosses: the probability distribution doesn't seem symmetric

Example of p(y)



- $P(y = blue) = \frac{12}{12+14}, P(y = red) = \frac{14}{12+14}$

Does the patient have cancer?

- We perform a lab test and the result comes back positive
- The test comes back positive in 98% of cases where the cancer is present
- The test comes back negative in 97% of cases where there is no cancer
- 0.008 of the population has cancer
  - And the cancer screening was random
  - (Why is this important?)

Generative process for data

- $P(cancer) = 0.008$
- $P(+|cancer) = .98$
- $P(-|\neg cancer) = .97$

θ

- θ determines how the test results are generated
  - Person i has cancer with prob. 0.008
  - The probability of a positive test for person i depends on whether they have cancer or not

- $P(cancer) = 0.008$
- $P(+|cancer) = .98$
- $P(-|\neg cancer) = .97$

- $P(cancer|+) = \frac{P(+|cancer)P(cancer)}{P(+)}$

$$= \frac{P(+|cancer)P(cancer)}{P(+|cancer)P(cancer) + P(+|\neg cancer)P(\neg cancer)}$$

Learning a Generative Model

- For the cancer data, just count the number of points in the training set (of size N) belonging to each category
- $P(cancer) \approx \frac{count(cancer,)}{N}$
- $P(+|cancer) \approx \frac{count(cancer,+)}{count(cancer)}$
- (Could get $P(cancer|+)$ by counting as well)

Gaussian Classifiers

- Suppose the test actually outputs a real number *t*
  - $P(cancer) \approx \frac{count(cancer)}{N}$
  - $P(t|cancer) = N(t|\mu_{cancer}, \sigma^2_{cancer})$
  - $P(t|\neg cancer) = N(t|\mu_{\neg cancer}, \sigma^2_{\neg cancer})$
    - $\theta = \{\mu_{cancer}, \mu_{\neg cancer}, \sigma_{cancer}, \sigma_{\neg cancer}, \dots\}$
  - θ determines how the test results are generated
    - Decide whether person *i* has cancer (with prob P(cancer))
    - Now generate the test output *t*

Learn using maximum likelihood

- What's the probability that the person has cancer, if we know θ?
  - $P_\theta(cancer|t) = \frac{P_\theta(t|cancer)P_\theta(cancer)}{P_\theta(t|cancer)P_\theta(cancer) + P_\theta(t|\neg cancer)P_\theta(\neg cancer)}$

Learning a Gaussian with Maximum Likelihood

- We have all the t's for patients *with* cancer
- Maximum Likelihood:
  - $argmax_{\mu_c, \sigma_c^2} \prod_i N(t^{(i)} | \mu_c, \sigma_c^2), i \in cancer$
  - Solution (show with calculus!):
    - $\widehat{\mu_c} = \overline{t^{(i)}}, i \in cancer$
    - $\widehat{\sigma_c^2} = \sum_i \frac{\left(t^{(i)} - \widehat{\mu_c}\right)^2}{\#ncancer}, i \in cancer$

Classification of new instances

- Suppose don't know θ
- What's $P(cancer|t, D)$?
  - *Not* $P_{\theta_{MAP}}(cancer|t, D)$*!*

Classification of new instances

- Suppose we are estimating θ from the data
- What's $P(cancer|t, D)$?
  - $\sum_{\theta' \in \Theta} P(cancer|\theta', t)P(\theta'|D) = \sum_{\theta' \in \Theta} P_{\theta'}(cancer|t)P(\theta'|D)$
  - Intuition: consider all the possible θ', compute the probability according to each of them, and weight them by how much we believe that the true θ could be θ'

- Suppose we are estimating θ from the data
- What's $P(cancer|t)$?
  - $P_{\theta_{MAP}}(cancer|t)$ is not a horrible estimate here

Multidimensional features

- In the previous example, x was scale (one-dimensional)
- In general, features x can be high-dimensional.
- Can do something similar:
  - $P(y = c|x_1, \ldots, x_p) = \frac{P(y=c)P\left(x_1, \ldots, x_p \middle| y = c\right)}{\sum_{c'} P(y=c')P\left(x_1, \ldots, x_p \middle| y = c'\right)}$
- But now, $P\left(x_1, \ldots, x_p \middle| y = c\right)$ is harder to model
  - More on this later

Naïve Bayes

- Assume the input features are conditionally independent given the class:
  $$P\left(x_1, \ldots, x_p \middle| y = c\right) = \Pi_{i=1}^{p} \, p(x_i | y = c)$$
  - Note: this is different from unconditional independence!

