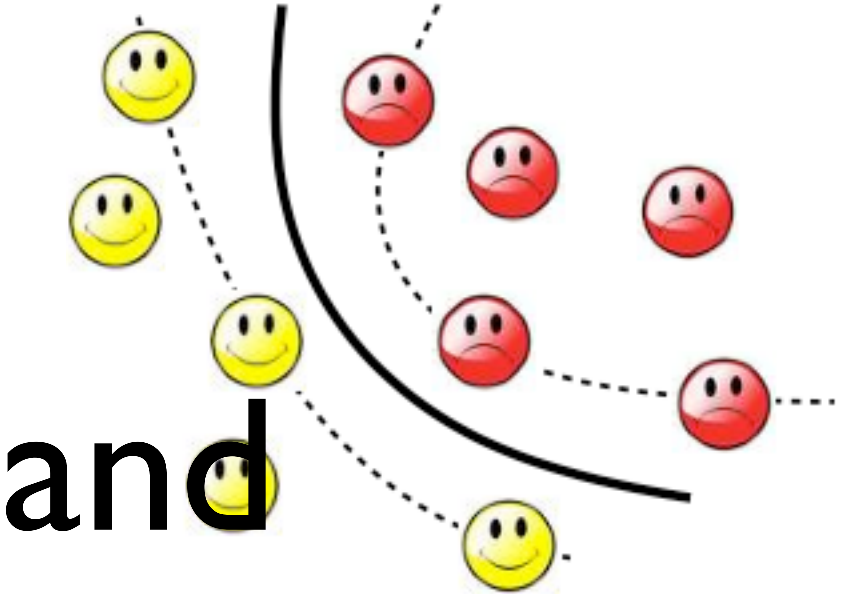
THE UNIVERSITY OF SYDNEY



Machine Learning and Data Mining

Review

Nguyen Hoang Tran

Principal component

- Given a data matrix $X \in \mathbb{R}^{n \times d}$, the principle components of X are the eigenvectors of $X^\top X$

- Principal component analysis (PCA) for X is to find the eigenvectors and eigenvalues of the matrix $X^\top X$.

Eigen Decomposition

- If A is symmetric, all its e-vals are real, and all its e-vecs are orthonormal, $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$

- Hence $U^T U = U U^T = I, \;\; |U|^2 = 1.$

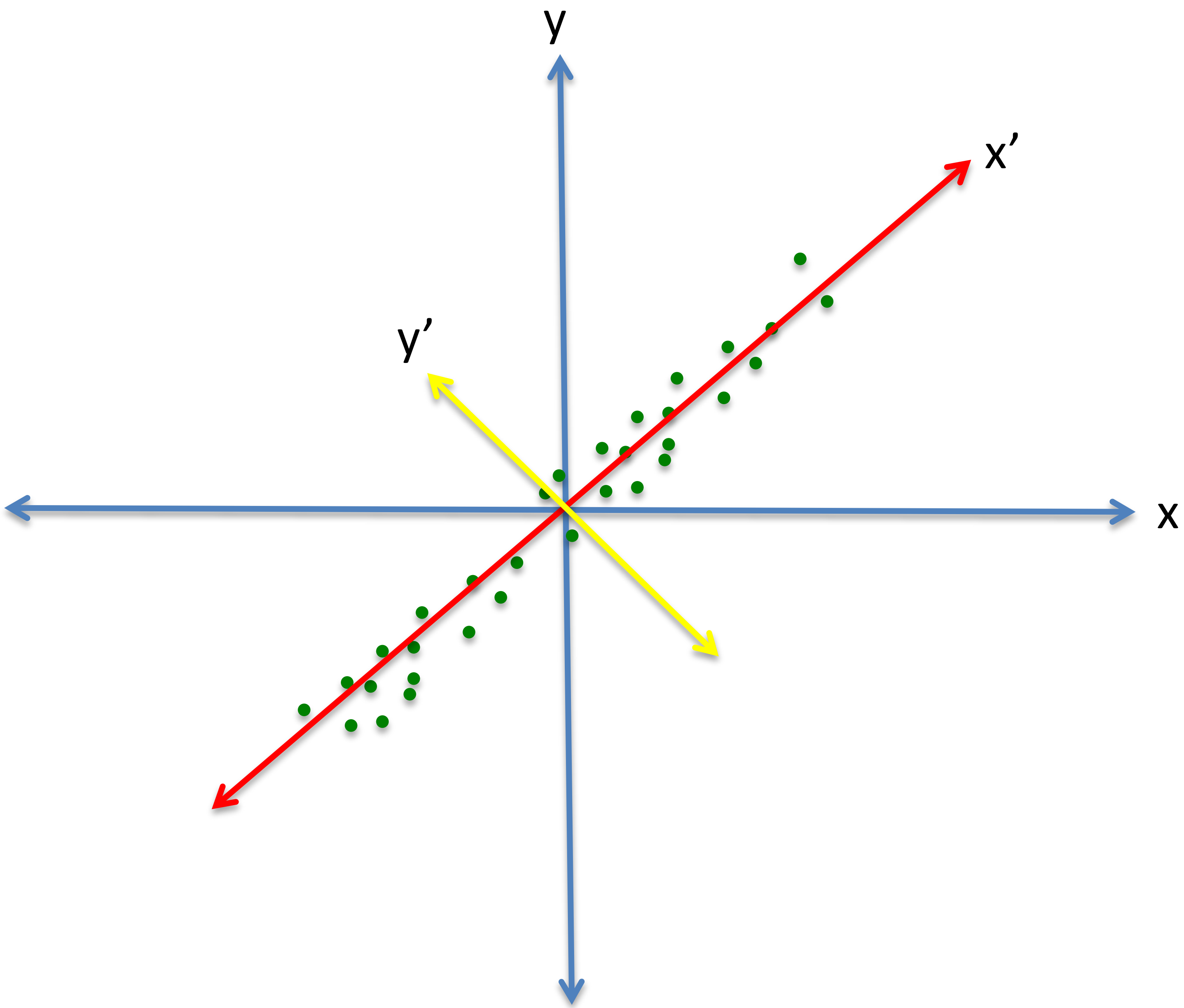
- and $A = U \Lambda U^T = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$

