CS 540 Introduction to Artificial Intelligence

Classification - KNN and Naive Bayes

University of Wisconsin-Madison
Spring 2026 Sections 1 & 2

Announcements

- **Homework**:
  - HW4 due **Wednesday February 25th at 11:59 PM**
  - HW5 will also be released on Wednesday

- TA Discussion/Review Session every **Thursday at 5:30 PM in MH 3610**.

- Class roadmap:

Machine Learning: kNN& Naive Bayes
Machine Learning: Neural Networks I (Perceptron)
Machine Learning: Neural Networks II
Machine Learning: Neural Networks III

Supervised Learning

Reminder

Academic Integrity

You are encouraged to discuss with your peers, the TA or the instructors ideas, approaches and techniques broadly. However, all examinations, programming assignments, and written homeworks must be written up individually. For example, code for programming assignments must not be developed in groups, nor should code be shared. Make sure you work through all problems yourself, and that your final write-up is your own. If you feel your peer discussions are too deep for comfort, declare it in the homework solution: "I discussed with X,Y,Z the following specific ideas: A, B, C; therefore our solutions may have similarities on D, E, F…".

You may use books or legit online resources to help solve homework problems, but you must always credit all such sources in your writeup and you must never copy material verbatim.

Use of AI Tools: All submitted work must be your own. You may use artificial intelligence tools (like ChatGPT, Claude, or Cursor) in this class only as you might consult a peer for help, as outlined in the guidelines above. You may consult an AI tool to brainstorm approaches, clarify instructions, review concepts. You may ask for help with language or package syntax. You may use an AI tool for debugging help as long as you remain the primary problem-solver. If AI tools are employed, you are required to document their use by including comments that explain the code logic and providing full citations, including the specific prompts used. You may **not** use AI to generate and/or copy solutions, code, or written work, even partially. When in doubt, ask: "Would it be okay if a friend did this for me?" If the answer is no, it's not okay to have an AI do it either.

We are aware that certain websites host previous years' CS540 homework assignments and solutions against the wish of instructors. Do not be tempted to use them: the solutions may contain "poisonous berries" previous instructors planted intentionally to catch cheating. If we catch you copy such solutions, you automatically fail.

Do not bother to obfuscate plagiarism (e.g. change variable names, code style, etc.) One application of AI is to develop sophisticated plagiarism detection techniques!

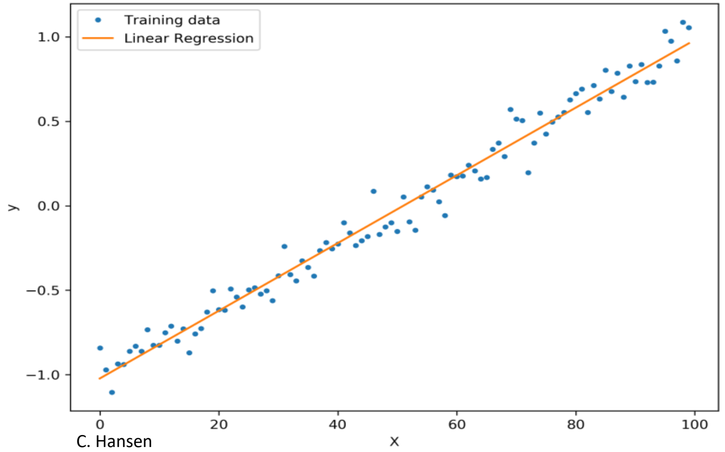
Cheating and plagiarism will be dealt with in accordance with University procedures (see the UW-Madison Academic Misconduct Rules and Procedures)

Outline

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

Review: Linear Regression

Simplest form of regression: find a line that fits the data



- When x is a scalar: $f_{(a,b)}(x) = ax + b$
- When x is a vector:

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d = \theta_0 + x^T \theta$$

Give x a "dummy dimension" to simplify notation

Old

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

New

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad f(x) = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_d \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ x_d \end{bmatrix} = \langle x, \theta \rangle = x^T \theta$$

Review: Linear Regression

Problem Setup

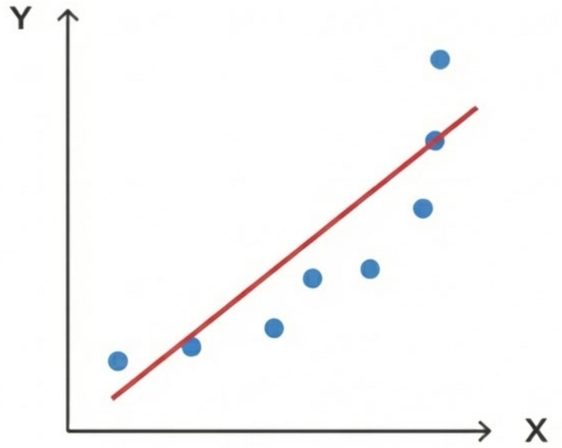
- Train set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
- Take train features and make it a n*(d+1) matrix, and y a vector:

$$X = \begin{bmatrix} x_1^T \\ \ldots \\ x_n^T \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \ldots \\ y_n \end{bmatrix}$$

- Loss: $\frac{1}{n}\|X\theta - y\|^2$
- Closed form solution: $\hat{\theta} = (X^T X)^{-1} X^T y$
- Another way to find the best parameters is using Gradient Descent and its variants

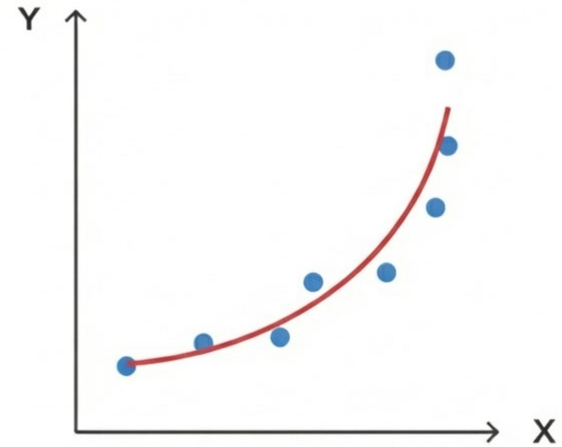
Polynomial Regression

- Linear regression tries to draw a straight line on your data.