- Fairly strong assumption, but works quite well in practice
- A model that is not right can still be useful
  - "All models are wrong, but some are useful" – George P. Box

Naïve Bayes classifier

- $c = argmax_c P(y = c) \prod_i \; P(x_i | y = c)$
- If x_i are discrete, learn $P(\boldsymbol{x} | class)$ using

$$P(x_i = 1|c) \approx \frac{count(x_i = 1, c)}{count(c)}$$

  - I.e., count how many times the attribute appears in emails of class *c*

Spam Filtering Examples

Email	Spam?
send us your password	Y
send us your review	N
review your password	N
review us	Y
send your password	Y
send us your account	Y
review us now	????

- Observe attributes $x_1, x_2, \dots, x_n$ (e.g., keyword 1, 2, 3, … are present in the email, respectively)
- Goal: classify email as **spam** or **non-spam**

“Bag of words” features: features based on the appearance of keywords in the text, disregarding the order of the words

Naïve Bayes Spam Filter

Email	Spam?
send us your password	Y
send us your review	N
review your password	N
review us	Y
send your password	Y
send us your account	Y
review us now	????

	p(spam) = 4/6	p(notspam) = 2/6
	p(w|spam)	**p(w|notspam)**
password	2/4	1/2
review	1/4	2/2
send	3/4	1/2
us	3/4	1/2
your	3/4	1/2
account	1/4	0/2

Naïve Bayes Spam Filter

Email	Spam?
send us your password	Y
send us your review	N
review your password	N
review us	Y
send your password	Y
send us your account	Y
review us ~~now~~	????

	p(spam) = 4/6	p(notspam) = 2/6
	p(w|spam)	**p(w|notspam)**
password	2/4	1/2
review	1/4	2/2
send	3/4	1/2
us	3/4	1/2
your	3/4	1/2
account	1/4	0/2

p(review us | spam) = (1-2/4)(1/4)(1-3/4)(3/4)(1-3/4)(1-1/4)

p(review us | notspam) = (1-1/2)(2/2)(1-1/2)(1/2)(1-1/2)(1-0/2)

$$p(spam \mid review\ us) = \frac{p(review\ us \mid spam)p(spam)}{p(review\ us \mid spam)p(spam) + p(review\ us \mid notspam)p(notspam)}$$

$$p(not\ spam \mid review\ us) = \frac{p(review\ us \mid notspam)p(notspam)}{p(review\ us \mid spam)p(spam) + p(review\ us \mid notspam)p(notspam)}$$

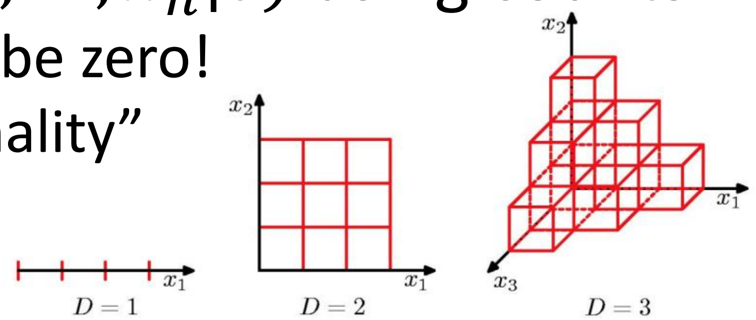
Naïve Bayes Classification

Prediction:

$$
\begin{aligned}
c_{MAP} &= argmax_c P(c|x_1, x_2, .., x_n) \\
&= argmax_c \frac{P(x_1, \ldots, x_n|c)P(c)}{P(x_1, \ldots, x_n)} \\
&= argmax_c P(x_1, \ldots, x_n|c)P(c)
\end{aligned}
$$

Naïve Bayes classifier: Why?

