CS 540 Introduction to Artificial Intelligence
Classification - KNN and Naive Bayes

University of Wisconsin–Madison
Fall 2025, Section 3
October 1, 2025

Announcements

- HW3 due Friday 10/3 at 11:59 PM

- Class roadmap:

ML: Unsupervised Learning
ML Linear Regression
Machine Learning: K - Nearest Neighbors & Naive Bayes
Machine Learning: Neural Networks I (Perceptron)
Machine Learning: Neural Networks II

Supervised Learning

Outline

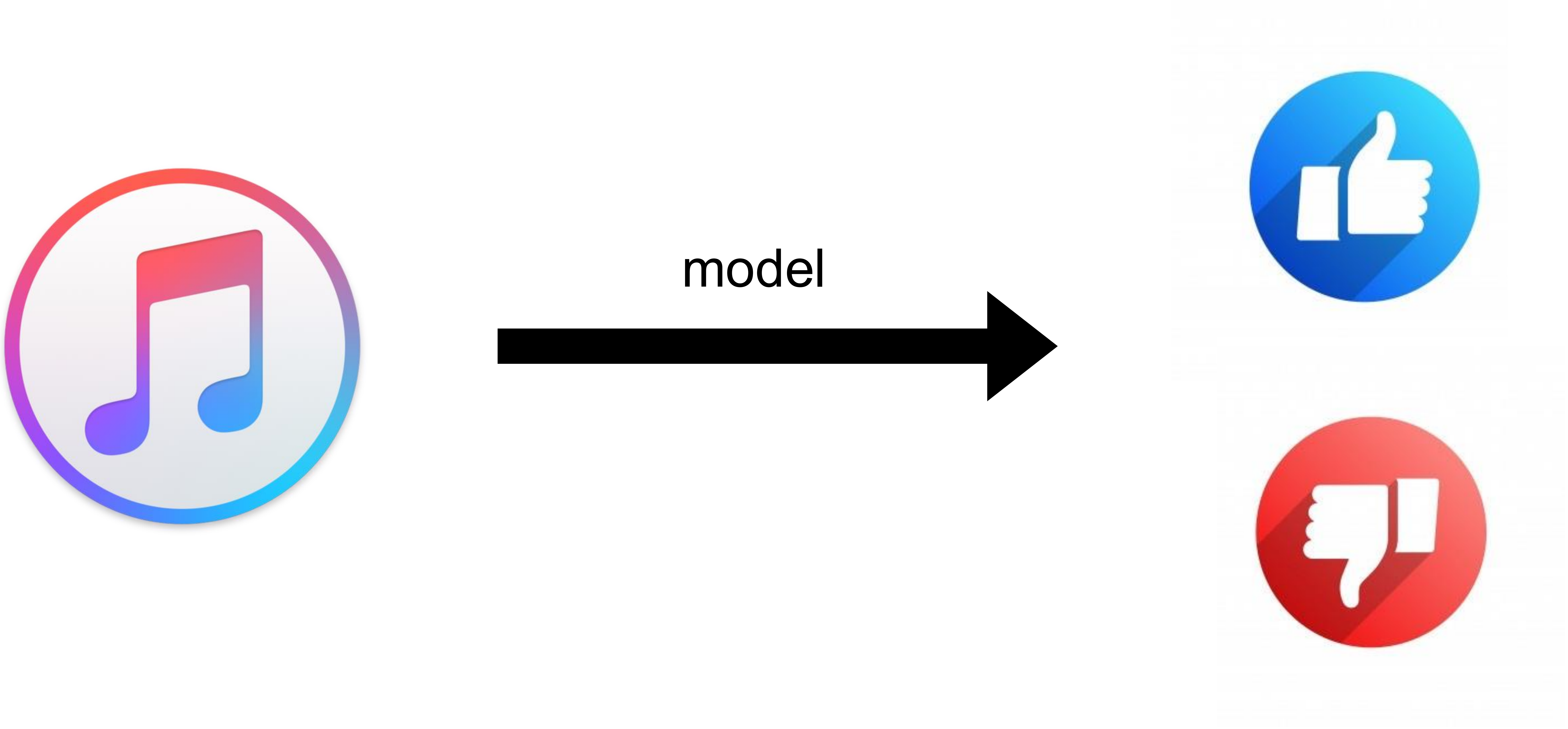
- K-Nearest Neighbors
- Maximum likelihood estimation
- Naive Bayes

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.

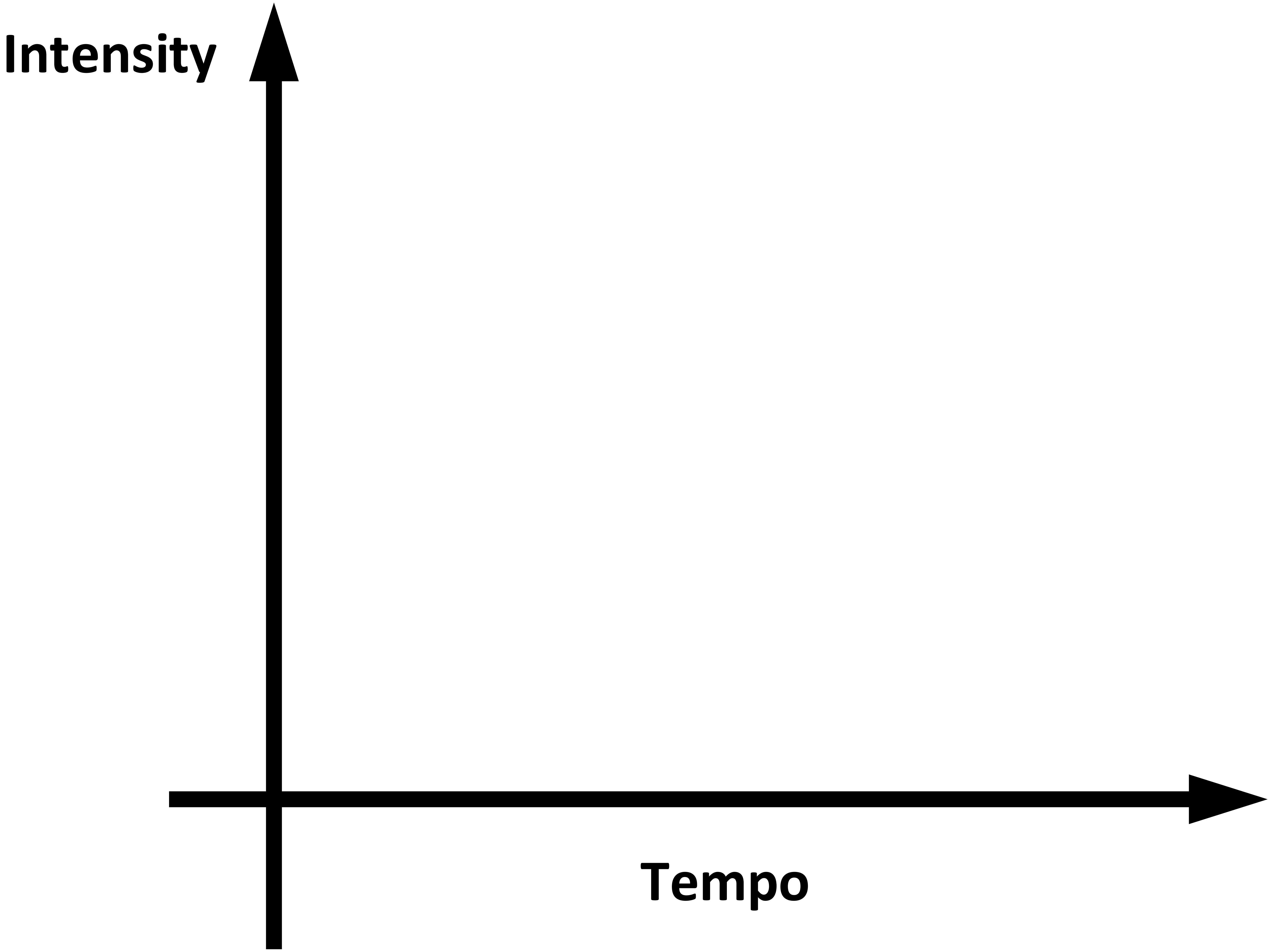
(source: wiki)