PCA

The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

[Jolliffe, Pricipal Component Analysis, 2nd edition]

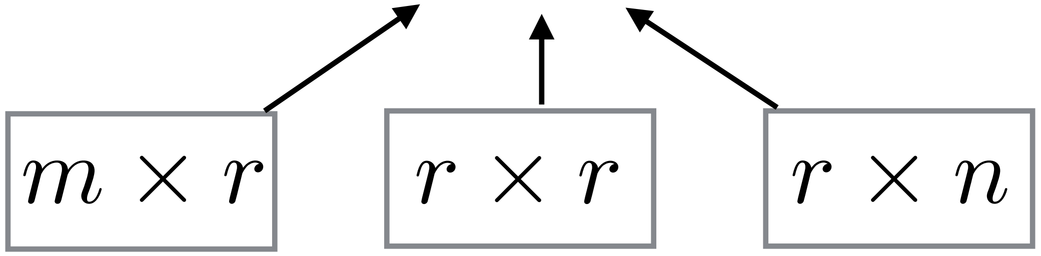
Geometric intuition



Singular Value Decomposition

- Given **any** real matrix X of size (m, n) it can be expressed as:

$$X = U\Sigma V^T$$



$m \times r$

$r \times r$

$r \times n$

- r is the rank of matrix X
- U is a (m, r) column-orthonormal matrix
- V is a (n, r) column-orthonormal matrix
- Σ is diagonal r x r matrix

Singular Value Decomposition

$$X = \underset{\boxed{m \times r}}{U}\,\underset{\boxed{r \times r}}{\Sigma}\,\underset{\boxed{r \times n}}{V^T}$$

$$X = \lambda_1 \mathbf{u}_1 \times \mathbf{v}_1^t + \lambda_2 \mathbf{u}_2 \times \mathbf{v}_2^t + \cdots + \lambda_r \mathbf{u}_r \times \mathbf{v}_r^t$$

Compression of X

- To get better compression we should look at the λ values.
  - These are called **singular values**.
- We can arrange them in descending order:

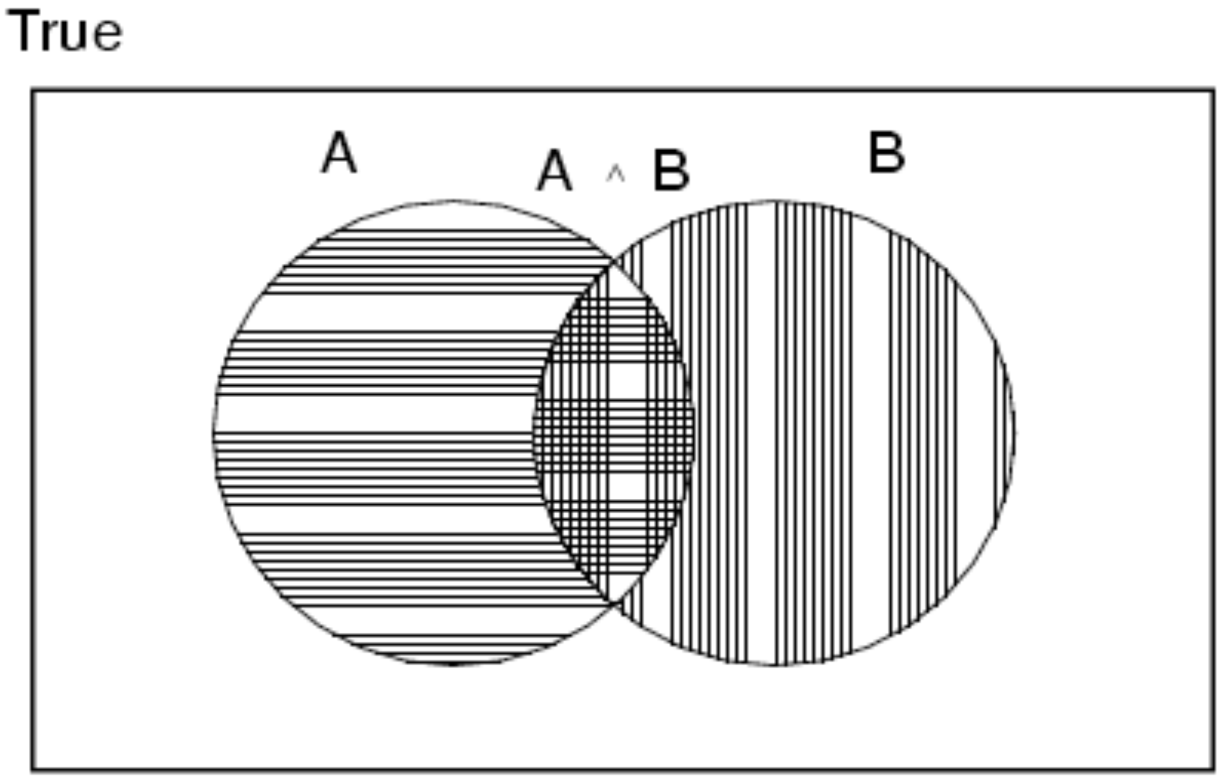
$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r > 0$$

- Now recall….

$$X = \lambda_1 \mathbf{u}_1 \times \mathbf{v}_1^t + \lambda_2 \mathbf{u}_2 \times \mathbf{v}_2^t + \cdots + \lambda_r \mathbf{u}_r \times \mathbf{v}_r^t$$

$$\hat{X} = \sum_{i=1}^{k} \lambda_i \mathbf{u}_i \times \mathbf{v}_i^t$$

Axioms of probability



$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cap B) \equiv P(AB) \equiv P(A, B)$$

$$P(A^c) = 1 - P(A)$$

Rules of Probability

- Sum Rule $P(X) = \sum_{Y} P(X,Y)$
- Product Rule $P(X,Y) = P(Y|X)P(X)$

Conditional Probabilities

- One of the most important concepts in all of Data Mining and Machine Learning

- P(A|B) = P(A,B)/P(B) …assuming P(B) not equal 0.

