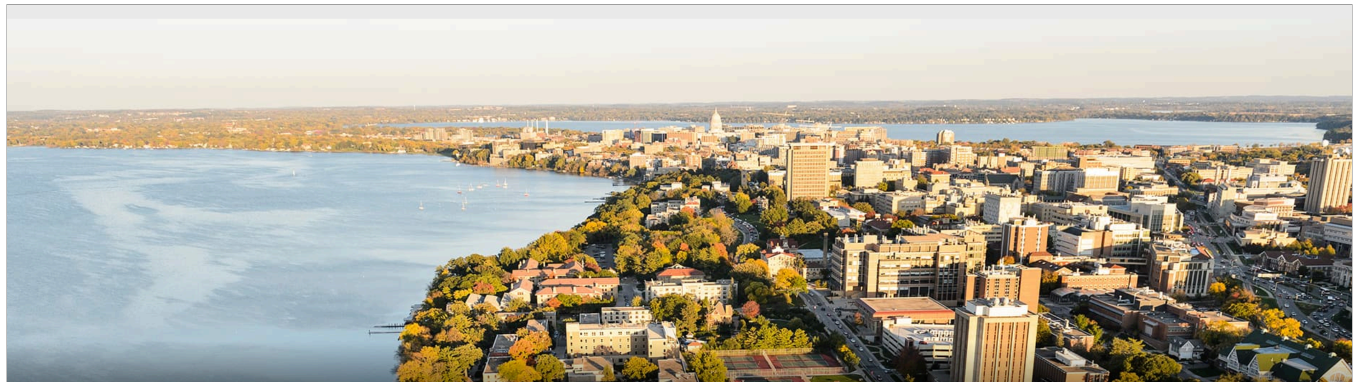


CS 540 Introduction to Artificial Intelligence

Classification - KNN and Naive Bayes

Yingyu Liang

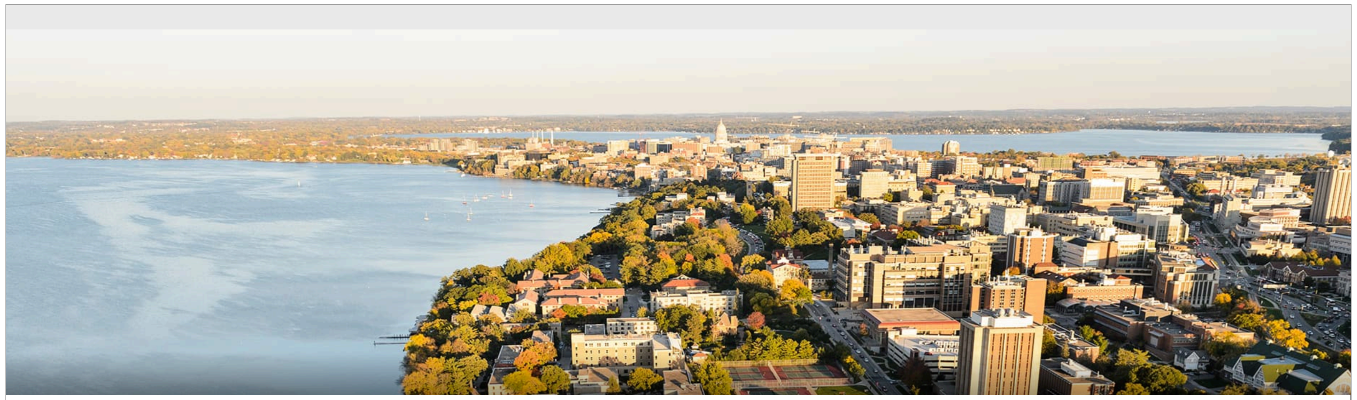
University of Wisconsin-Madison

Oct 14, 2021

Slides created by Sharon Li [modified by Yingyu Liang]

Today’s outline

- K-Nearest Neighbors
- Maximum likelihood estimation
- Naive Bayes

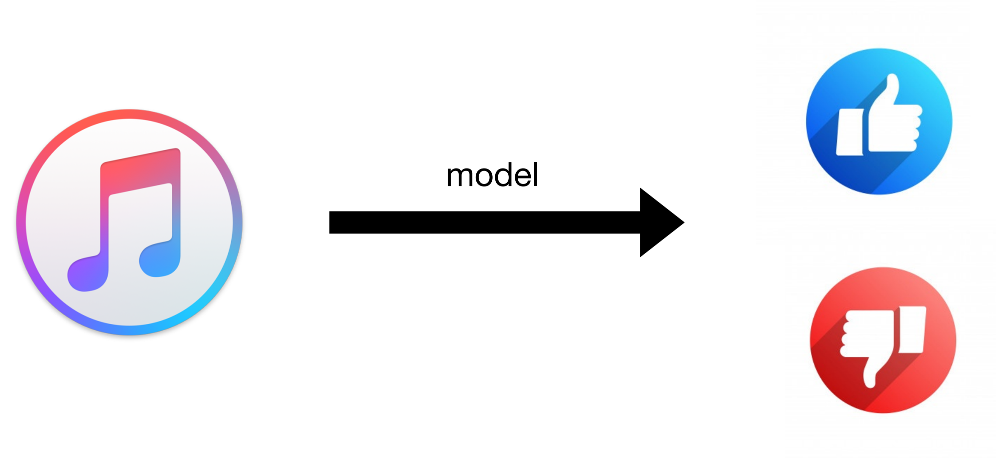


Part I: K-nearest neighbors


WIKIPEDIA
The Free Encyclopedia
Main page
Article
Talk
k-nearest neighbors algorithm
From Wikipedia, the free encyclopedia
Not to be confused with k-means clustering.