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



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

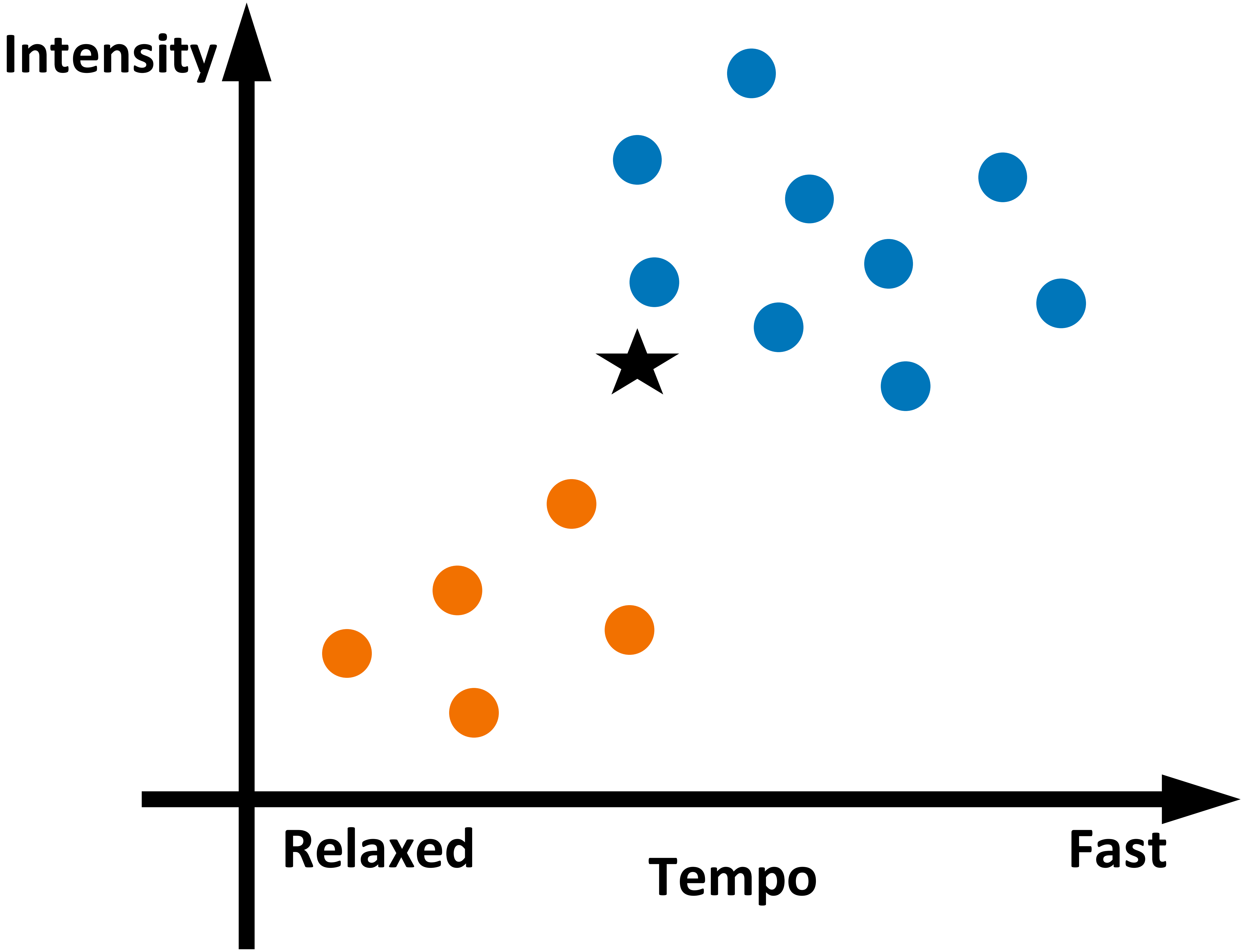
User Sharon



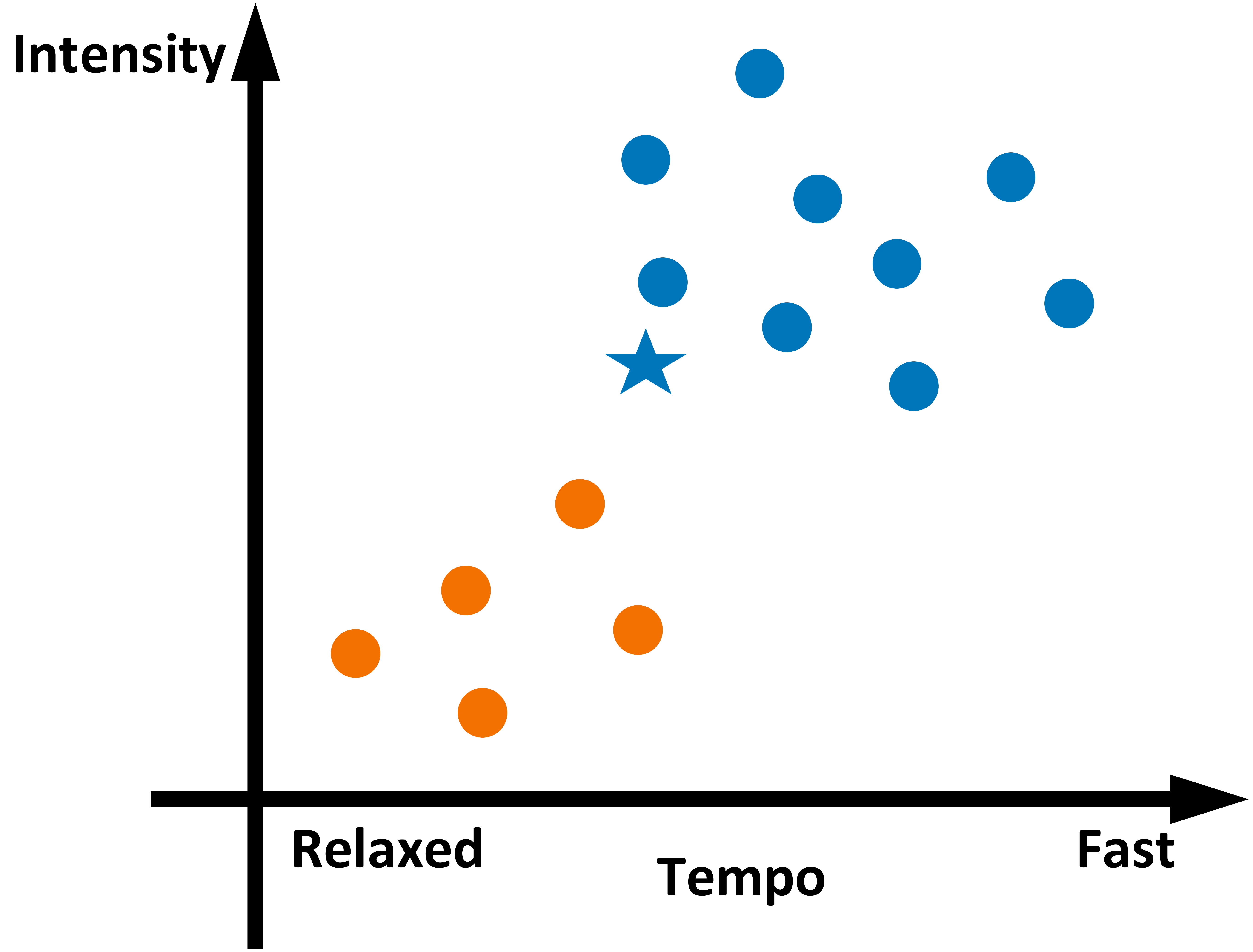
Example 1: Predict if a user likes a song or not 1-NN