– Conditional probability of A given B has occurred.

- Probability it will rain tomorrow given it has rained today.

– P(A|B) = P(AB)/P(B) = 0.1/0.4 = 1⁄4 = 0.25

– In general P(A|B) is not equal to P(B|A)

Bayes’ rule

- $P(A|B) = P(AB)/P(B)$; $P(B|A) = P(BA)|P(A)$
- Now $P(AB) = P(BA)$
- Thus $P(A|B)P(B) = P(B|A)P(A)$
- Thus $P(A|B) = [P(B|A)P(A)] / [P(B)]$

– This is called Bayes Rule
– Basis of almost all prediction
– Latest theories hypothesise that human memory and action is Bayes' rule in action.

Bayes' Rule

Likelihood

Posterior

Prior

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Normaliser

$$P(hypothesis|data) = \frac{P(data|hypothesis)P(hypothesis)}{P(data)}$$

Example

The ASX market goes up 60% of the days of a year. 40% of the time it stays the same or goes down. The day the ASX is up, there is a 50% chance that the Shanghai Index is up. On other days there is 30% chance that Shanghai goes up. Suppose the Shanghai market is up. What is the probability that ASX was up?

- Define A1 as "ASX is up"; A2 is "ASX is not up"
  Define S1 as "Shanghai is up"; S2 is "Shanghai is not up"
- We want to calculate $P(A1 \mid S1)$?
- $P(A1) = 0.6$; $P(A2) = 0.4$;
  $P(S1 \mid A1) = 0.5$; $P(S1|A2) = 0.3$
- $P(S2 \mid A1) = 1 - P(S1 \mid A1) = 0.5$;
  $P(S2 \mid A2) = 1 - P(S1 \mid A2) = 0.7$;

Example cont.

- We want to calculate $P(A1 \mid S1)$?
- $P(A1) = 0.6$; $P(A2) = 0.4$;
  $P(S1 \mid A1) = 0.5$; $P(S1 \mid A2) = 0.3$
  $P(S2 \mid A1) = 1 - P(S1 \mid A1) = 0.5$;
  $P(S2 \mid A2) = 1 - P(S1 \mid A2) = 0.7$;
- $P(A1 \mid S1) = P(S1 \mid A1)P(A1) / (P(S1))$
- How do we calculate $P(S1)$?

Example cont.

- $P(S1) = P(S1,A1) + P(S1,A2)$ [Key Step]

  $= P(S1 \mid A1)P(A1) + P(S1 \mid A2)P(A2)$

  $= 0.5 \text{ x } 0.6 + 0.3 \text{ x } 0.4$

  $= 0.42$

- Finally,

  $P(A1 \mid S1) = P(S1 \mid A1)P(A1) / P(S1)$

  $= (0.5 \text{ x } 0.6)/0.42$

  $= 0.71$

Loss functions

- Squared error, $0\text{-}1$ Loss

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$L(y, \hat{y}) = I(y \neq \hat{y})$$

- Minimise risk, (expected risk, empirical risk)