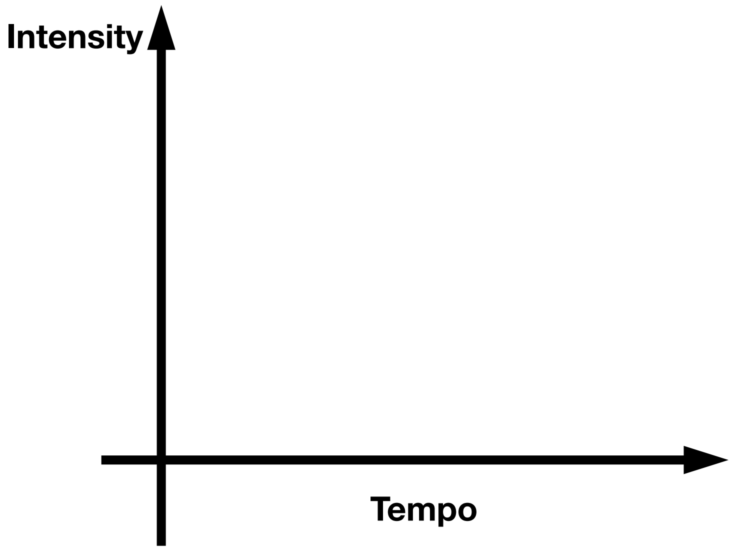
(source: wiki)

Example 1: Predict whether a user likes a song or not



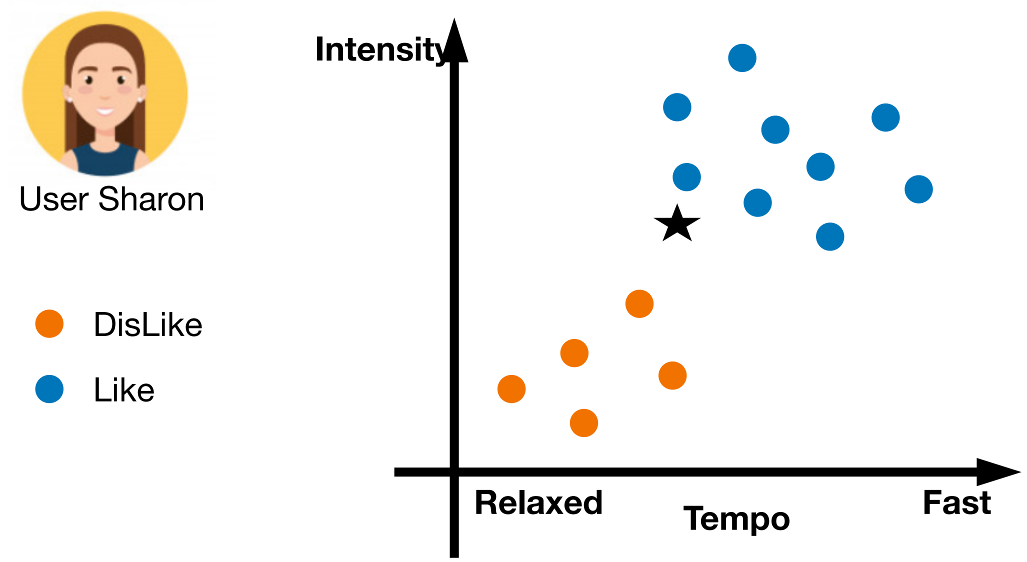
Example 1: Predict whether a user likes a song or not

User Sharon

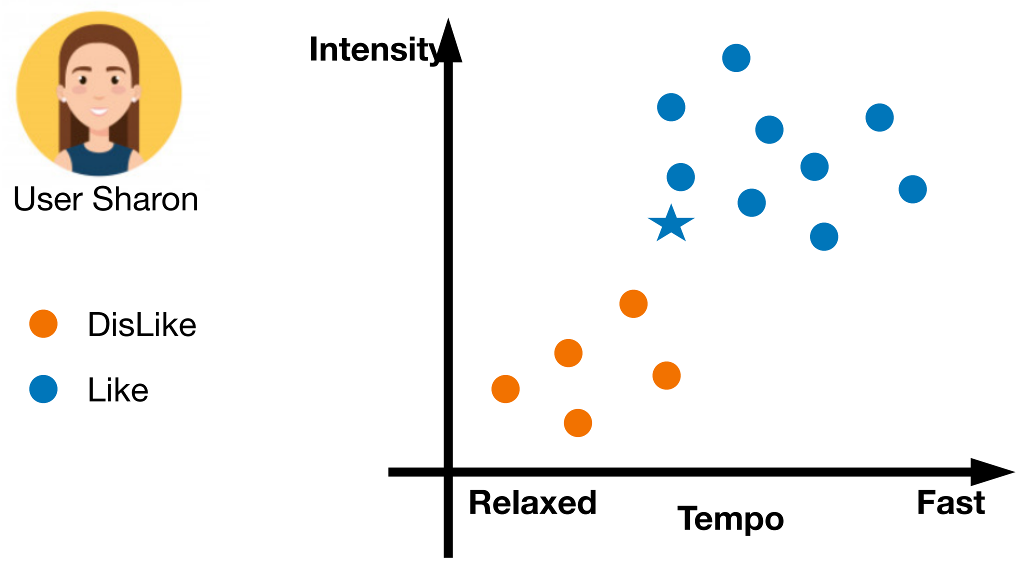


Example 1: Predict whether a user likes a song or not

1-NN



Example 1: Predict whether a user likes a song or not
1-NN



K-nearest neighbors for classification

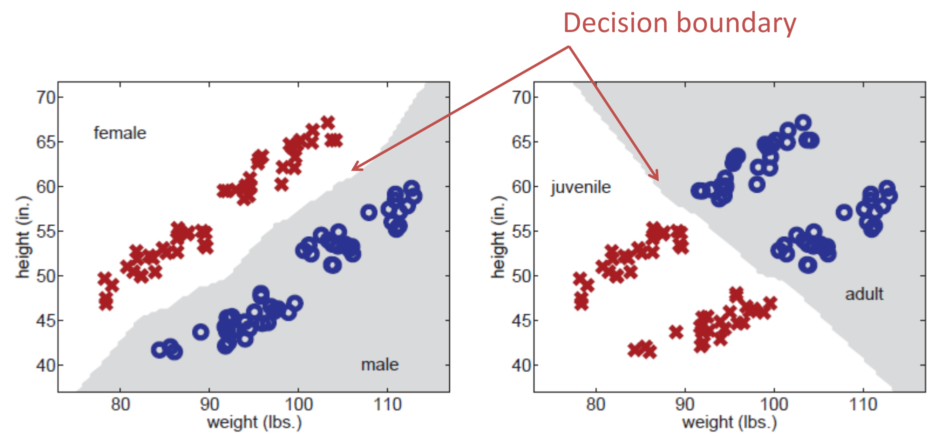
- **Input**: Training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
  Distance function $d(\mathbf{x}_i, \mathbf{x}_j)$; number of neighbors k; test data $\mathbf{x}^*$

1. Find the k training instances $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$ closest to $\mathbf{x}^*$ under $d(\mathbf{x}_i, \mathbf{x}_j)$
2. Output y^* as the majority class of $y_{i_1}, \ldots, y_{i_k}$. Break ties randomly.