User Sharon

- Dislike
- Like



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



Dislike

Like

K-nearest neighbors for classification

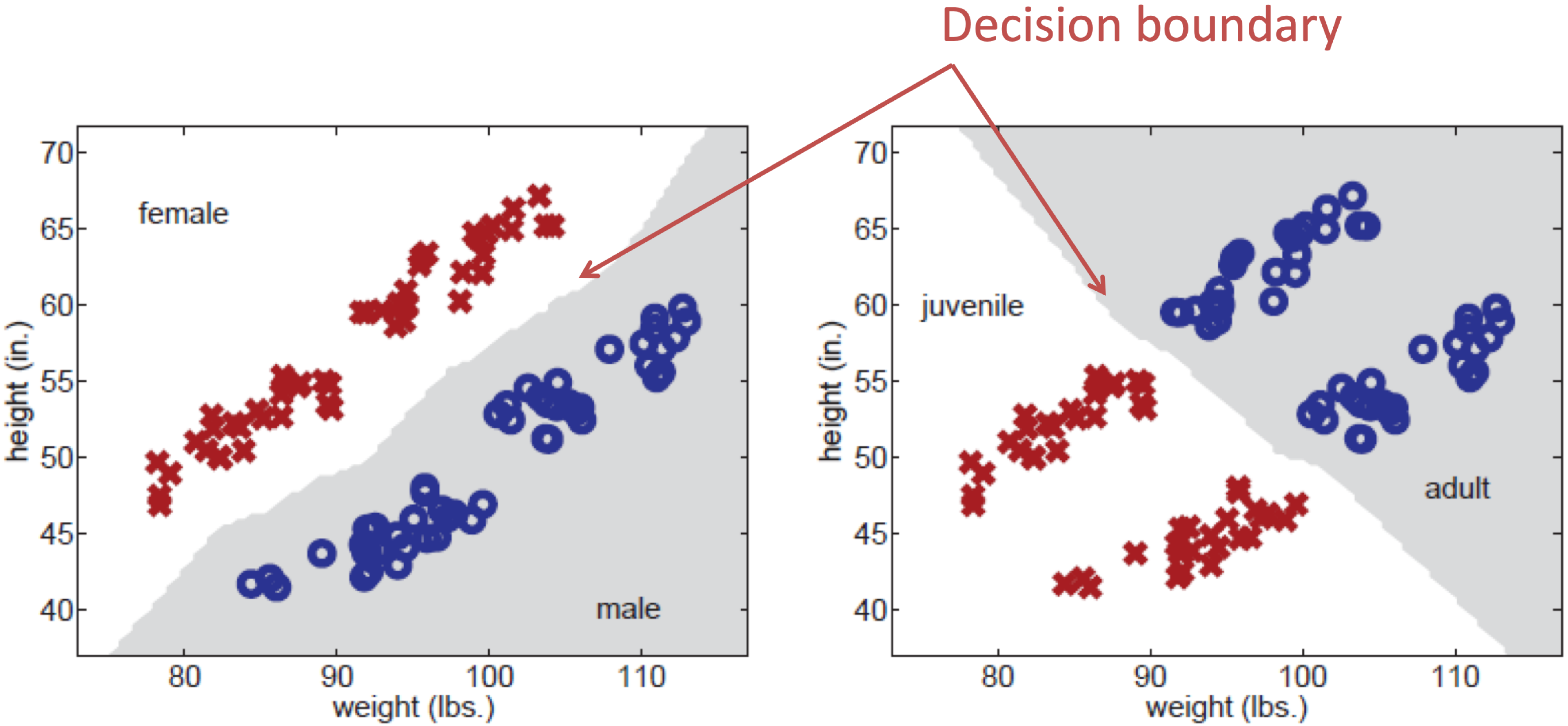
- **Input**: **Training data** $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