- If data naturally curves, we may need to use polynomial regression.

$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^n$$



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.

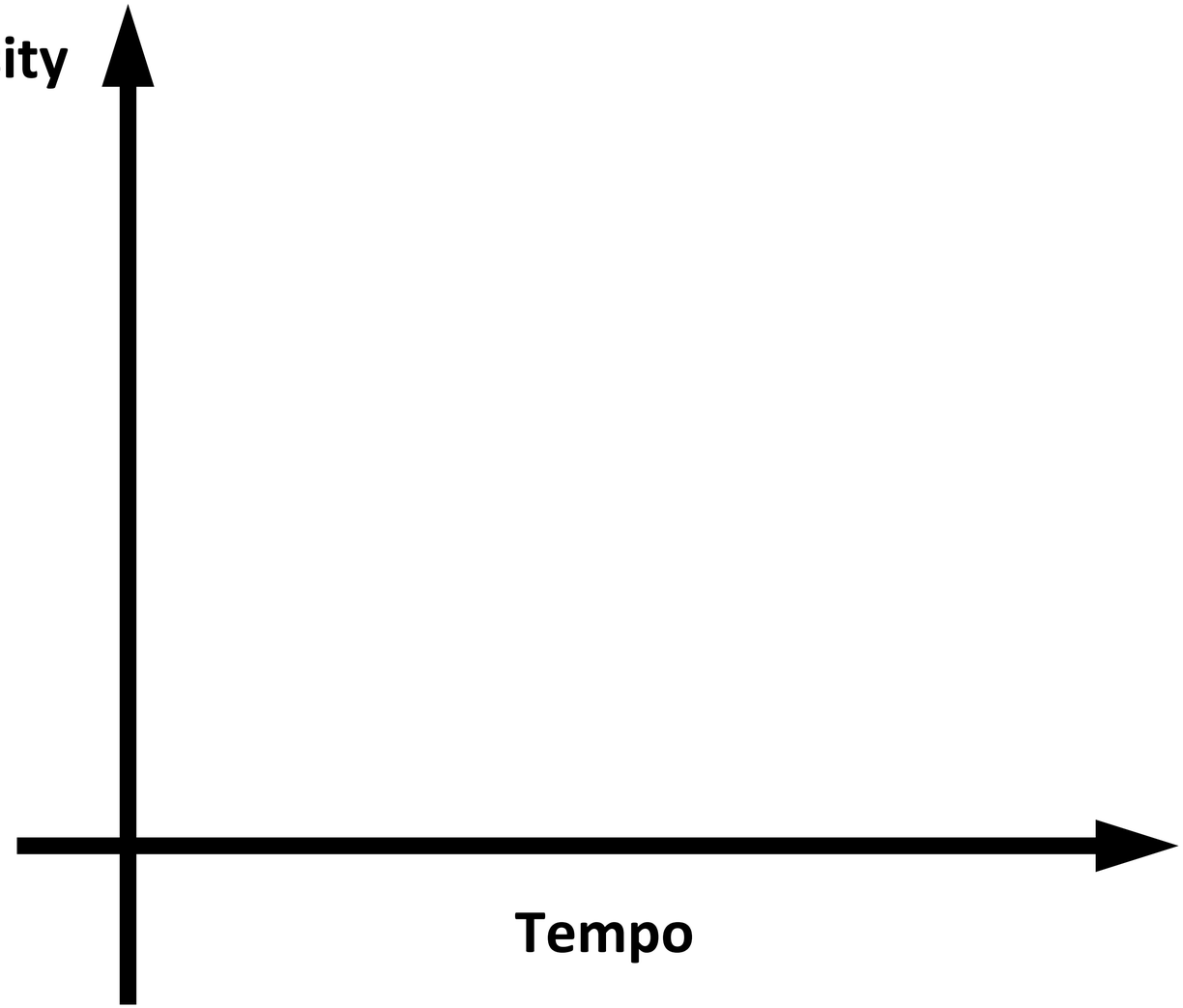
(source: wiki)