Do nothing during the training.
When given a test input, find its k nearest neighbors in the training dataset, and do majority voting to predict the label.

Example 2: 1-NN for little green man

- Predict gender (M,F) from weight, height
- Predict age (adult, juvenile) from weight, height

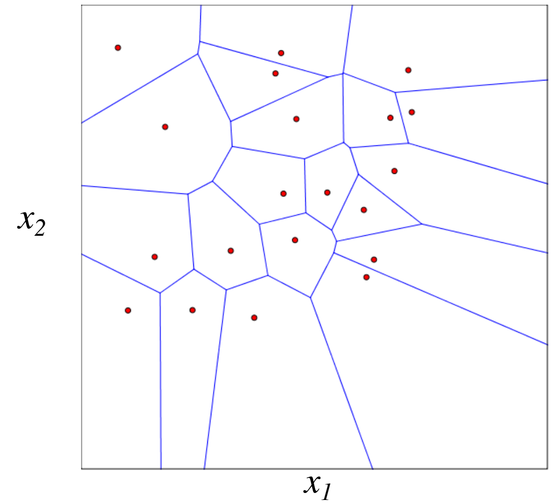


(a) classification by gender (b) classification by age

For any location in the input space, we predict its label using 1-NN. This determines the region of different predicted classes. The boundary between different classes is called the decision boundary.

The decision regions for 1-NN

Voronoi diagram: each polyhedron indicates the region of feature space that is in the nearest neighborhood of each training instance



Each red dot is a training data point.

1-NN divide the input space into regions. Each region will be given the label of the corresponding training data point.

K-NN for regression

- What if we want regression?

- Instead of majority vote, take average of neighbors' labels
  - Given test point $\mathbf{x}^*$, find its k nearest neighbors $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$
  - Output the predicted label $\frac{1}{k}(y_{i_1} + \ldots + y_{i_k})$

How can we determine distance?

suppose all features are discrete

- Hamming distance: count the number of features for which two instances differ

How can we determine distance?

suppose all features are discrete

- Hamming distance: count the number of features for which two instances differ

suppose all features are continuous

- Euclidean distance: sum of squared differences

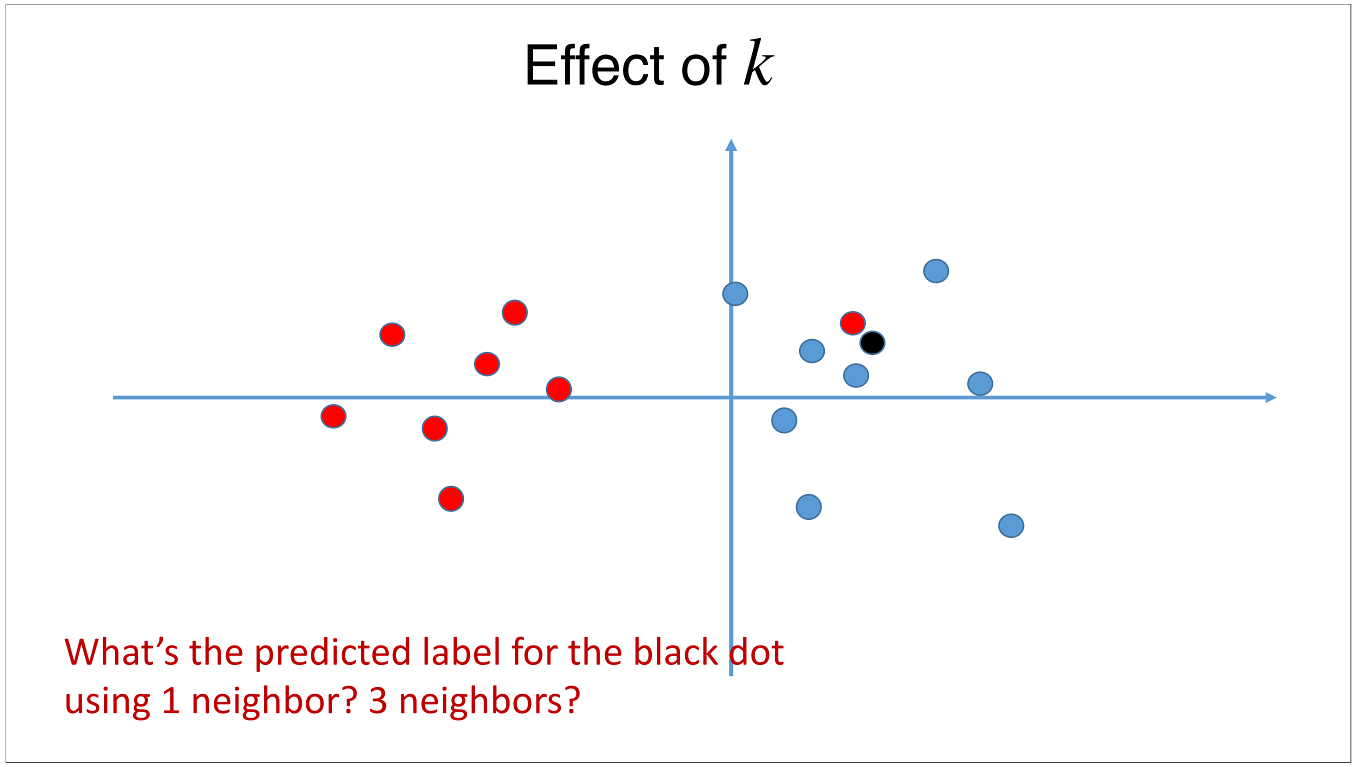
$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

- Manhattan distance:

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} |p_i - q_i|$$

How to pick the number of neighbors

- Split data into training and **tuning sets**
- Classify tuning set with different k
- Pick k that produces least tuning-set error



Small k: curvy decision boundary, sensitive to the noise. Can be viewed as having large model capacity
Large k: smooth decision boundary, not sensitive to the noise. Can be viewed as having small model capacity.
Extreme case k=#training data points: then any location in the input space will get the same prediction, ie, the prediction is a constant function.