  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

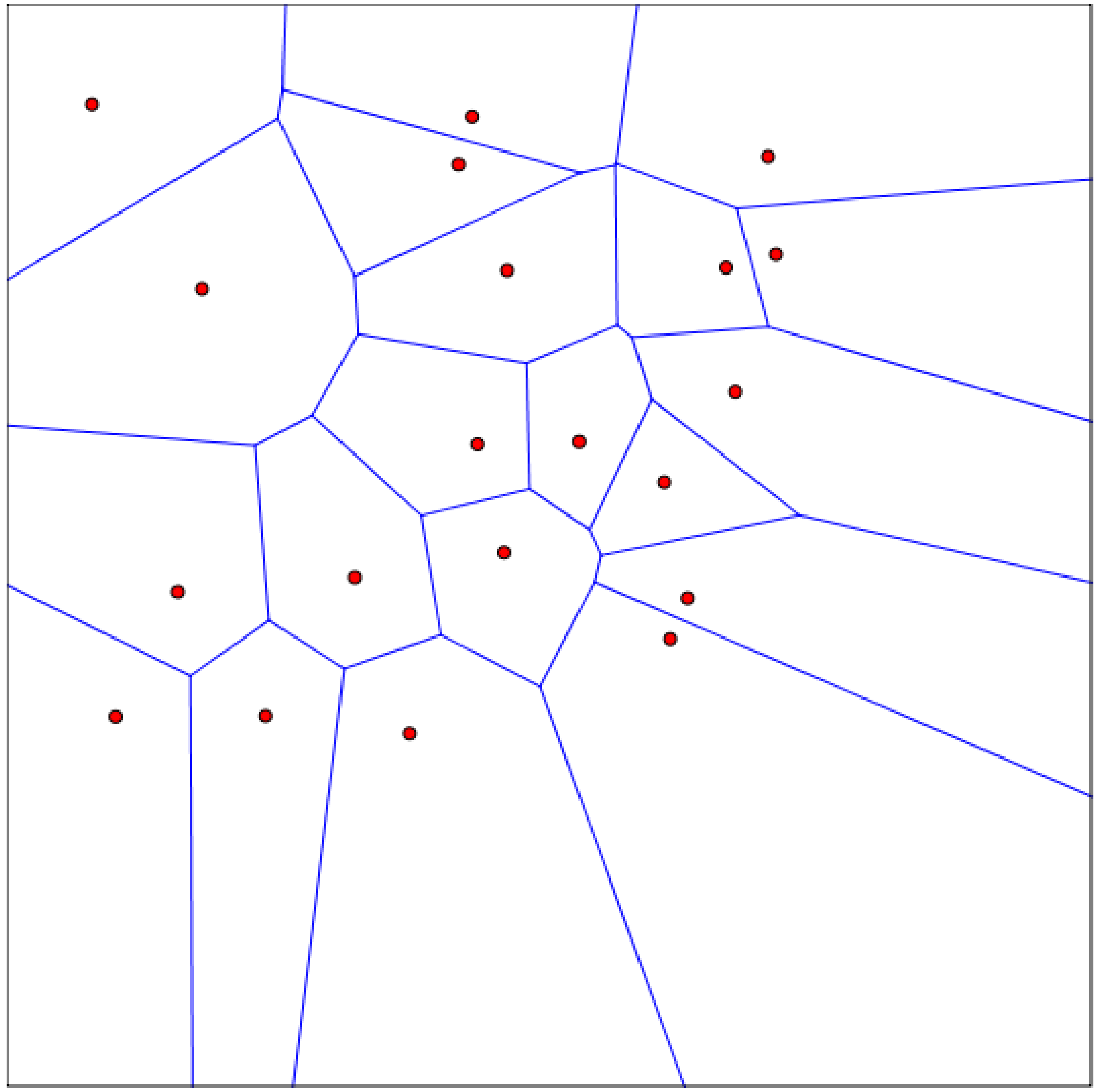


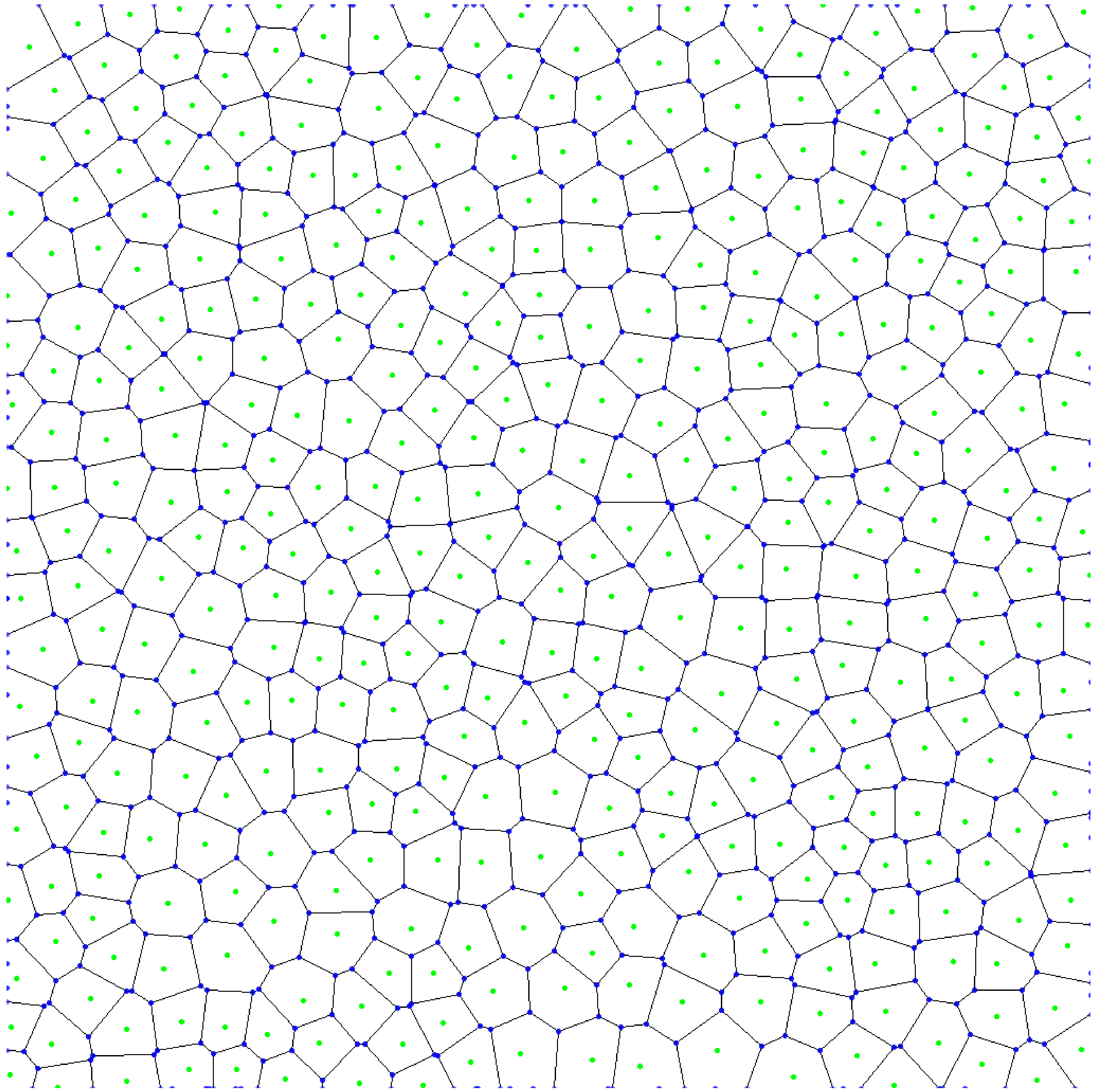
(a) classification by gender (b) classification by age