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

model

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

User Sharon

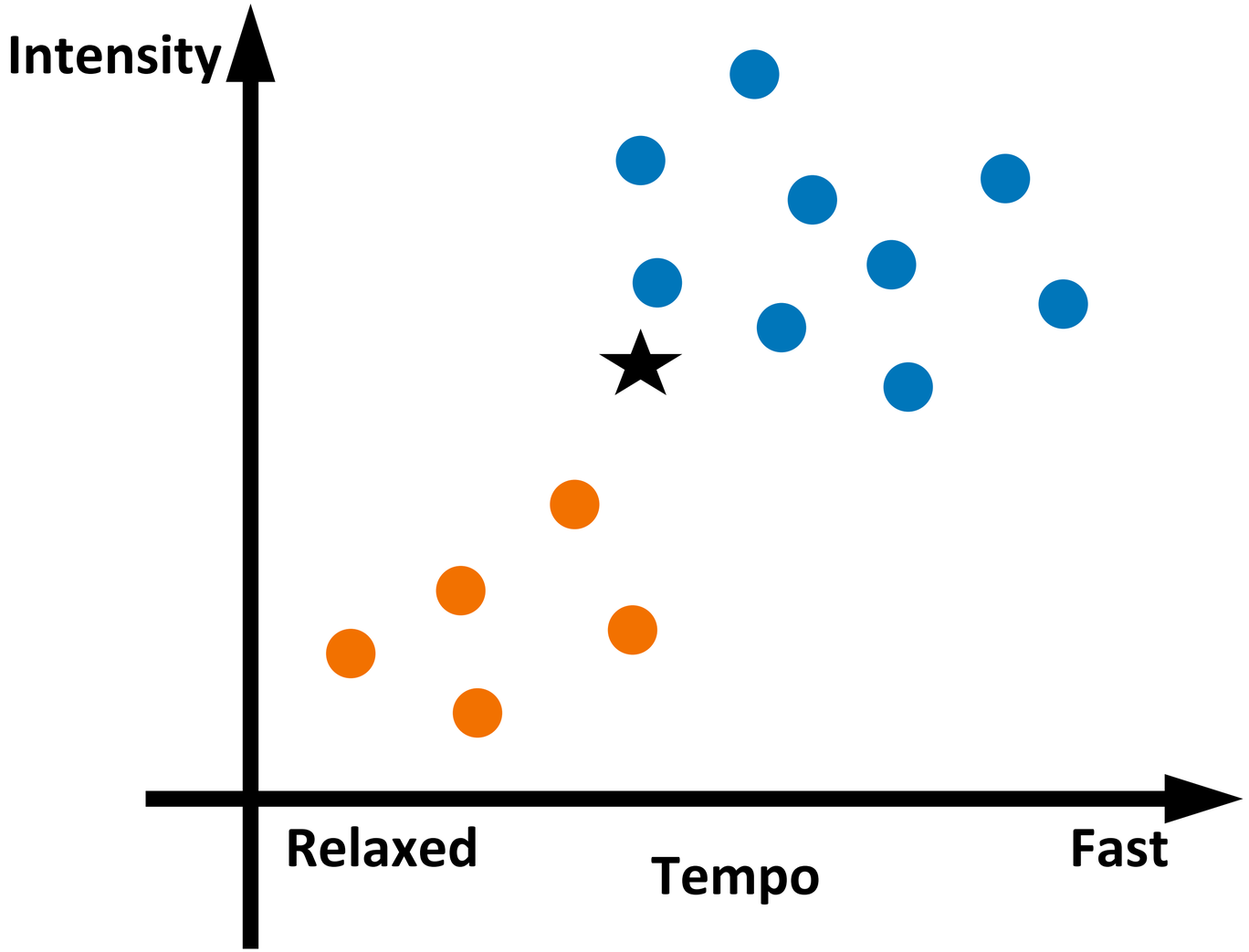


Example 1: Predict if a user likes a song or not 1-NN

User Sharon

Dislike

Like

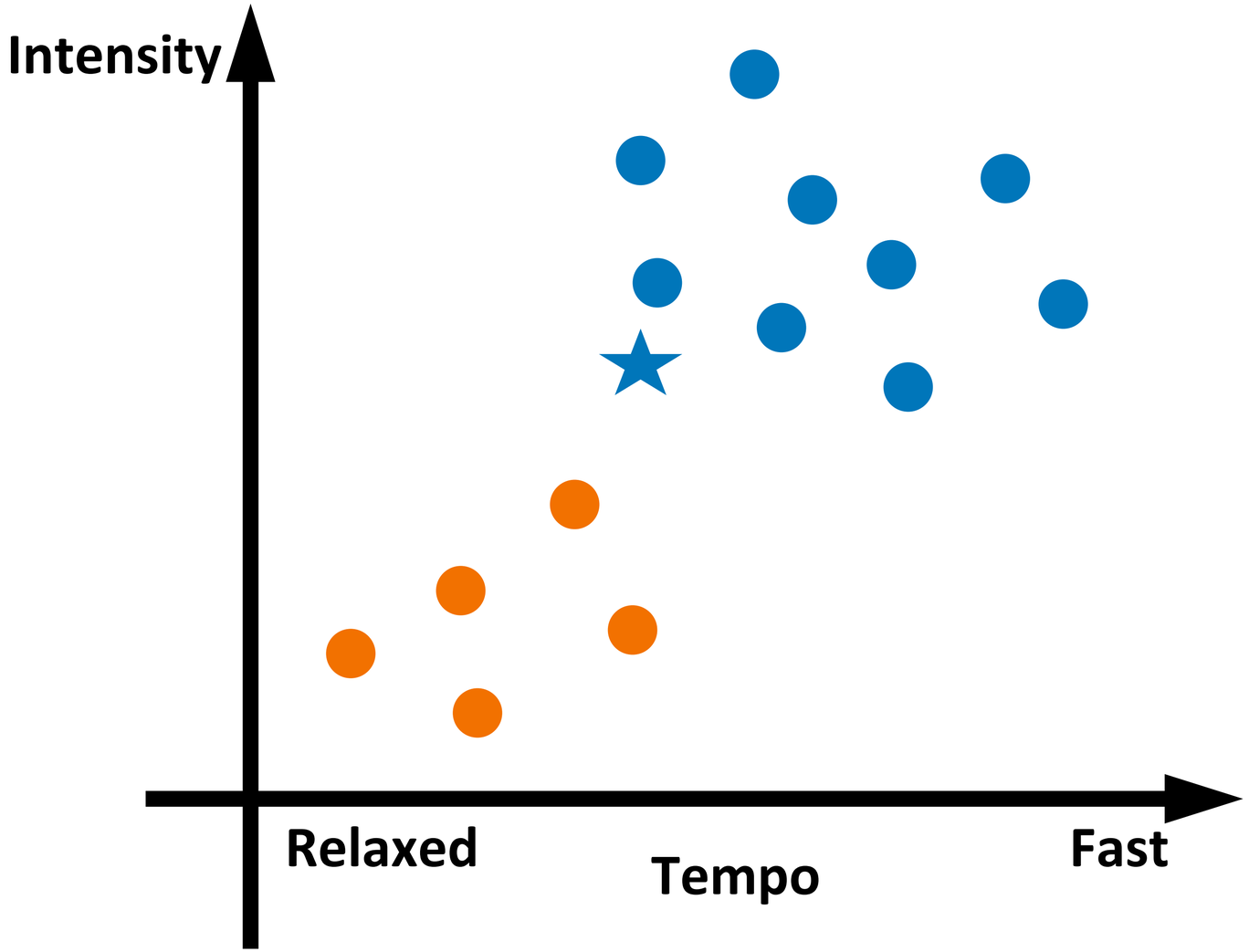


Example 1: Predict if a user likes a song or not 1-NN