Quiz break

Q1-1: K-NN algorithms can be used for:

- A Only classification
- B Only regression
- C Both

Quiz break

Q1-1: K-NN algorithms can be used for:

- A Only classification
- B Only regression
- C Both

Quiz break

Q1-2: Which of the following distance measure do we use in case categorical variables in k-NN?

- A Hamming distance
- B Euclidean distance
- C Manhattan distance

Quiz break

Q1-2: Which of the following distance measure do we use in case categorical variables in k-NN?

- A Hamming distance
- B Euclidean distance
- C Manhattan distance

Quiz break

Q1-3: Consider binary classification in 2D where the intended label of a point x = (x1, x2) is positive if x1>x2 and negative otherwise. Let the training set be all points of the form x = [4a, 3b] where a,b are integers. Each training item has the correct label that follows the rule above. With a 1NN classifier (Euclidean distance), which ones of the following points are labeled positive? Multiple answers.

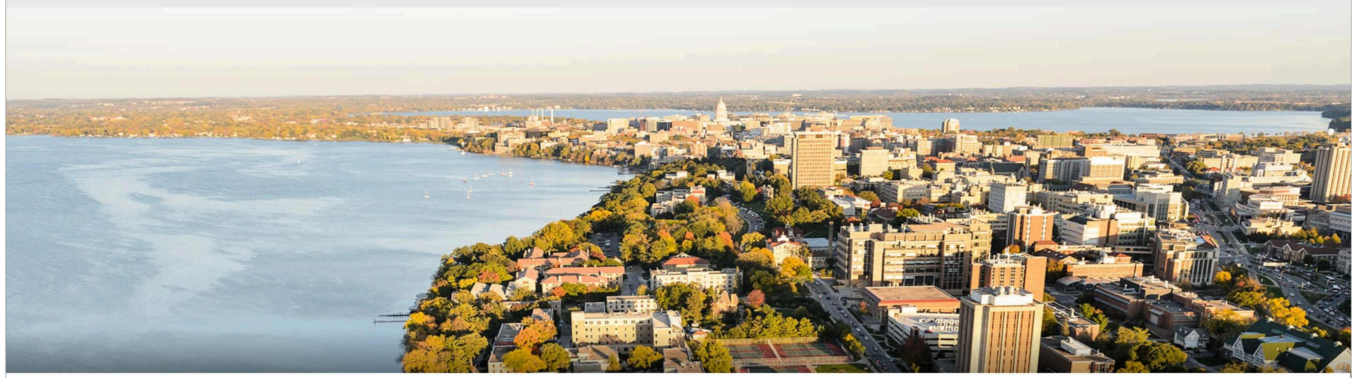
- [5.52, 2.41]
- [8.47, 5.84]
- [7,8.17]
- [6.7,8.88]

Quiz break

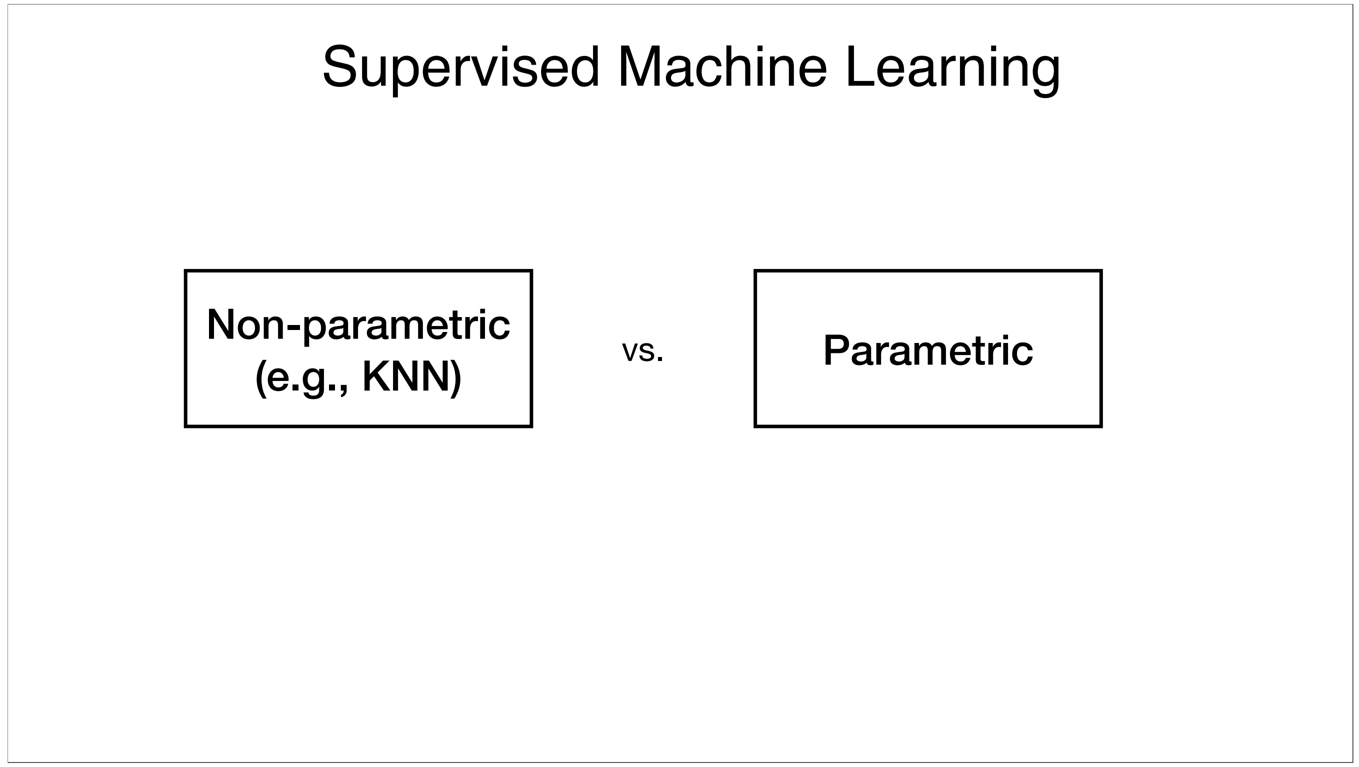
Q1-3: Consider binary classification in 2D where the intended label of a point x = (x1, x2) is positive if x1>x2 and negative otherwise. Let the training set be all points of the form x = [4a, 3b] where a,b are integers. Each training item has the correct label that follows the rule above. With a 1NN classifier (Euclidean distance), which ones of the following points are labeled positive? Multiple answers.