- Can't estimate $P(x_1, \dots, x_n|c)$ using counts
  - Most counts would be zero!
  - "curse of dimensionality"



- What if $count(x_i = 1, c)$ is 0?
  - We would never assign class *c* to examples with $x_i = 1$
  - So use $P(x_i = 1|c) \approx \frac{count(x_i=1,c)+m\,\hat{p}}{count(c)+m}$
  - m is a parameter
    - Interpretation: the number of "virtual" examples added to the training set

Prior estimate for $P(x_i = 1|c)$

Naïve Bayes classifier

- Pros:
  - Fast to train (single pass through data)
  - Fast to test
  - Less overfitting
  - Easy to add/remove classes
  - Handles partial data
- Cons:
  - When naïve i.i.d. assumption does not hold, can perform much worse

Naïve Bayes assumptions

- Conditional independence:
  - Appearances of “loan” and “diploma” need not be statistically independent in general
    - This is useful! If both are indicative of the email being spam, they would not be statistically independent, since both are likely to appear in a spam email – what we are counting on
  - If we *know* the email is a spam email, the fact that “dog” appears in it doesn’t make it more or less likely that “homework” will appear in
    - Does this sound like it would be true?
- Effect of conditional non-independence
  - “homework” and “assignment” would count as *two* pieces of evidence for classifying the email, even though the second keyword doesn’t add any information

Naïve Bayes vs Logistic Regression

Compute the log-odds of $y = c$ given the inputs using NB:

$$\log \frac{P(y = c | x_1, \dots, x_p)}{P(y = c' | x_1, \dots, x_p)} = \log \frac{P(y = c)\Pi_j P(x_j | c)}{P(y = c')\Pi_j P(x_j | c')}$$

$$= \log \frac{P(y = c)}{P(y = c')} + \sum_j \log \frac{P(x_j | y = c)}{P(x_j | y = c')}$$

We can write this as:

$$\log \frac{P(y = c | x_1, \dots, x_p)}{P(y = c' | x_1, \dots, x_p)} = \beta_0 + \sum_j \beta_j x_j$$

With $\beta_0 = \log \frac{P(y=c)}{P(y=c')} + \sum_j \log \frac{P(x_j = 0 | y = c)}{P(x_j = 0 | y = c')}$

And $\beta_j = \log \frac{P(x_j = 1 | y = c)}{P(x_j = 1 | y = c')} - \log \frac{P(x_j = 0 | y = c)}{P(x_j = 0 | y = c')}$

Naïve Bayes vs Logistic Regression

- Naïve Bayes model:

$$\log \frac{P(y = c | x_1, \dots, x_p)}{P(y = c' | x_1, \dots, x_p)} = \beta_0 + \sum_j \beta_j x_j$$

for specific $\beta_j's$.

- Same form as the Logistic Regression model, but in LR the β_j are whatever maximizes the likelihood
- We assume $x_j \in \{0, 1\}$

- What if x_j are continuous?

Naïve Bayes vs Logistic Regression

Assume $x_j \sim N\left(\mu_{c,j}, \sigma_j^2\right)$

$$\log \frac{P(x_j \mid y = c)}{P(x_j \mid y = c')} = \log \frac{(2\pi\sigma_j^2)^{-1/2} \exp(-(x_j - \mu_{c,j})^2 / 2\sigma_j^2)}{(2\pi\sigma_j^2)^{-1/2} \exp(-(x_j - \mu_{c',j})^2 / 2\sigma_j^2)}$$

$$= \log \frac{\exp(-(x_j^2 - 2x_j\mu_{c,j} + \mu_{c,j}^2)/2\sigma_j^2)}{\exp(-(x_j^2 - 2x_j\mu_{c',j} + \mu_{c',j}^2)/2\sigma_j^2)}$$

$$= \log \frac{\exp(x_j \, \mu_{c,j} \, / \, \sigma_j^2)}{\exp(x_j \, \mu_{c',j} \, / \, \sigma_j^2)} + \log \frac{\exp(-\mu_{c,j}^2 \, / \, 2\sigma_j^2)}{\exp(-\mu_{c',j}^2 \, / \, 2\sigma_j^2)}$$

Define $\beta_j = (\mu_{c,j} - \mu_{c',j}) \, / \sigma_j^2$ to again get:

$$\log \frac{P\left(y = c \middle| x_1, \ldots, x_p\right)}{P\left(y = c' \middle| x_1, \ldots, x_p\right)} = \beta_0 + \sum_j \beta_j x_j$$

Naïve Bayes vs Logistic Regression

- $\log \frac{P(y=c|x_1,\ldots,x_p)}{P(y=c'|x_1,\ldots,x_p)} = \beta_0 + \sum_j \beta_j x_j$

- Naïve Bayes: the ML estimate for the model if we assume that the features are independent given the class label
  - Definitely better if the conditional independence assumption is true
  - Possibly better if we want to constrain the model capacity and prevent overfitting

- Logistic regression: the ML estimate for the model
  - More capacity: arbitrary $\beta_j's$ allowed. Might be able to fit the data better
  - More likely to overfit