$$R(\hat{f}) = E_{\mathbf{x},y} L(f(\mathbf{x}), \hat{f}(\mathbf{x}))$$

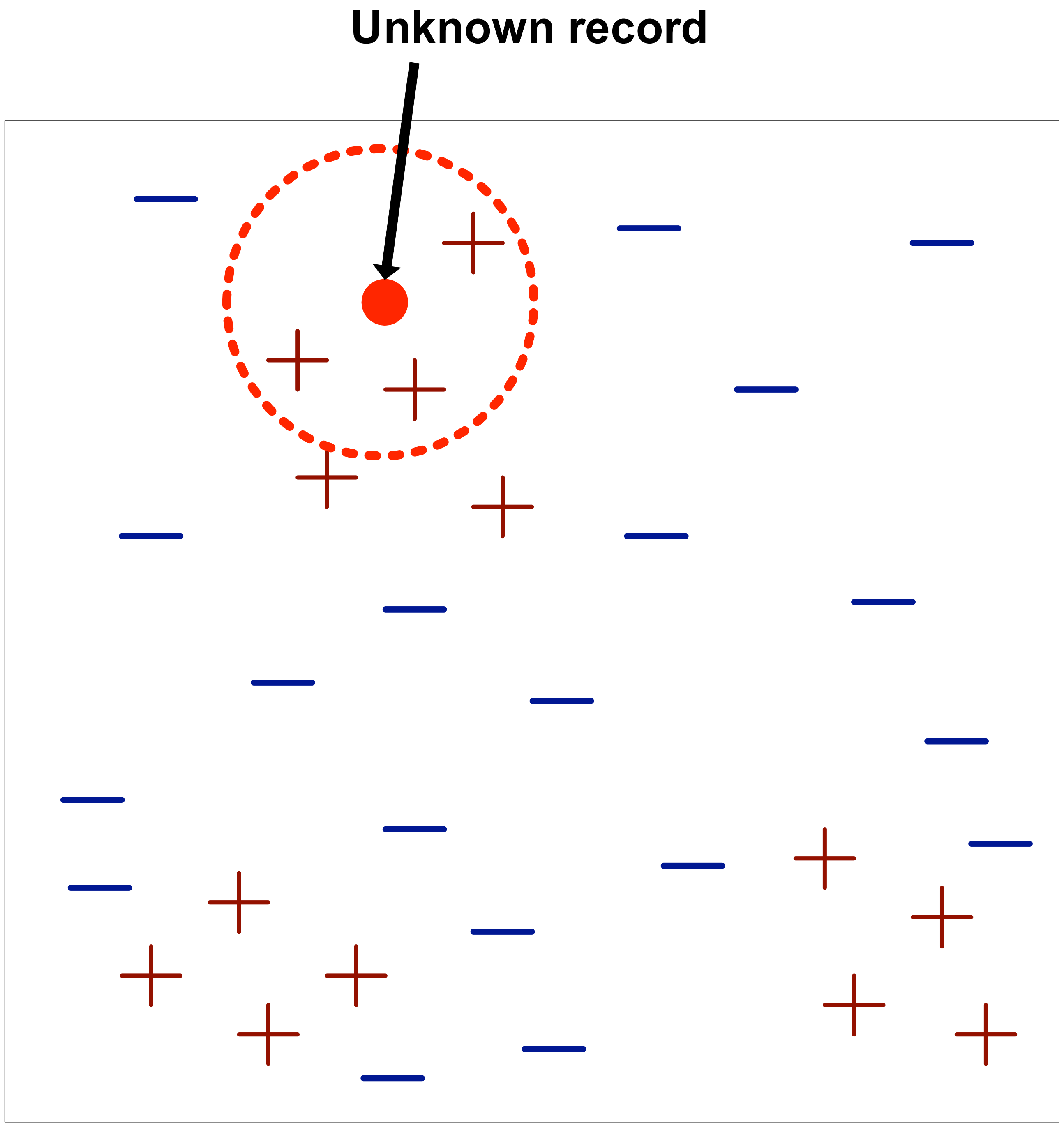
$$\hat{R}(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}(\mathbf{x}_i))$$

Generative vs Discriminative

- Generative approach:
  - Model $p(y, \mathbf{x}) = p(\mathbf{x}|y)p(y)$
  - Use Bayes' theorem $p(y|\mathbf{x}) = \dfrac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$

- Discriminative approach:
  - Model $p(y|\mathbf{x})$ directly

Nearest Neighbour Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbours to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbours
  - Use class labels of nearest neighbours to determine the class label of unknown record (e.g., by taking majority vote)

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes' theorem

$$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

  - Choose value of C that maximises
    $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximises
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$
- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C) = P(A_1| C_j)\, P(A_2| C_j) \ldots P(A_n| C_j)$
  - Can estimate $P(A_i| C_j)$ for all A_i and C_j.
  - New point is classified to C_j if $P(C_j) \prod P(A_i| C_j)$ is maximal.

How to Estimate Probabilities from Data?

#	Refund	Status	Salary	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$
  - e.g., $P(No)$ = 7/10,
    $P(Yes)$ = 3/10
- For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}|/ N_{c^k}$$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
  - Examples:
    $P(Status{=}Married|No)$ = 4/7
    $P(Refund{=}Yes|Yes)$ = 0

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

$P(A|M)P(M) > P(A|N)P(N)$

=> Mammals

Logistic Regression

- Assumes a parametric form for directly estimating P(Y | X). For binary concepts, this is:

$$P(Y=0|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

$$P(Y=1|X) = 1 - P(Y=0|X)$$

$$= \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

- Equivalent to a one-layer backpropagation neural net.
- Logistic regression is the source of the sigmoid function used in backpropagation.
- Objective function for training is somewhat different.

Logistic Regression Objective

- Equivalently viewed as maximising the **conditional log likelihood** (CLL)

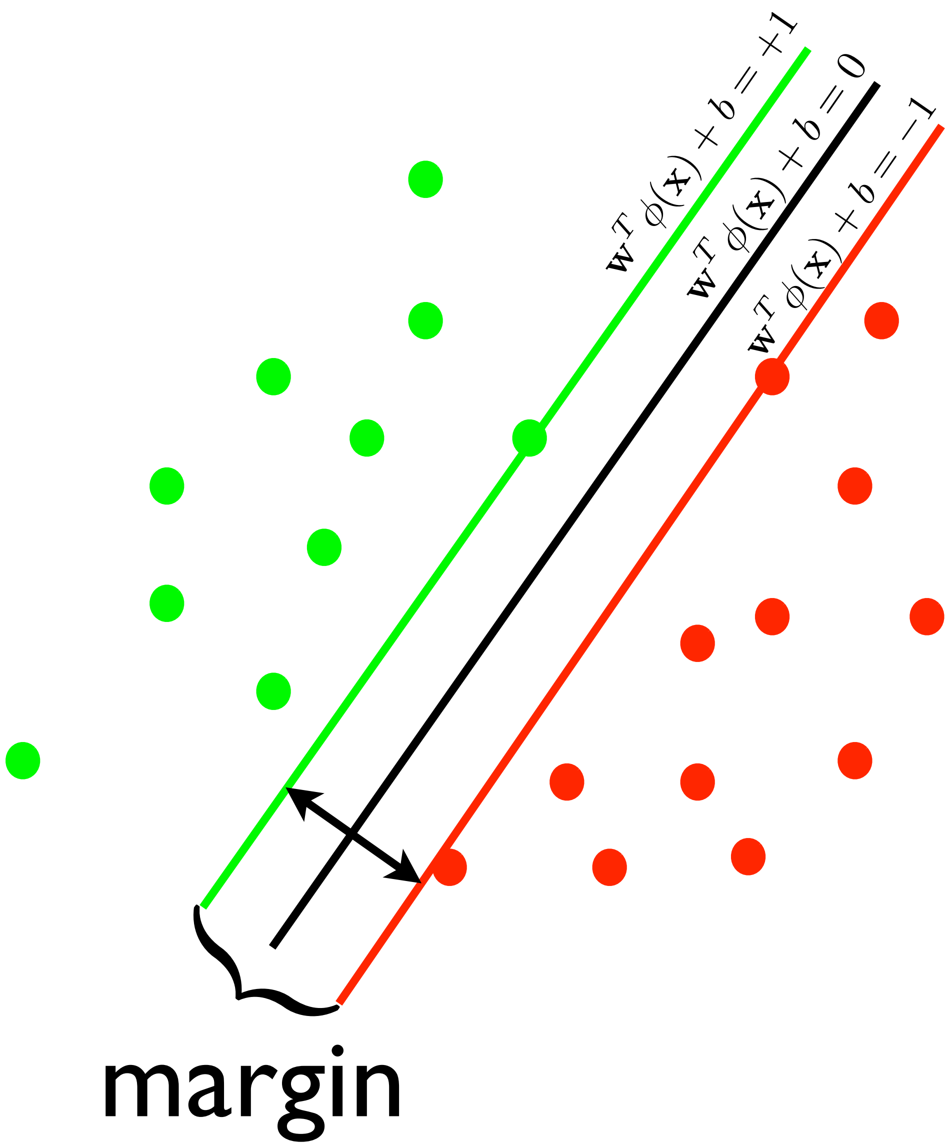
$$W \leftarrow \underset{W}{\operatorname{argmax}} \sum_{d \in D} \ln P(Y^d \mid X^d, W)$$