- [5.52, 2.41]
- [8.47, 5.84]
- [7,8.17]
- [6.7,8.88]

Nearest neighbors are
[4,3] => positive
[8,6] => positive
[8,9] => negative
[8,9] => negative
Individually.



Part II: Maximum Likelihood Estimation



Parametric here means using a class of functions with parameters.

Supervised Machine Learning

Statistical modeling approach

Labeled training data (n examples)

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$$

drawn **independently** from
a fixed underlying distribution
(also called the i.i.d. assumption)

Supervised Machine Learning

Statistical modeling approach

Labeled training data (n examples) → **Learning algorithm** → **Classifier** $\hat{f}$

$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

drawn **independently** from
a fixed underlying distribution
(also called the i.i.d. assumption)

select $\hat{f}(\theta)$ from a pool of models $\mathscr{F}$
that **best describe the data observed**

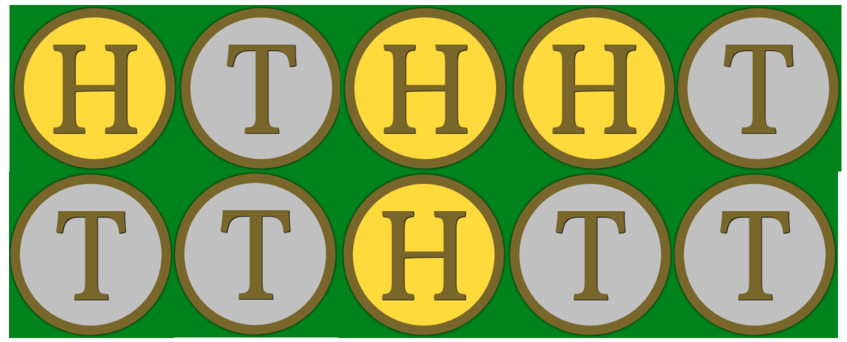
How to select $\hat{f} \in \mathscr{F}$?

- **Maximum likelihood (best fits the data)**
- Maximum a posteriori (best fits the data but incorporates prior assumptions)
- Optimization of ‘loss’ criterion (best discriminates the labels)

Note that some losses can be derived from MLE (Maximum Likelihood Estimation) or MAP (Maximum A Posteriori).

Maximum Likelihood Estimation: An Example

Flip a coin 10 times, how can you estimate $\theta = p(\text{Head})$?



Intuitively, $\theta = 4/10 = 0.4$

MLE is a general approach to estimate the parameter \theta of a distribution. Forget about labels for now.

Suppose we have a set of iid samples x_i's from a distribution p_\theta with parameter \theta. We want to estimate \theta. The given example: we have a set of 10 iid samples from the distribution of coin-flipping where the parameter is p(Head); we want to estimate p(Head).

MLE:

1. Write down the likelihood for different \theta values. (Usually use the log of the likelihood.)
2. Find the \theta value that can maximize the likelihood. (Or equivalently maximize the log-likelihood, since the log doesn't change the maximizer.)

How good is θ?

It depends on how likely it is to generate the observed data $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ (Let's forget about label for a second)

Likelihood function $L(\theta) = \Pi_i p(\mathbf{x}_i | \theta)$

Under i.i.d assumption

Interpretation: How **probable** (or how likely) is the data given the probabilistic model p_θ?

p(x_i | \theta): assume \theta is the truth, what is the probability of getting x_i?

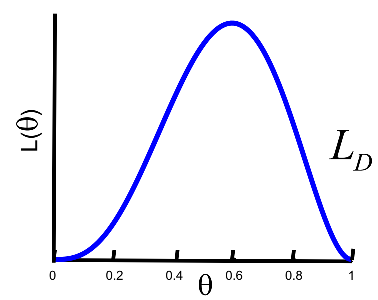
How good is θ?

It depends on how likely it is to generate the observed data

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ (Let's forget about label for a second)

Likelihood function $L(\theta) = \Pi_i p(\mathbf{x}_i \mid \theta)$

H,T, T, H, H



$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

Bernoulli distribution

Log-likelihood function

$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

$$= \theta^{N_H} \cdot (1-\theta)^{N_T}$$

Log-likelihood function

$$\ell(\theta) = \log L(\theta)$$

$$= N_H \log \theta + N_T \log(1-\theta)$$

Usually we use the log of the likelihood (called log-likelihood) which is convenient.

Maximum Likelihood Estimation (MLE)

Find optimal θ^* to maximize the likelihood function (and log-likelihood)

$$\theta^* = \arg\max N_H \log\theta + N_T \log(1-\theta)$$

$$\frac{\partial l(\theta)}{\partial \theta} = \frac{N_H}{\theta} - \frac{N_T}{1-\theta} = 0 \Rightarrow \theta^* = \frac{N_H}{N_T + N_H}$$

which confirms your intuition!

To maximize the

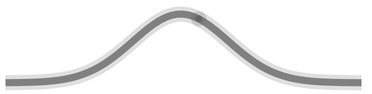
Maximum Likelihood Estimation: Gaussian Model

Fitting a model to heights of females

Observed some data (in inches): 60, 62, 53, 58,… $\in \mathbb{R}$

$$\{x_1, x_2, \ldots, x_n\}$$

Model class: Gaussian model



$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

So, what's the MLE for the given data?