User Sharon

Dislike

Like



K-nearest neighbors for classification

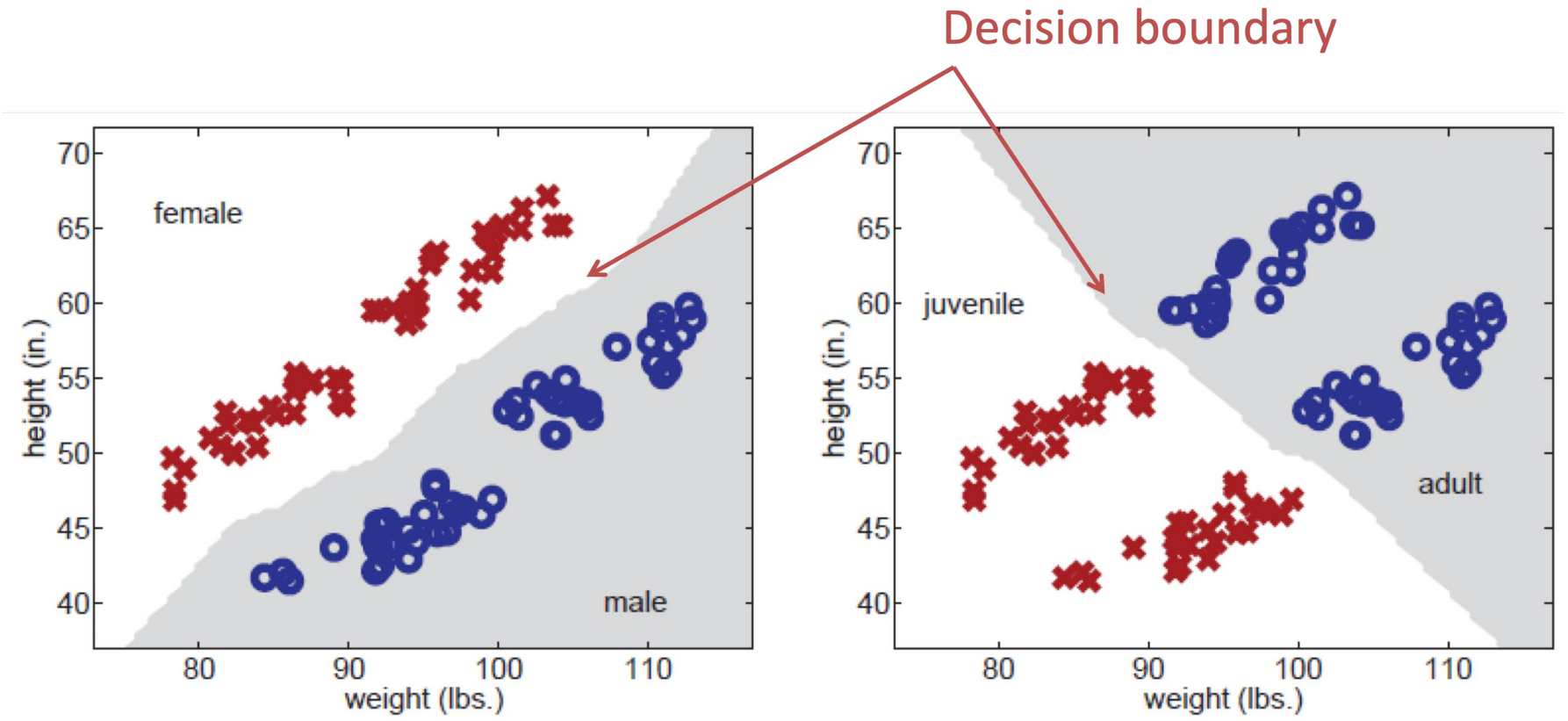
- **Input**: **Training data** $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

  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^*, the majority class of $y_{i_1}, \ldots, y_{i_k}$. Break ties randomly.