- The objective function

$$\min_W -\frac{1}{|D|} \sum_{d \in D} \left(Y^d \ln \left(\frac{\exp(W^\top X^d)}{1 + \exp(W^\top X^d)} \right) + (1 - Y^d) \ln \left(\frac{1}{1 + \exp(W^\top X^d)} \right) \right)$$

$$= \min_W \frac{1}{|D|} \sum_{d \in D} \ln \left(1 + \exp(W^\top X^d)\right) - Y^d W^\top X^d$$

Support Vector Machines



$\mathbf{w}^T\phi(\mathbf{x})+b=+1$
$\mathbf{w}^T\phi(\mathbf{x})+b=0$
$\mathbf{w}^T\phi(\mathbf{x})+b=-1$
margin

Quadratic programming

$$\arg\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2$$

s.t.

$$t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right)\geqslant 1,\;\; n=1,\ldots,N$$

- Solve efficiently by quadratic programming (QP)
- Hyperplane defined by support vectors

Clustering

Dataset $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ with N Observations

Each data point is D dimension

Goal: Partition dataset into K clusters. (For now, assume K is given)

$\boldsymbol{\mu}_k = (\mu_1, ..., \mu_D)$ prototype for each cluster $k \in 1, ..., K$

Binary indicator variables

$$r_{nk} = \begin{cases} 1, & \text{if datapoint } n \text{ belongs to cluster } k \\ 0, & \sim \end{cases}$$

If $\mathbf{x}_n$ is assigned to cluster k, then $r_{nk} = 1 \ \wedge \ r_{nj} = 0 \ \forall j \neq k$

K-Means

Objective function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Represents the sum of the squares of the distances of each datapoint to its assigned prototype vector.

Goal: Find $\{\boldsymbol{\mu}_k\}$ and $\{r_{nk}\}$ that minimise J.

$$\{r_{nk}, \boldsymbol{\mu}_k\}^\star = \underset{\{r_{nk}, \boldsymbol{\mu}_k\}}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

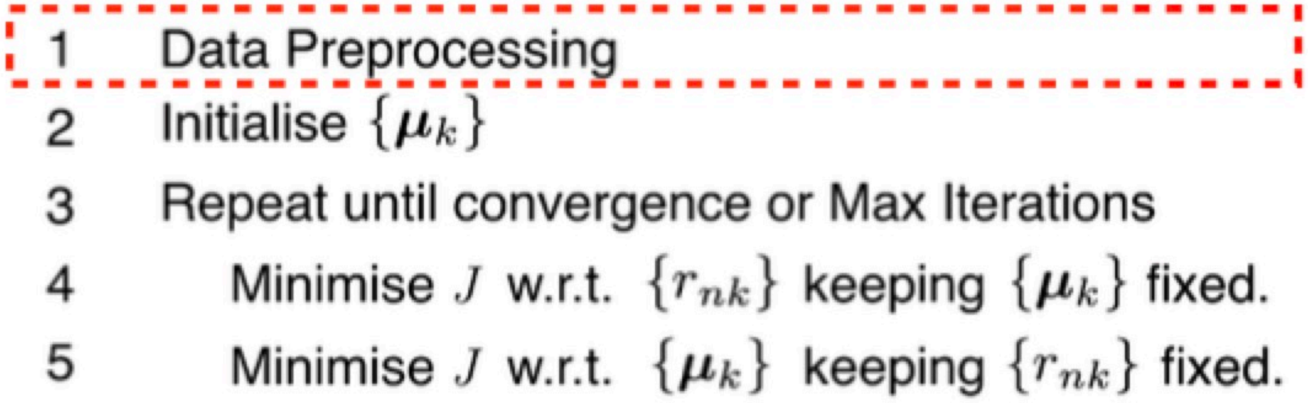
K-Means

Iterative solution to minimise J:

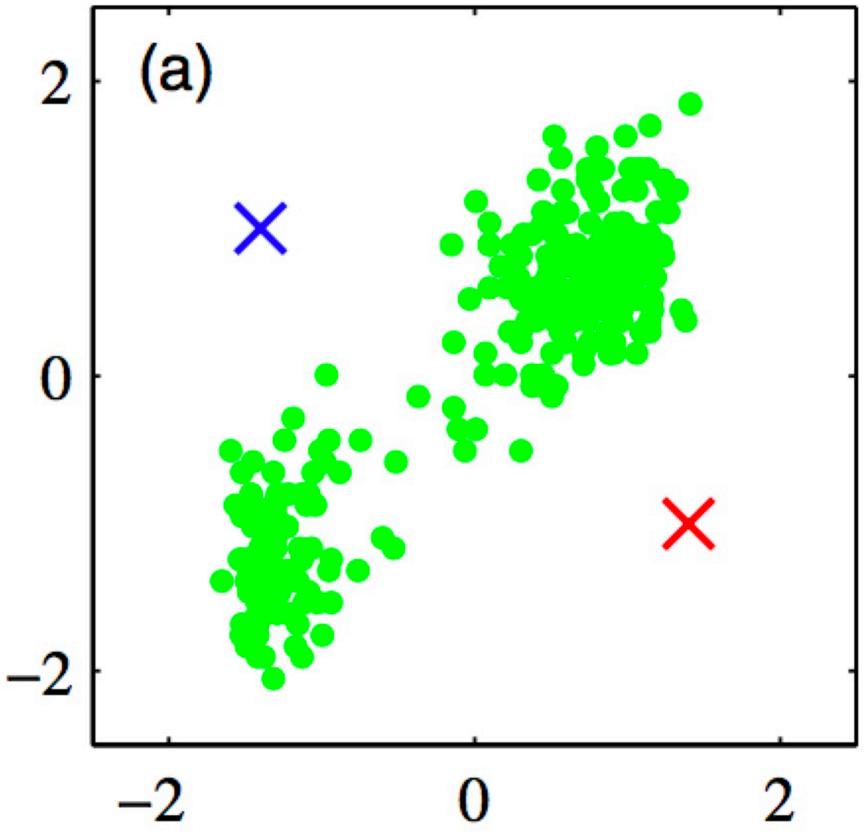
1. Data Preprocessing
2. Initialise $\{\boldsymbol{\mu}_k\}$
3. Repeat 4 and 5 until convergence or Max Iterations
4. Minimise J w.r.t. $\{r_{nk}\}$ keeping $\{\boldsymbol{\mu}_k\}$
5. Minimise J w.r.t. $\{\boldsymbol{\mu}_k\}$ keeping $\{r_{nk}\}$

K-Means Example

Number of clusters: $K = 2$



```
Data Preprocessing
Initialise $\{\mu_k\}$
Repeat until convergence or Max Iterations
    Minimise $J$ w.r.t. $\{r_{nk}\}$ keeping $\{\mu_k\}$ fixed.
    Minimise $J$ w.r.t. $\{\mu_k\}$ keeping $\{r_{nk}\}$ fixed.
```

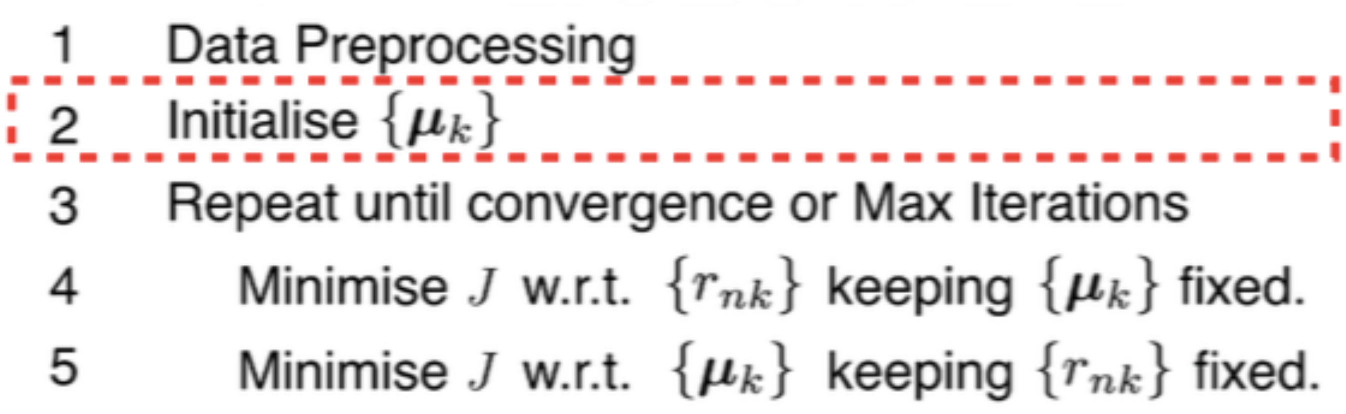


Each dimension has zero mean and unit standard deviation.