Estimating the parameters in a Gaussian

- **Mean**

$$\mu = \mathbf{E}[x] \text{ hence } \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- **Variance**

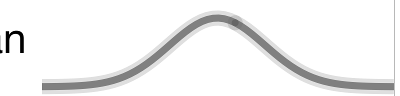
$$\sigma^2 = \mathbf{E}\left[(x-\mu)^2\right] \text{ hence } \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

Why?

Maximum Likelihood Estimation: Gaussian Model

Observe some data (in inches): $x_1, x_2, \ldots, x_n \in \mathbb{R}$

Assume that the data is drawn from a Gaussian



$$L(\mu, \sigma^2 \,|\, X) = \prod_{i=1}^{n} p(x_i; \mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

Fitting parameters is maximizing likelihood w.r.t μ, σ^2
(maximize likelihood that data was generated by model)

MLE $$\arg\max_{\mu, \sigma^2} \prod_{i=1}^{n} p(x_i; \mu, \sigma^2)$$

Maximum Likelihood

- Estimate parameters by finding ones that explain the data

$$\operatorname*{arg\,max}_{\mu,\sigma^2} \prod_{i=1}^{n} p(x_i; \mu, \sigma^2) = \operatorname*{arg\,min}_{\mu,\sigma^2} - \log \prod_{i=1}^{n} p(x_i; \mu, \sigma^2)$$

- **Decompose likelihood**

$$\sum_{i=1}^{n} \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(x_i - \mu)^2 = \frac{n}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2$$

Minimized for $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$

Maximum Likelihood

- Estimating the variance

$$\frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

Maximum Likelihood

- Estimating the variance

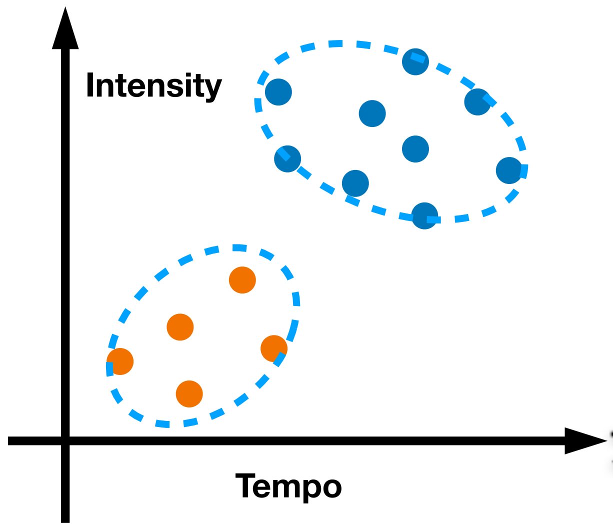
$$\frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

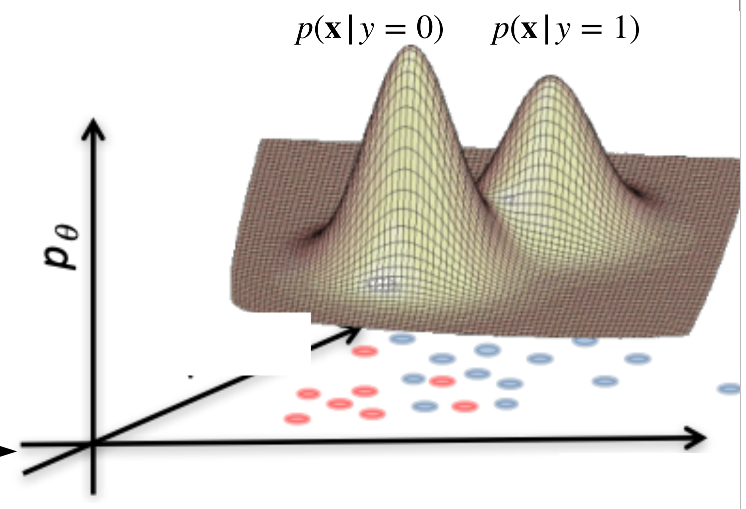
- Take derivatives with respect to it

$$\partial_{\sigma^2}[\,\cdot\,] = \frac{n}{2\sigma^2} - \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - \mu)^2 = 0$$

$$\implies \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Classification via Bayes' rule + MLE





Classification via Bayes' rule + MLE

$$\boxed{\hat{y}} = \hat{f}(\mathbf{x}) = \arg\max \boxed{p(y \mid \mathbf{x})} \quad \text{(Posterior)}$$

(Prediction)

Classification via Bayes' rule + MLE

$$\hat{y} = \hat{f}(\mathbf{x}) = \arg\max p(y \mid \mathbf{x}) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_{y} \frac{p(\mathbf{x} \mid y) \cdot p(y)}{p(\mathbf{x})} \quad \text{(by Bayes' rule)}$$

$$= \arg\max_{y} \; p(\mathbf{x} \mid y) p(y)$$

Then use MLE labelled training data, to learn **class conditionals** and **class priors**

Stages:

1. Formulate the decision making (ie, discrete prediction label for y) into a conditional probability problem p(y|x): the distribution over all possible labels given x.
2. Apply Bayes' rule, turn the problem into maximizing the product of class conditionals p(x|y) and class priors p(y)
3. Use the training data to estimate p(x|y) and p(y), and plug in the Bayes' rule to make the prediction. Can use MLE or MAP. We will talk about MLE.

Quiz break

Q2-2: True or False
Maximum likelihood estimation is the same regardless of whether we maximize the likelihood or log-likelihood function.

- A True
- B False

Quiz break

Q2-2: True or False
Maximum likelihood estimation is the same regardless of whether we maximize the likelihood or log-likelihood function.

- A True
- B False

Log is monotonically increasing so doesn't change the maximizer

Quiz break

Q2-3: Suppose the weights of randomly selected American female college students are normally distributed with unknown mean μ and standard deviation σ. A random sample of 10 American female college students yielded the following weights in pounds: 115 122 130 127 149 160 152 138 149 180. Find a maximum likelihood estimate of μ.