Example 2: 1-NN for little green man

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



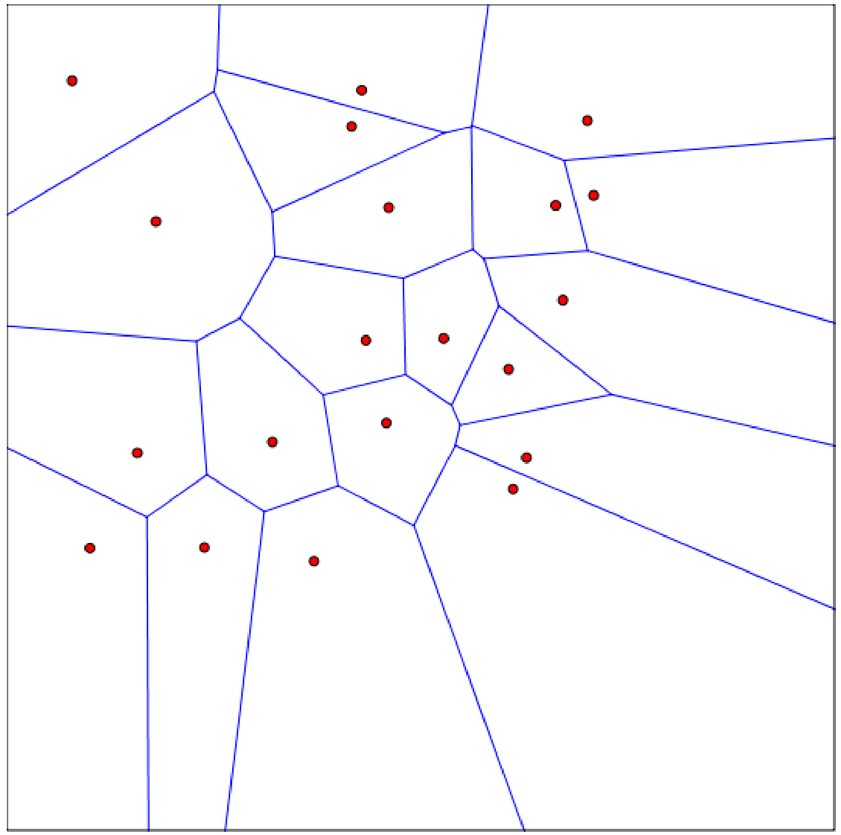
(a) classification by gender

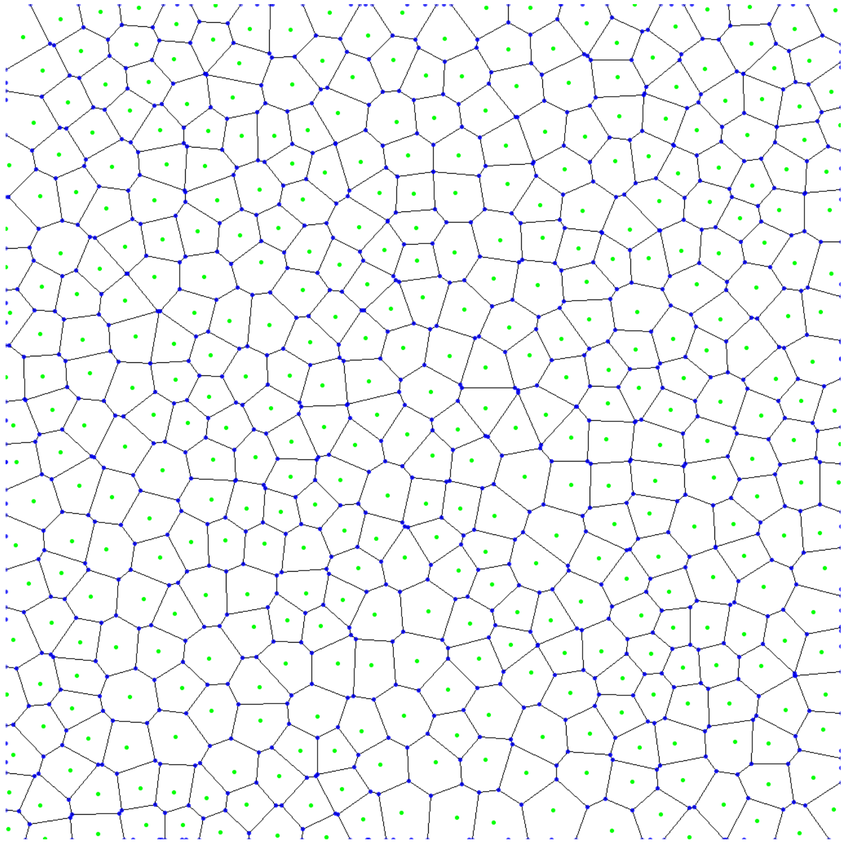
(b) classification by age

1NN: Decision Regions

Defined by "**Voronoi Diagram**"

- Each cell contains points closer to a particular training point





k-Nearest Neighbors: Distances

Discrete features: Hamming distance

$$d_H(x^{(i)}, x^{(j)}) = \sum_{a=1}^{d} 1\{x_a^{(i)} \neq x_a^{(j)}\}$$

Continuous features:

- Euclidean distance:

$$d(x^{(i)}, x^{(j)}) = \left(\sum_{a=1}^{d} (x_a^{(i)} - x_a^{(j)})^2 \right)^{\frac{1}{2}}$$

- L1 (Manhattan) dist.:

$$d(x^{(i)}, x^{(j)}) = \sum_{a=1}^{d} |x_a^{(i)} - x_a^{(j)}|$$

k-Nearest Neighbors: Regression

Training/learning: given

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$$

Prediction: for x , find ***k*** most similar training points

Return

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y^{(i)}$$

- I.e., among the ***k*** points, output mean label.