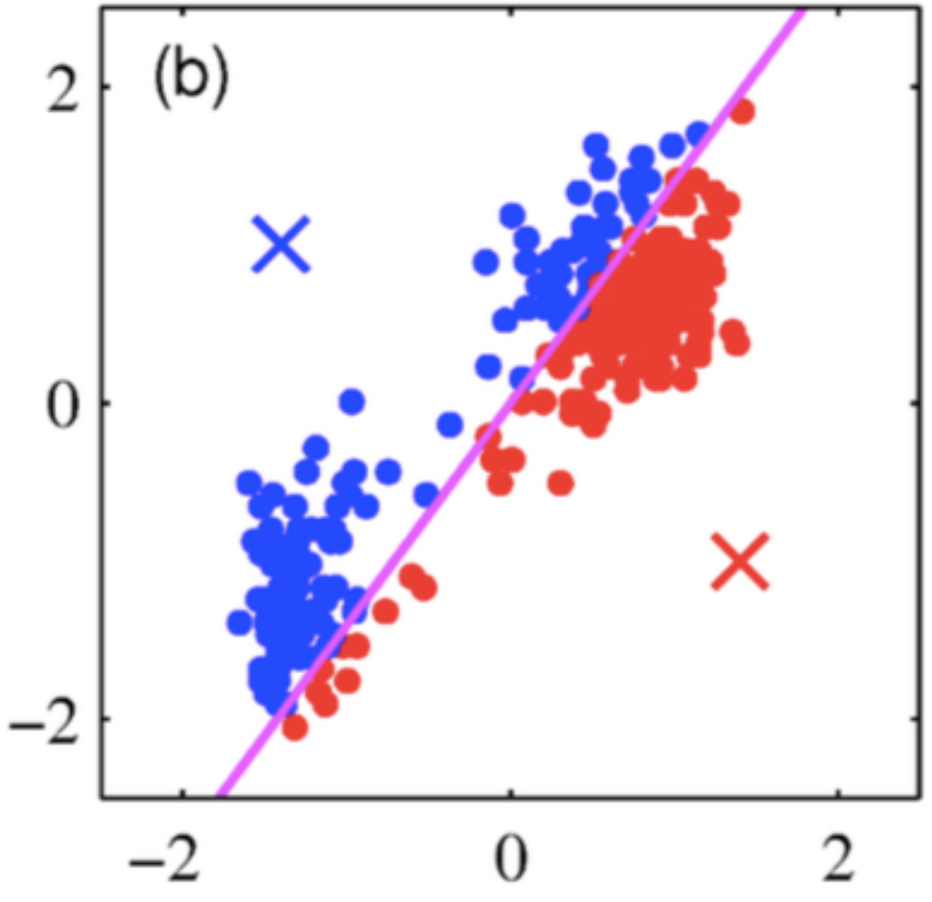
Better initialisation: Choose$\{\mu_k\}$ as average of a random subset.

K-Means Example

Number of clusters: $K = 2$



```
Data Preprocessing
Initialise $\{\boldsymbol{\mu}_k\}$
Repeat until convergence or Max Iterations
    Minimise $J$ w.r.t. $\{r_{nk}\}$ keeping $\{\boldsymbol{\mu}_k\}$ fixed.
    Minimise $J$ w.r.t. $\{\boldsymbol{\mu}_k\}$ keeping $\{r_{nk}\}$ fixed.
```

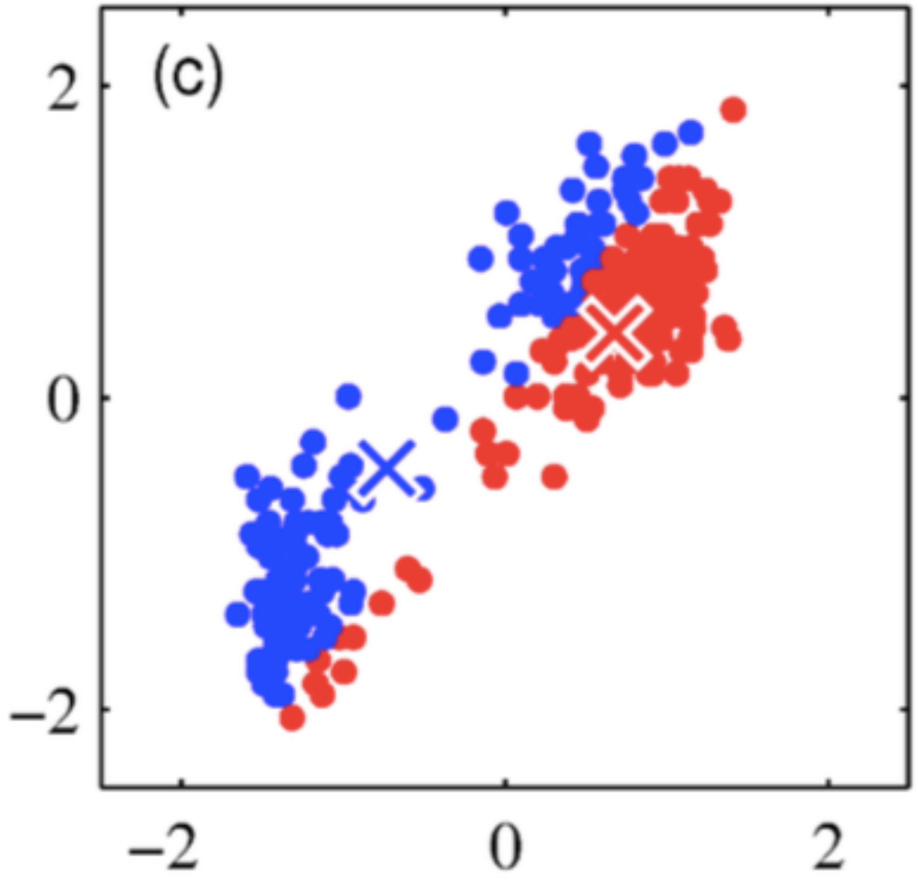


Each data point is assigned to the closest cluster centre.