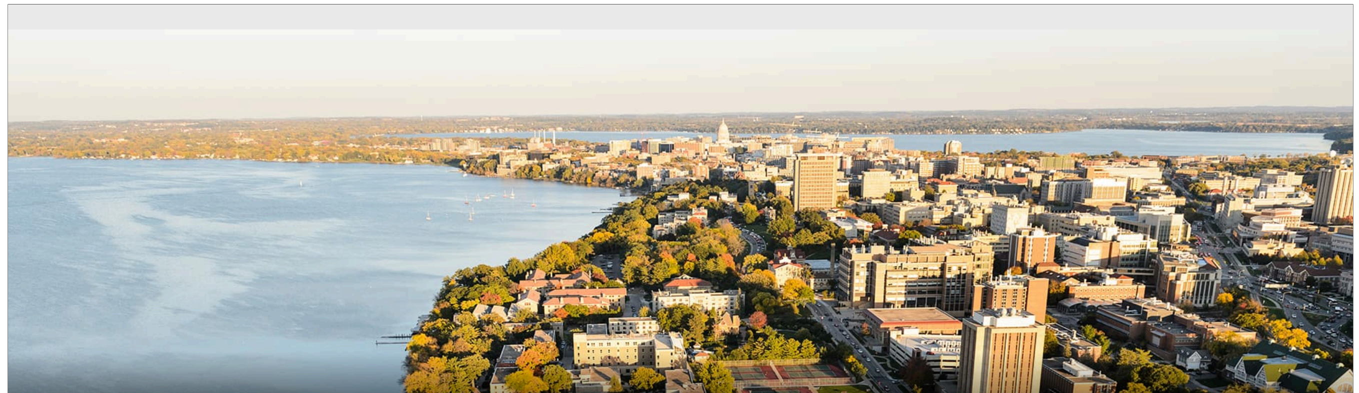
- A 132.2
- B 142.2
- C 152.2
- D 162.2

Quiz break

Q2-3: Suppose the weights of randomly selected American female college students are normally distributed with unknown mean μ and standard deviation σ. A random sample of 10 American female college students yielded the following weights in pounds: 115 122 130 127 149 160 152 138 149 180. Find a maximum likelihood estimate of μ.

- A 132.2
- B 142.2
- C 152.2
- D 162.2

Take the mean



Part II: Naïve Bayes

Recall the stages:

1. Formulate the decision making (ie, discrete prediction label for y) into a conditional probability problem p(y|x): the distribution over all possible labels given x.
2. Apply Bayes' rule, turn the problem into maximizing the product of class conditionals p(x|y) and class priors p(y)
3. Use the training data to estimate p(x|y) and p(y), and plug in the Bayes' rule to make the prediction.

Naive Bayes is using Naive Bayes assumption on p(x|y), to get p(x|y) = \prod_i p(x_i|y) where x_i is the i-th feature of the input x. Then use MLE to estimate p(x_i|y) and p(y). For discrete x, MLE is essentially counting.

Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

Posterior probability p(Yes | ☀️) vs. p(No | ☀️)

Stage 1: formulate the decision making problem (play outside or not) into a conditional probability problem: p(Play|sunny) for two labels Play=Yes and Play=No.

Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

Posterior probability p(Yes | ☀️) vs. p(No | ☀️)

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

Posterior probability p(Yes | ☀) vs. p(No | ☀)

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

$$p(\text{Play} \mid ☀) = \frac{p(☀ \mid \text{Play})\, p(\text{Play})}{p(☀)}$$

Bayes rule

Stage 2: apply Bayes rule. Need to estimate the terms p(sunny|Play) and p(Play).

Example 1: Play outside or not?

- **Step 1**: Convert the data to a frequency table of Weather and Play

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Stage 3: MLE to estimate the terms. Essentially counting (we have proved this for Bernoulli distribution in the coin-flipping example; a similar proof holds for the multinomial distribution.) Then plug in Bayes' rule to make the prediction

Example 1: Play outside or not?

Step 1: Convert the data to a frequency table of Weather and Play

Step 2: Based on the frequency table, calculate **likelihoods** and **priors**

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

p(Play = Yes) = 0.64

p(☀ | Yes) = 3/9 = 0.33

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Example 1: Play outside or not?

Step 3: Based on the likelihoods and priors, calculate posteriors

P(Yes| ☀)
=P(☀ |Yes)*P(Yes)/P(☀)

?

P(No| ☀)
=P(☀ |No)*P(No)/P(☀)

?

Example 1: Play outside or not?

Step 3: Based on the likelihoods and priors, calculate posteriors

P(Yes| ☀)
=P(☀ |Yes)*P(Yes)/P(☀)
=0.33*0.64/0.36
=0.6

P(No| ☀)
=P(☀ |No)*P(No)/P(☀)
=0.4*0.36/0.36
=0.4

P(Yes| ☀) > P(No| ☀) go outside and play!

Bayesian classification

$$\hat{y} = \arg\max p(y \mid \mathbf{x})$$ (Posterior)

(Prediction)

$$= \arg\max \frac{p(\mathbf{x} \mid y) \cdot p(y)}{p(\mathbf{x})}$$ (by Bayes' rule)

$$= \arg\max \; p(\mathbf{x} \mid y) p(y)$$

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_{y} p(y \mid X_1, \ldots, X_k)$$

(Prediction) (Posterior)

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\boxed{\hat{y}} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{\boxed{p(X_1, \ldots, X_k)}} \quad \text{(by Bayes' rule)}$$

Independent of y

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes' rule)}$$

$$= \arg\max_y \underbrace{p(X_1, \ldots, X_k \mid y)}_{\text{Class conditional likelihood}} \; p(y)$$

Class prior

Recall the stages:

1. Formulate the decision making (ie, discrete prediction label for y) into a conditional probability problem p(y|x): the distribution over all possible labels given x.
2. Apply Bayes' rule, turn the problem into maximizing the product of class conditionals p(x|y) and class priors p(y)
3. Use the training data to estimate p(x|y) and p(y), and plug in the Bayes' rule to make the prediction.

Naïve Bayes Assumption

Conditional independence of feature attributes

$$p(X_1, \ldots, X_k \mid y)p(y) = \Pi_{i=1}^{k} \boxed{p(X_i \mid y)} p(y)$$

Easier to estimate

(using MLE!)