More on **distance functions**…

- Be careful with **scale**
- Same feature but different units may change relative distance (fixing other features)
- Sometimes OK to normalize each feature dimension

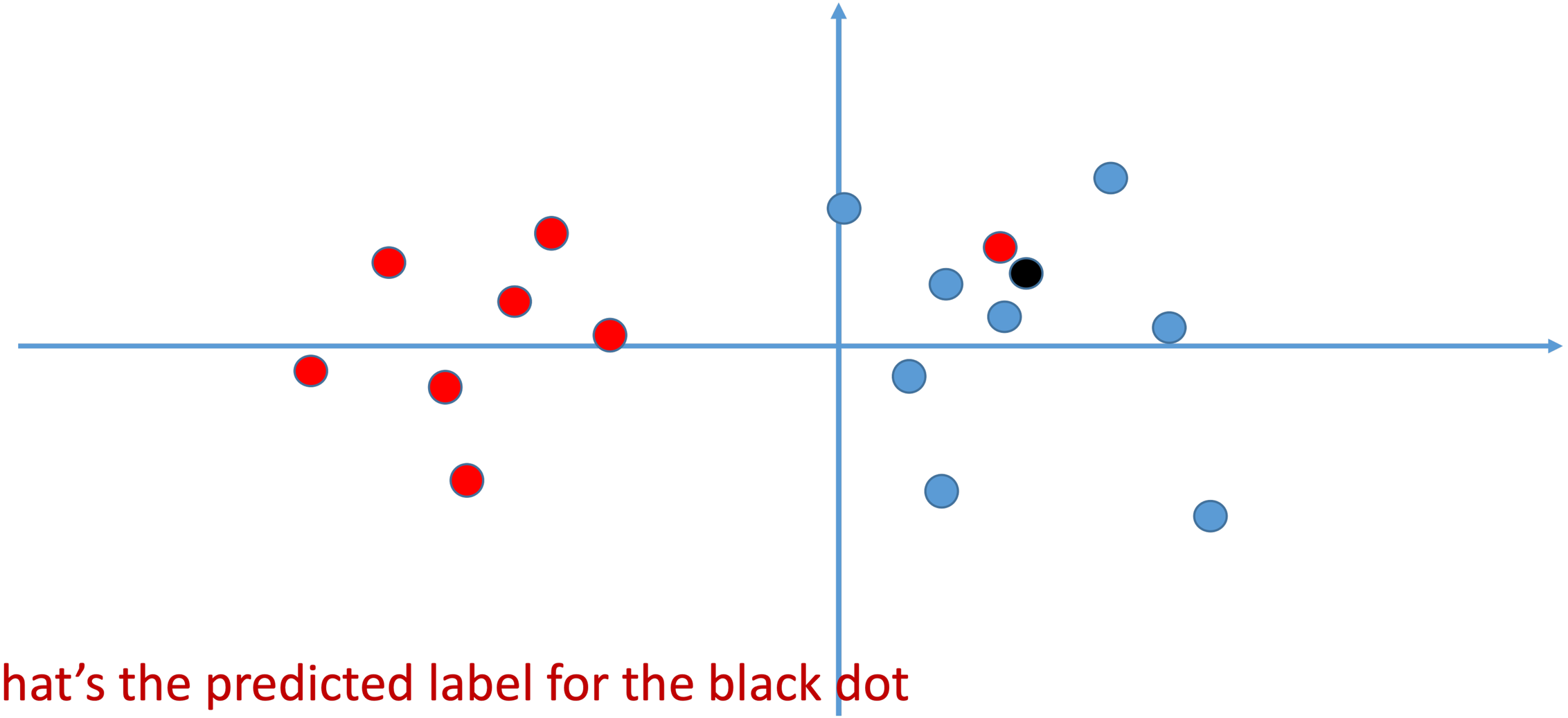
Training set mean for dimension d

$$x'_{id} = \frac{x_{id} - \mu_d}{\sigma_d}, \forall i = 1 \ldots n, \forall d$$

Training set standard deviation for dimension d

- Other times not OK: e.g. dimension contains small random noise

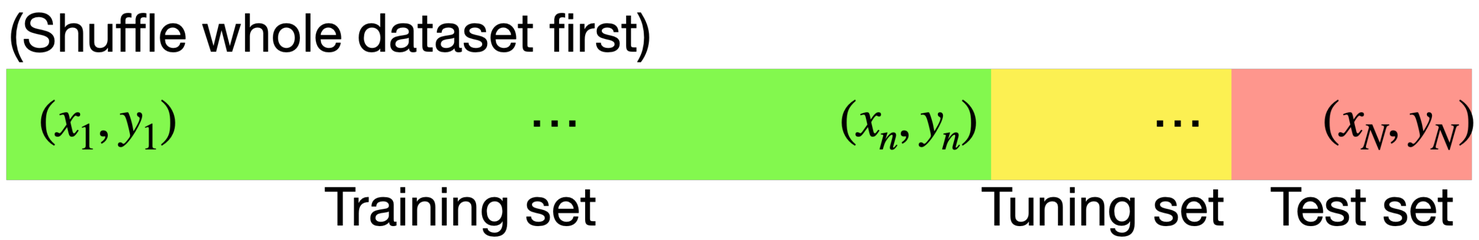
Effect of k



What's the predicted label for the black dot using 1 neighbor? 3 neighbors?