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

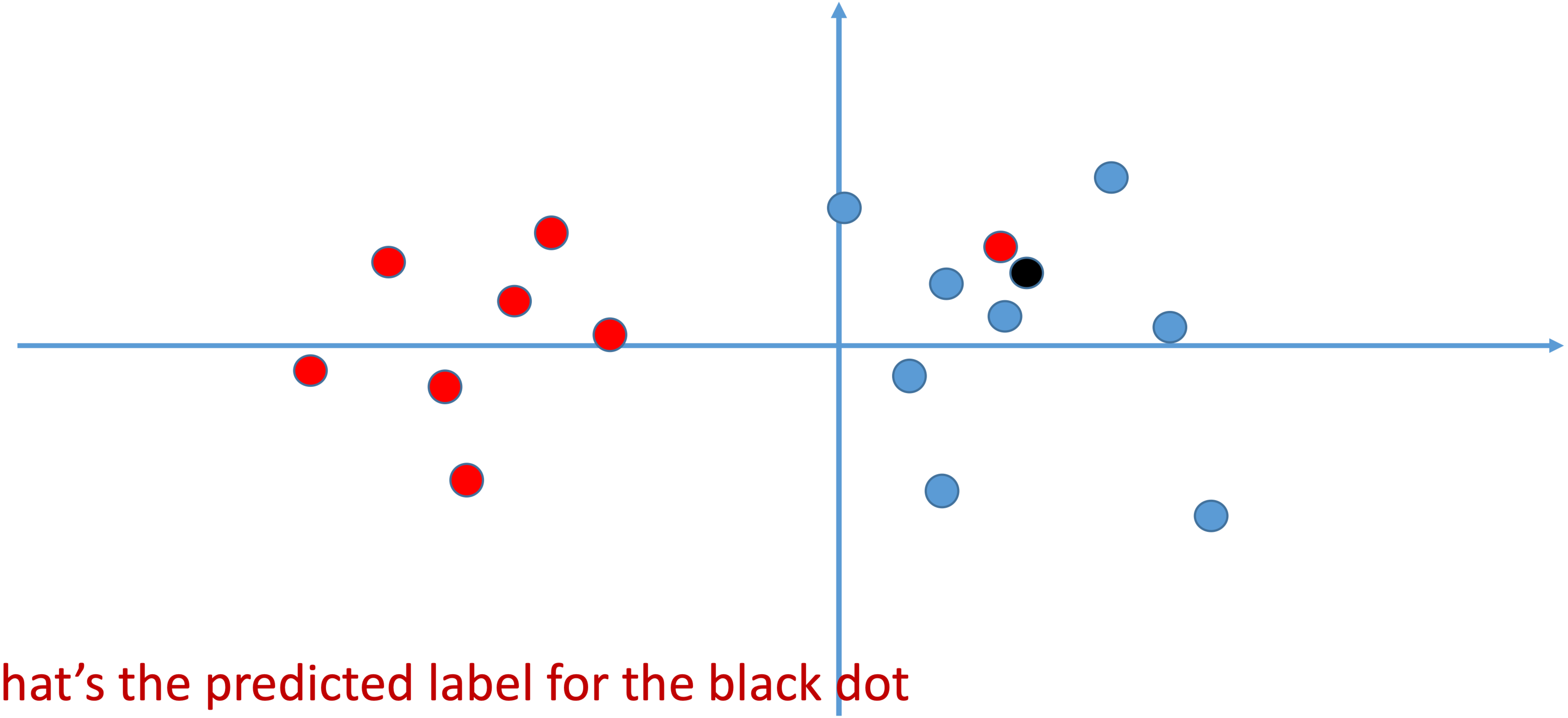
Training set mean for dimension d

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

Training set standard deviation for dimension d

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

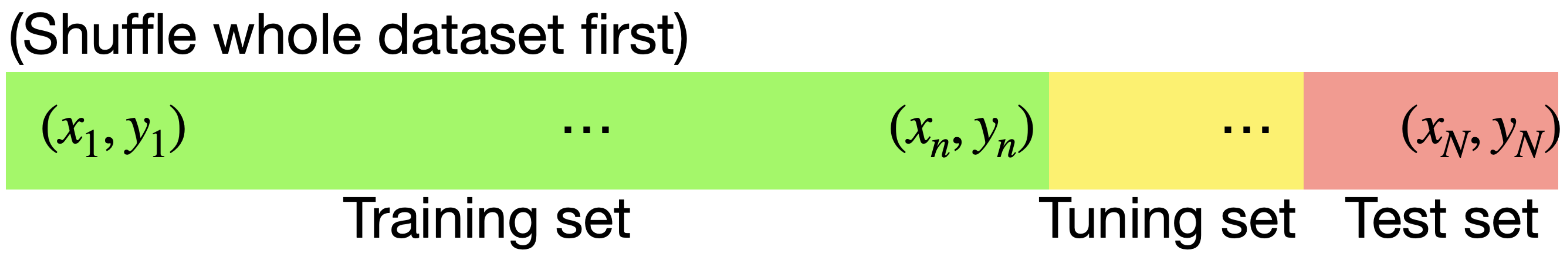
Effect of k



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/validation sets**
- Classify tuning set with different k
- Pick k that produces least tuning-set error



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 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 $\mathcal{F}$
that **best describe the data observed**