Naive Bayes is using Naive Bayes (ie conditional indolence) assumption on p(x|y), to get p(x|y) = \prod_i p(x_i|y) where x_i is the i-th feature of the input x. Then use MLE to estimate each p(x_i|y) and p(y). For discrete x, MLE is essentially counting.

Quiz break

Q3-1: Which of the following about Naive Bayes is incorrect?

- A Attributes can be nominal or numeric
- B Attributes are equally important
- C Attributes are statistically dependent of one another given the class value
- D Attributes are statistically independent of one another given the class value
- E All of above

Quiz break

Q3-1: Which of the following about Naive Bayes is incorrect?

- A Attributes can be nominal or numeric
- B Attributes are equally important
- C Attributes are statistically dependent of one another given the class value
- D Attributes are statistically independent of one another given the class value
- E All of above

Naive Bayes assumption: Attributes are statistically **independent** of one another given the class value.

Quiz break

Q3-2: Consider a classification problem with two binary features, $x_1, x_2 \in \{0,1\}$, and $y \in \{1,2,\ldots,32\}$. Suppose $P(Y = y) = 1/32$, $P(x_1 = 1 | Y = y) = y/46$, $P(x_2 = 1 | Y = y) = y/62$. Which class will naive Bayes classifier produce on a test item with $x_1 = 1$ and $x_2 = 0$?

- A 16
- B 26
- C 31
- D 32

Quiz break

Q3-2: Consider a classification problem with two binary features, $x_1, x_2 \in \{0,1\}$, and $y \in \{1,2,\dots,32\}$. Suppose $P(Y = y) = 1/32$, $P(x_1 = 1 | Y = y) = y/46$, $P(x_2 = 1 | Y = y) = y/62$. Which class will naive Bayes classifier produce on a test item with $x_1 = 1$ and $x_2 = 0$?

- A 16
- B 26
- C 31
- D 32

Stage 1: need to estimate P(Y=y|x1=1, x2=0) for different y's.

Stage 2: Apply Bayes' rule and get
Prediction= \argmax_y p(x1=1,x2=0|Y=y)P(Y=y)

Stage 3: estimate the terms and plug in the Bayes' rule to make the prediction.
Apply Naive Bayes assumption:
p(x1=1,x2=0|Y=y)=p(x1=1|Y=y) p(x2=0|Y=y)
Then we have:
Prediction= \argmax_y p(x1=1|Y=y) p(x2=0|Y=y) p(Y=y)
= \argmax_y y/46 * (1-y/62) * 1/32
= \argmax_y y * (62-y)
= 31

Quiz break

Q3-3: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other. We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

- A Pass
- B Fail

Quiz break

Q3-3: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other. We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

- A Pass
- B Fail

Stage 1: need to estimate P(Y=y|Confident=Yes, Studied=Yes, Sick=No) for y in {Pass, Fail}.

Stage 2: Apply Bayes' rule and get
Prediction= \argmax_y p(Confident=Yes, Studied=Yes, Sick=No|Y=y)P(Y=y)

Stage 3: estimate the terms and plug in the Bayes' rule to make the prediction.

Apply Naive Bayes assumption:
p(Confident=Yes, Studied=Yes, Sick=No|Y=y)=p(Confident=Yes|Y=y) p(Studied=Yes|Y=y) p(Sick=No|Y=y)

Apply MLE on the training data (ie, counting):
1) For Y=Pass
p(Confident=Yes|Y=Pass) = 2/3,
p(Studied=Yes|Y=Pass) = 2/3,
p(Sick=No|Y=Pass) = 1/3,
p(Y=Pass) = 3/5

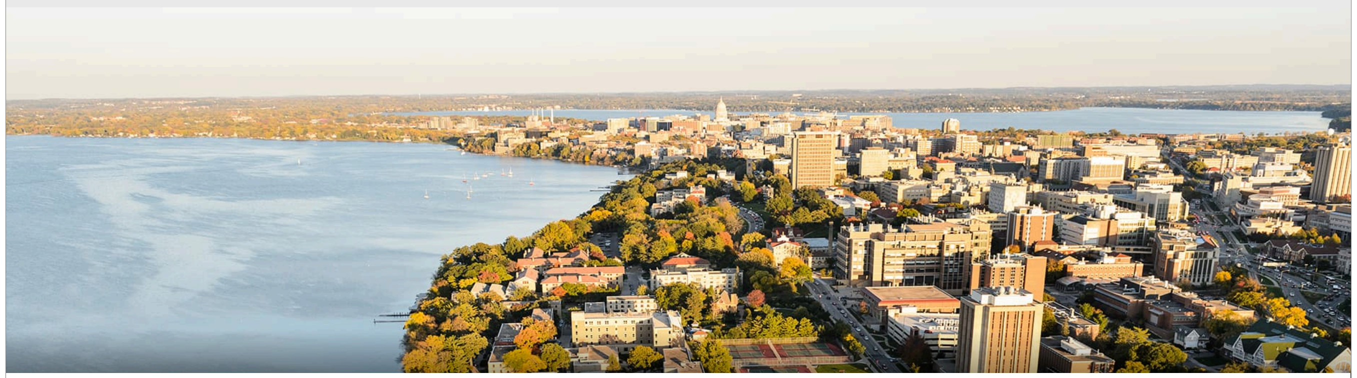
2) For Y=Fail
p(Confident=Yes|Y=Fail) = 1/2,
p(Studied=Yes|Y=Fail) = 1/2,

p(Sick=No|Y=Fail) = 1/2,
p(Y=Fail) = 2/5

Then we have:
p(Confident=Yes, Studied=Yes, Sick=No|Y=Pass)P(Y=Pass) = 2/3 * 2/3 * 1/3 * 3/5 = 4/9 * 1/5
p(Confident=Yes, Studied=Yes, Sick=No|Y=Fail)P(Y=Fail) = 1/2 * 1/2 * 1/2 * 2/5 = 1/4 * 1/5
The former is larger than the latter, so:
Prediction= \argmax_y p(Confident=Yes, Studied=Yes, Sick=No|Y=y)P(Y=y) = Pass

What we’ve learned today…

- K-Nearest Neighbors
- Maximum likelihood estimation
  - Bernoulli model
  - Gaussian model
- Naive Bayes
  - Conditional independence assumption



Thanks!