How to pick k, **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



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 of 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 of 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

Supervised Machine Learning

Non-parametric
(e.g., KNN)

vs.

Parametric

Supervised Machine Learning

Statistical modeling approach

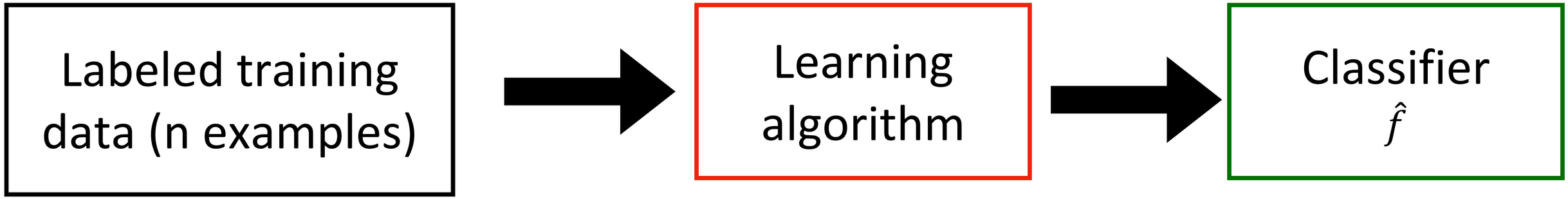
Labeled training data (n examples)

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

drawn **independently** from
a fixed underlying distribution,
also called the i.i.d.
(independent and identically distributed)
assumption

Supervised Machine Learning

Statistical modeling approach



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

drawn **independently** from
a fixed underlying distribution,
also called the i.i.d.
(independent and identically distributed)
assumption

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

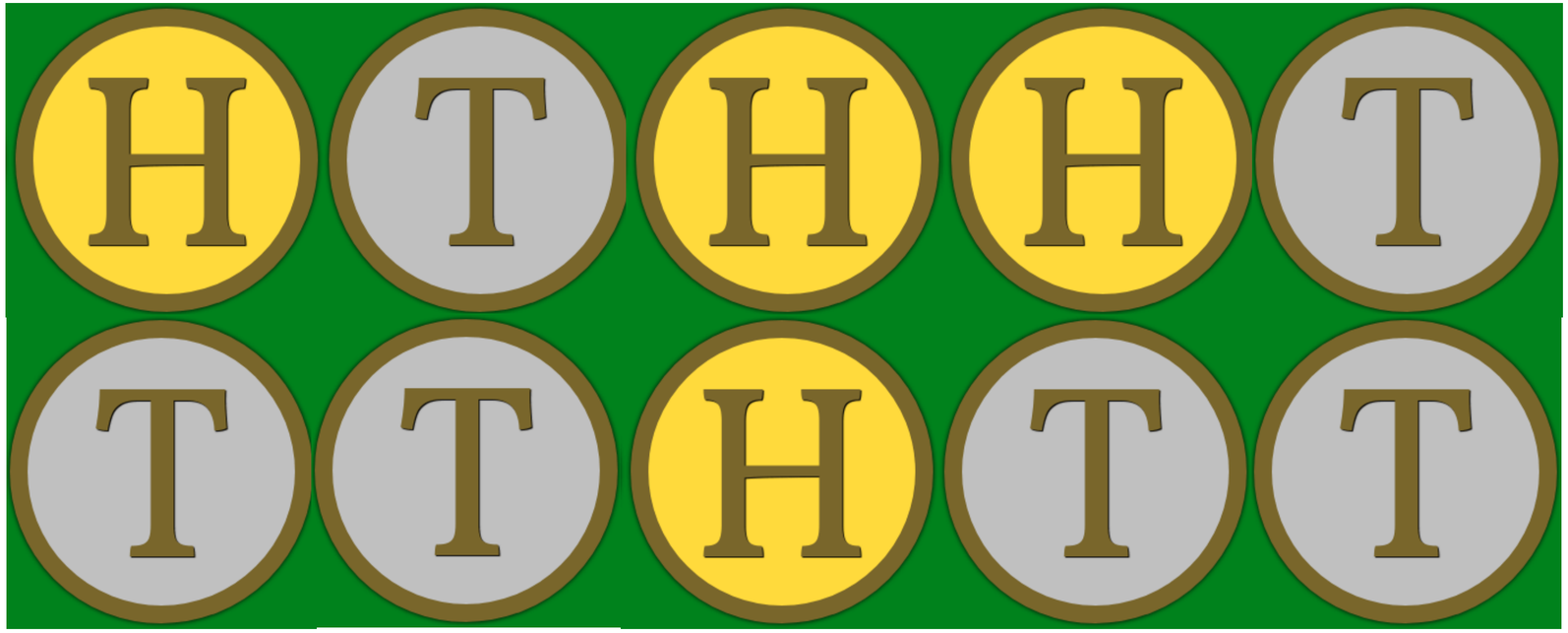
How to select $\hat{f} \in \mathcal{F}$?