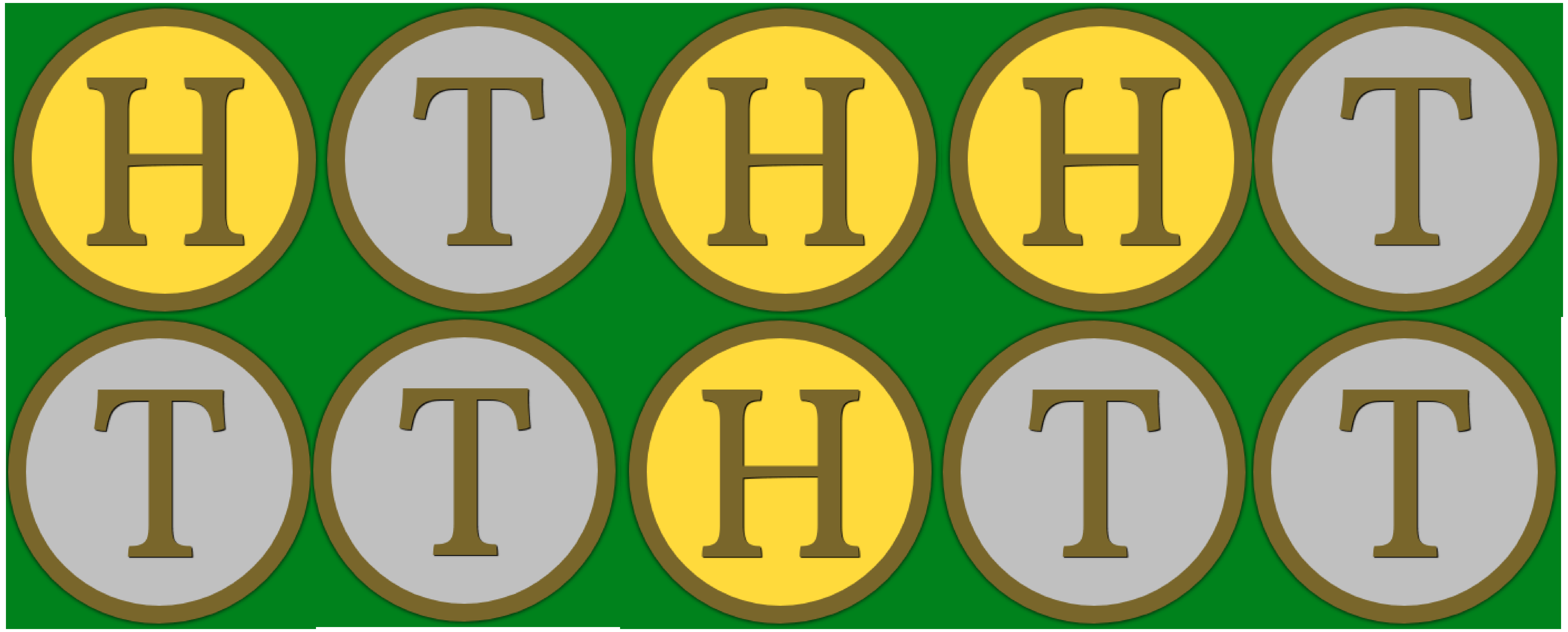
How to select $\hat{\boldsymbol{f}} \in \boldsymbol{\mathcal{F}}$?

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

Under i.i.d assumption

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

How good is θ?

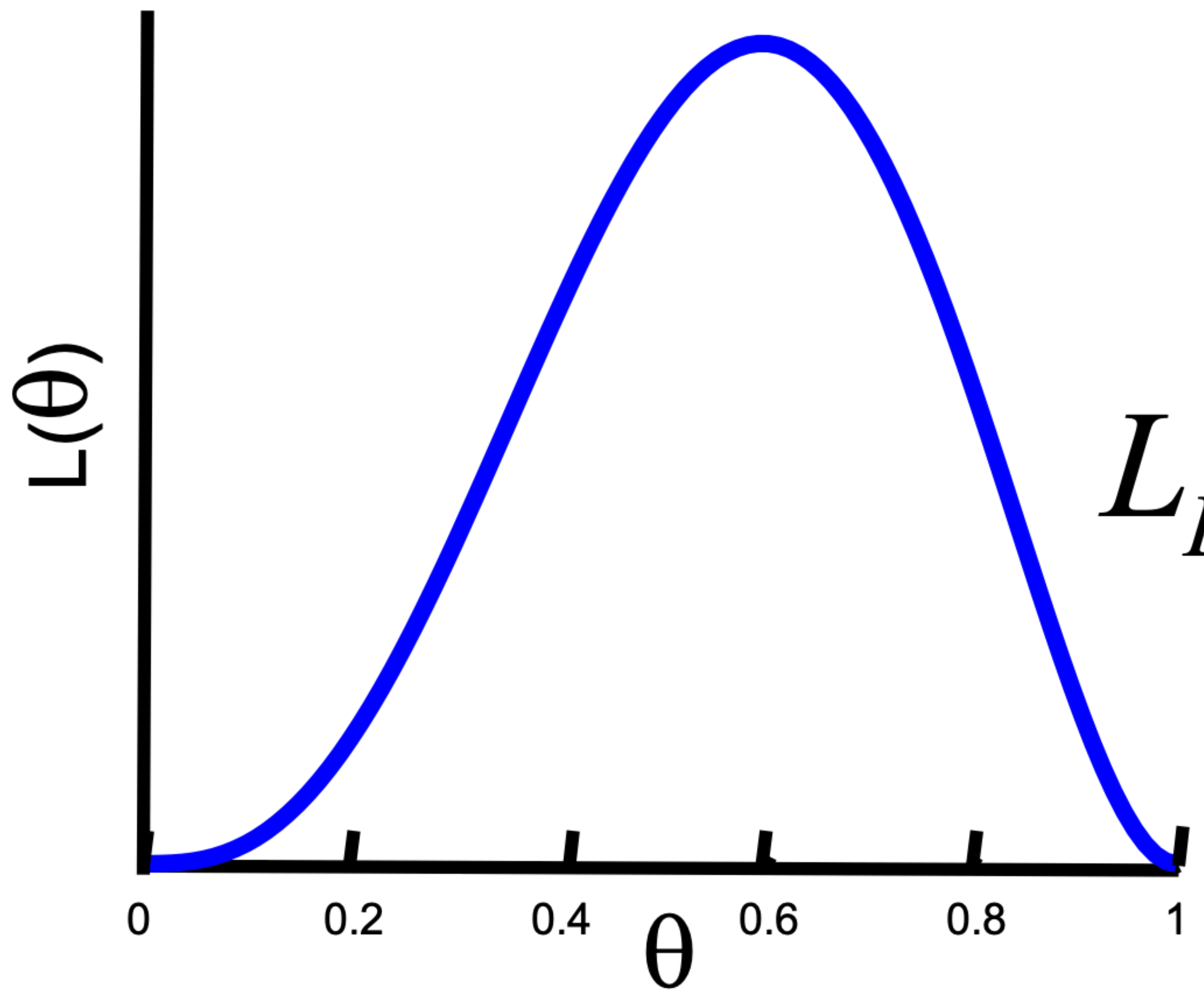
It depends on how likely it is to generate the observed data

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

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

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

which confirms your intuition!

Connecting MLE and Loss Minimization

- MLE solves

$$\operatorname*{argmax}_{\theta} p(x_1, \dots, x_n \mid \theta) = \operatorname*{argmax}_{\theta} \prod_{i=1}^{n} p(x_i \mid \theta)$$

- Rewrite the problem in an equivalent form

$$\operatorname*{argmax}_{\theta} p(x_1, \dots, x_n \mid \theta) = \operatorname*{argmin}_{\theta}(-\log p(x_1, \dots, x_n \mid \theta))$$

$$= \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} -\log p(x_i \mid \theta)$$

Connecting MLE and Loss Minimization

- We call "$-\log p(x_i \mid \theta)$" the **negative log likelihood**
- May define $\ell(\theta; x_i) := -\log p(x_i \mid \theta)$
- Maximum likelihood estimation is **loss minimization.** Different notation, same computation.

$$\underset{\theta}{\operatorname{argmax}}\, p(\, x_1, \dots, x_n \mid \theta\,) = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \ell(\theta; x_i)$$

Lecture stopped here on 10/1/25

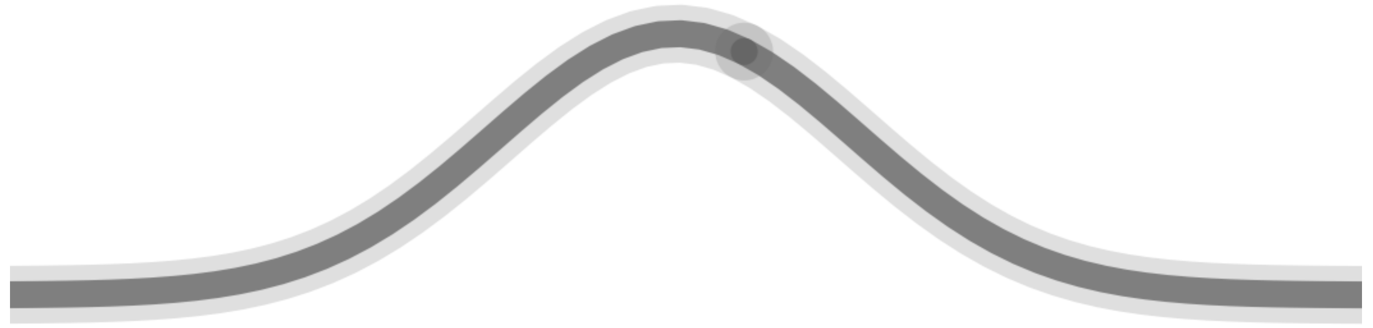
Maximum Likelihood Estimation: Gaussian Model

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**

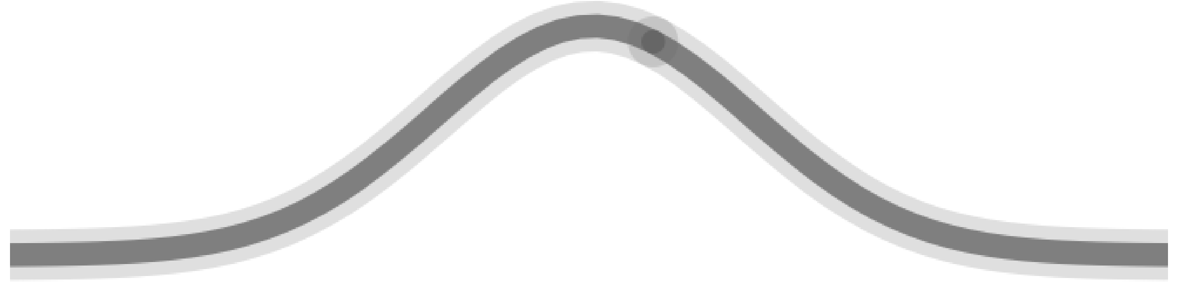
$$\sigma^2 = \mathbf{E}[(x-\mu)^2] \quad \text{hence} \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

Why?

Maximum Likelihood Estimation: Gaussian Model

Observe some data (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

$$\underset{\mu,\sigma^2}{\operatorname{argmax}} \prod_{i=1}^{n} p(x_i; \mu, \sigma^2) = \underset{\mu,\sigma^2}{\operatorname{argmin}} - \log \prod_{i=1}^{n} p(x_i; \mu, \sigma^2)$$

- **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

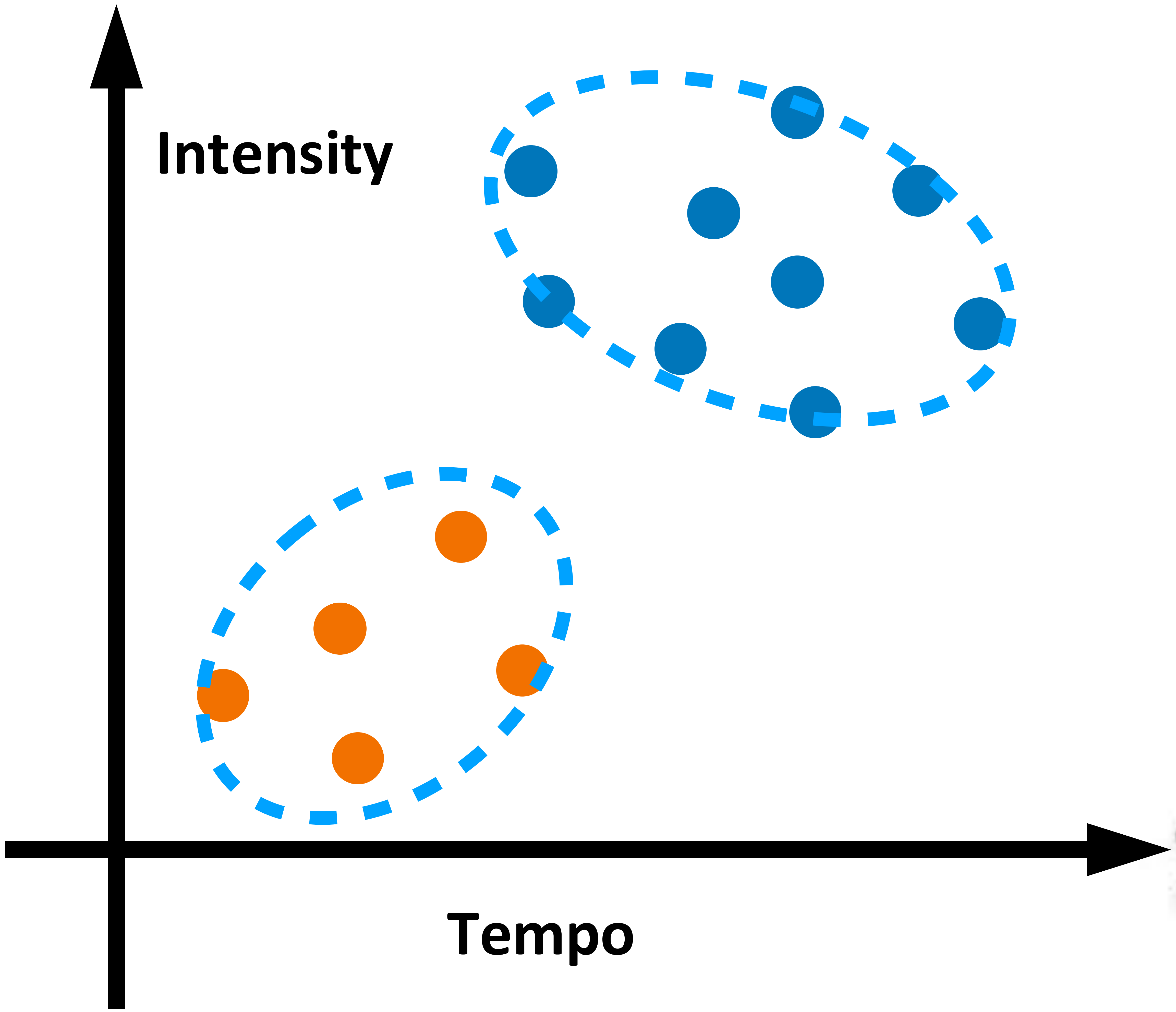
$$\frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