- **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)

Maximum Likelihood Estimation: An Example

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



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

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 $L(\theta) = \prod_i p(\mathbf{x}_i|\theta)$

Under i.i.d assumption

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

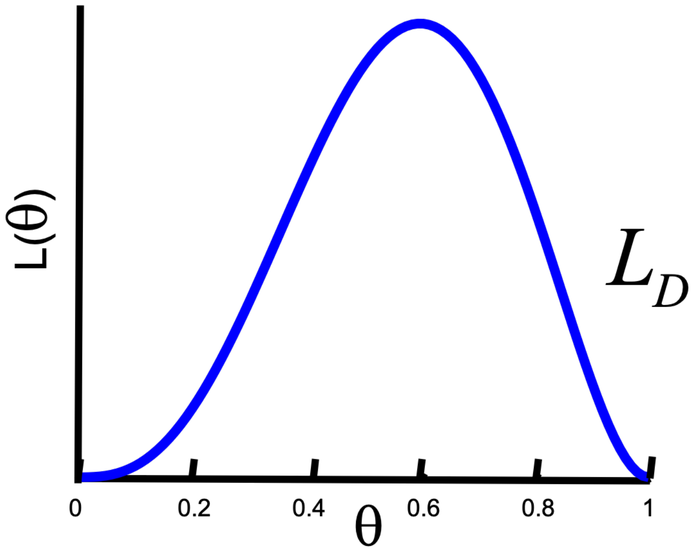
How good is θ?

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

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

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

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

Maximum Likelihood Estimation (MLE)

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

$$\theta^* = \text{argmax}\ N_H \log\theta + N_T \log(1-\theta)$$

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

which confirms your intuition!

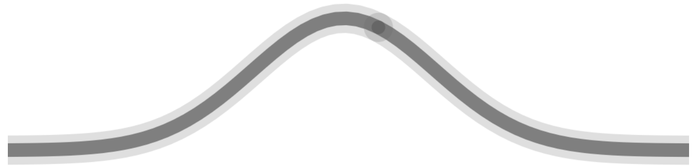
Maximum Likelihood Estimation: Gaussian Model

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**

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

- **Variance**

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

Why?

Maximum Likelihood Estimation: Gaussian Model

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

We assume that data are drawn from a Gaussian model.

Likelihood:

$$L(\mu, \sigma^2) = \prod_{i=1}^{n} p(x_i \mid \mu, \sigma^2) = \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2} \right) \right]$$

Fitting parameters is maximizing likelihood w.r.t to $\boldsymbol{\mu}, \boldsymbol{\sigma}^2$
(maximize likelihood that data was generated by the model)

Maximum Likelihood Estimation: Gaussian Model

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

Equivalently we can maximize the log likelihood:

$$\arg\max_{\mu,\sigma^2} \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)\right)$$

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

Maximum Likelihood Estimation: Gaussian Model

Setting the partial derivatives to zero:

$$\frac{\partial L}{\partial \mu} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0 \qquad \Rightarrow \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\frac{\partial L}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i - \mu)^2 = 0 \qquad \Rightarrow \sigma = \sqrt{\sum_{i=1}^{n}\left(\frac{(x_i - \mu)^2}{n}\right)}$$

Classification via MLE

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

(Prediction)

(Posterior)

Classification via MLE

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

(Prediction) (Posterior)

$$= \arg\max_y \frac{p(\mathbf{x} \mid y) \cdot p(y)}{p(\mathbf{x})}$$

(by Bayes' rule)

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

Using labelled training data, learn **class priors** and **class conditionals**

Quiz break

Q2-1: 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

Q2-1: True or False

Maximum likelihood estimation is the same regardless of whether we maximize the likelihood or log-likelihood function.

- A True
- B False

Part III: Naïve Bayes

Example 1: Play outside or not?

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

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

Example 1: Play outside or not?

- If weather is sunny, would you like 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 like 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}

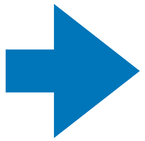
$$p(\text{Play} \mid ☀) = \frac{p(☀ \mid \text{Play})\, p(\text{Play})}{p(☀)}$$

Bayes rule

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



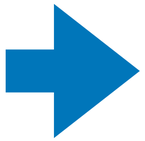
Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

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

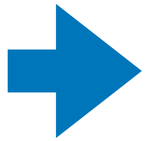
Example 1: Play outside or not?

- **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 $\mathbf{x} = \{X_1, \ldots, X_k\}$

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

(Posterior)

(Prediction)

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)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)}$$

(by Bayes' rule)

Independent of y

Bayesian classification

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

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

(Prediction)

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

$$= \arg\max_y \; p(X_1, \ldots, X_k \mid y)\; p(y)$$

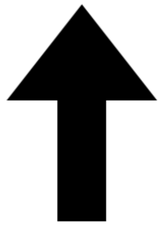
Class conditional likelihood

Class prior

Naïve Bayes Assumption

Conditional independence of feature attributes

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



Easier to estimate

(using MLE!)

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

Quiz break

Q3-2: 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

Q3-2: 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

We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

- A Pass
- B Fail

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

$$P(y = P|x_1 = Y, x_2 = Y, x_3 = N)$$
$$= \frac{P(x_1 = Y|Y = P)P(x_2 = Y|Y = P)P(x_3 = N|Y = P)P(y = P)}{P(x_1 = Y, x_2 = Y, x_3 = N)}$$
$$= \frac{2}{3} * \frac{2}{3} * \frac{1}{3} * \frac{3}{5} / P(x_1 = Y, x_2 = Y, x_3 = N)$$

$$\propto \frac{4}{9*5}$$ Larger!

$$P(y = F|x_1 = Y, x_2 = Y, x_3 = N)$$
$$= \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{2}{5} / P(x_1 = Y, x_2 = Y, x_3 = N)$$

$$\propto \frac{1}{4 * 5}$$

What we've learned today…

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

Suggested Readings

- Textbook: Artificial Intelligence: A Modern Approach (4th edition). Stuart Russell and Peter Norvig. Pearson, 2020.
  - Sections 19.7.1, 20.1, 20.2.1-20.2.4

Thanks!