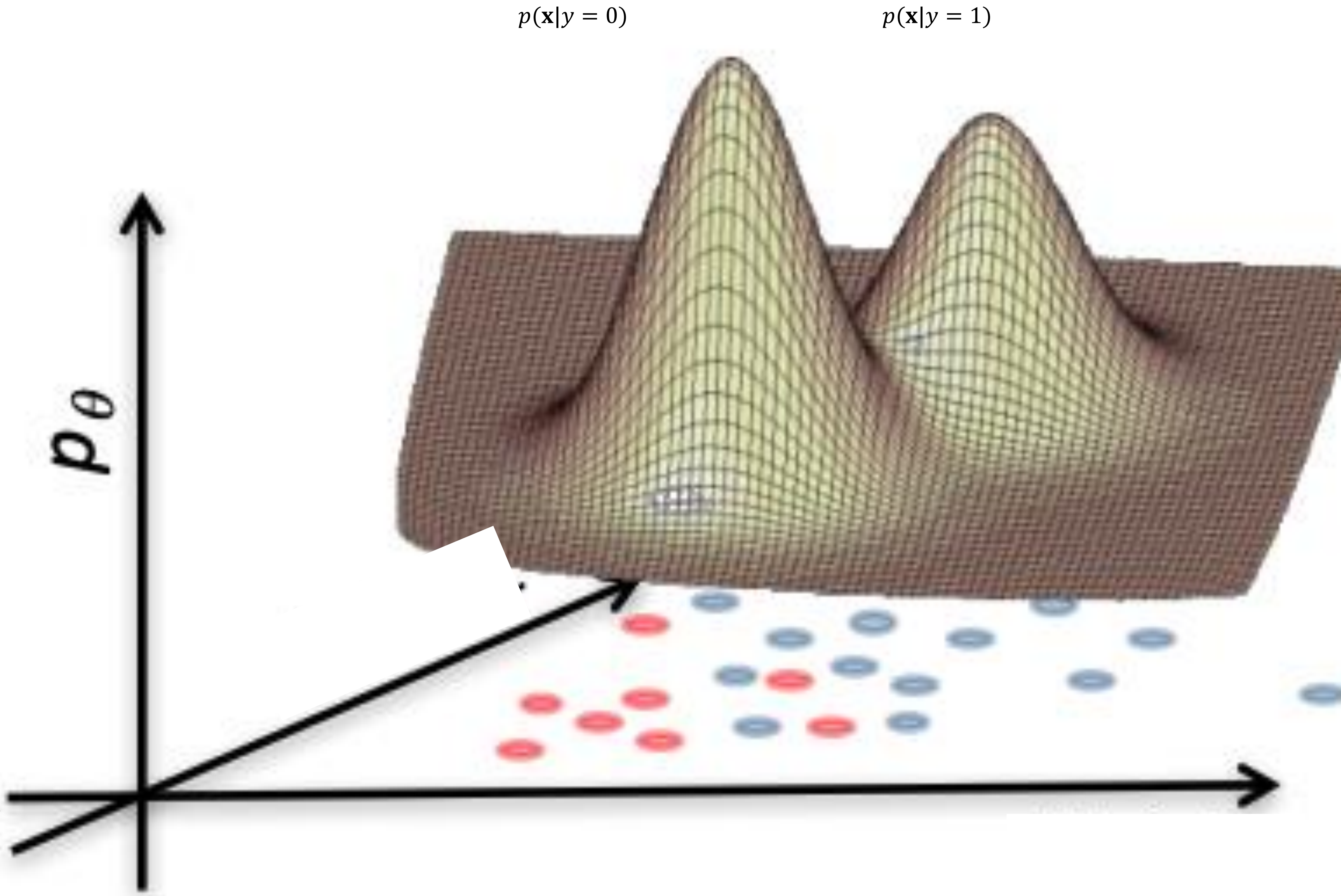
- Take derivatives with respect to it

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

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

Classification via MLE





Classification via MLE

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

(Posterior)

(Prediction)

Classification via MLE

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

(Posterior)

(Prediction)

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

(by Bayes' rule)

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

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

Part III: Naïve Bayes

Example 1: Play outside or not?

- If weather is sunny, will my 2-year-old daughter want to play outside?

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

Example 1: Play outside or not?

- If weather is sunny, will my 2-year-old daughter want to play outside?

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

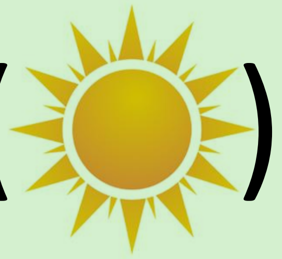
- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

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(Play | ) = p( | Play) p(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 $\mathbf{x} = \{X_1, \ldots, X_k\}$

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

(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 $\mathbf{x}$ 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)

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

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

Class prior

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!)

Thanks!