K-Means Example

Number of clusters: $K = 2$

```
1   Data Preprocessing
2   Initialise $\{\boldsymbol{\mu}_k\}$
3   Repeat until convergence or Max Iterations
4       Minimise $J$ w.r.t. $\{r_{nk}\}$ keeping $\{\boldsymbol{\mu}_k\}$ fixed.
5       Minimise $J$ w.r.t. $\{\boldsymbol{\mu}_k\}$ keeping $\{r_{nk}\}$ fixed.
```

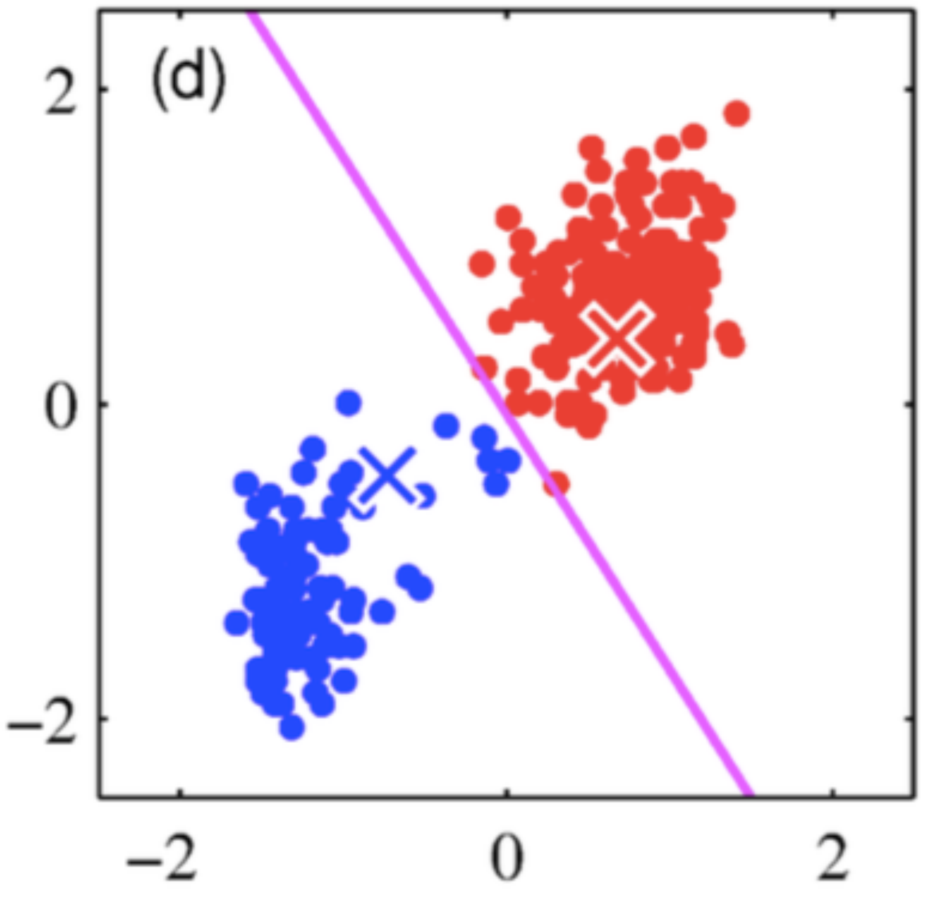


Re-compute each cluster centre to be the mean of the points previously assigned.

K-Means Example

Number of clusters: $K = 2$

```
Data Preprocessing
Initialise $\{\boldsymbol{\mu}_k\}$
Repeat until convergence or Max Iterations
    Minimise $J$ w.r.t. $\{r_{nk}\}$ keeping $\{\boldsymbol{\mu}_k\}$ fixed.
    Minimise $J$ w.r.t. $\{\boldsymbol{\mu}_k\}$ keeping $\{r_{nk}\}$ fixed.
```

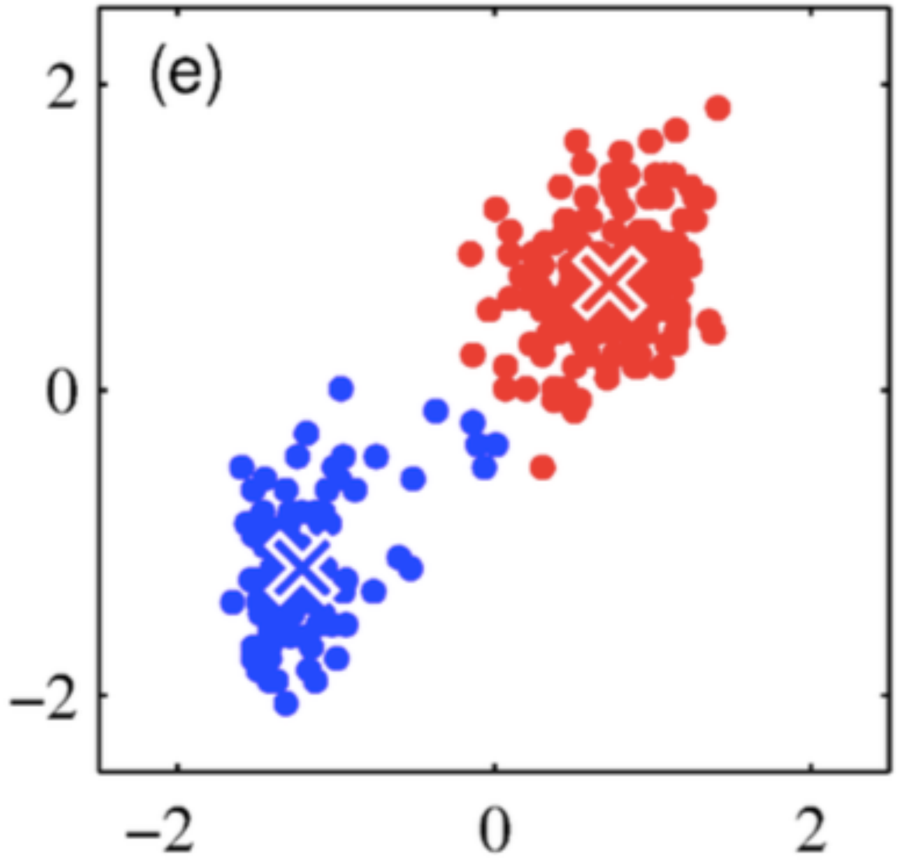


Each data point is assigned to the closest cluster centre.

K-Means Example

Number of clusters: $K = 2$

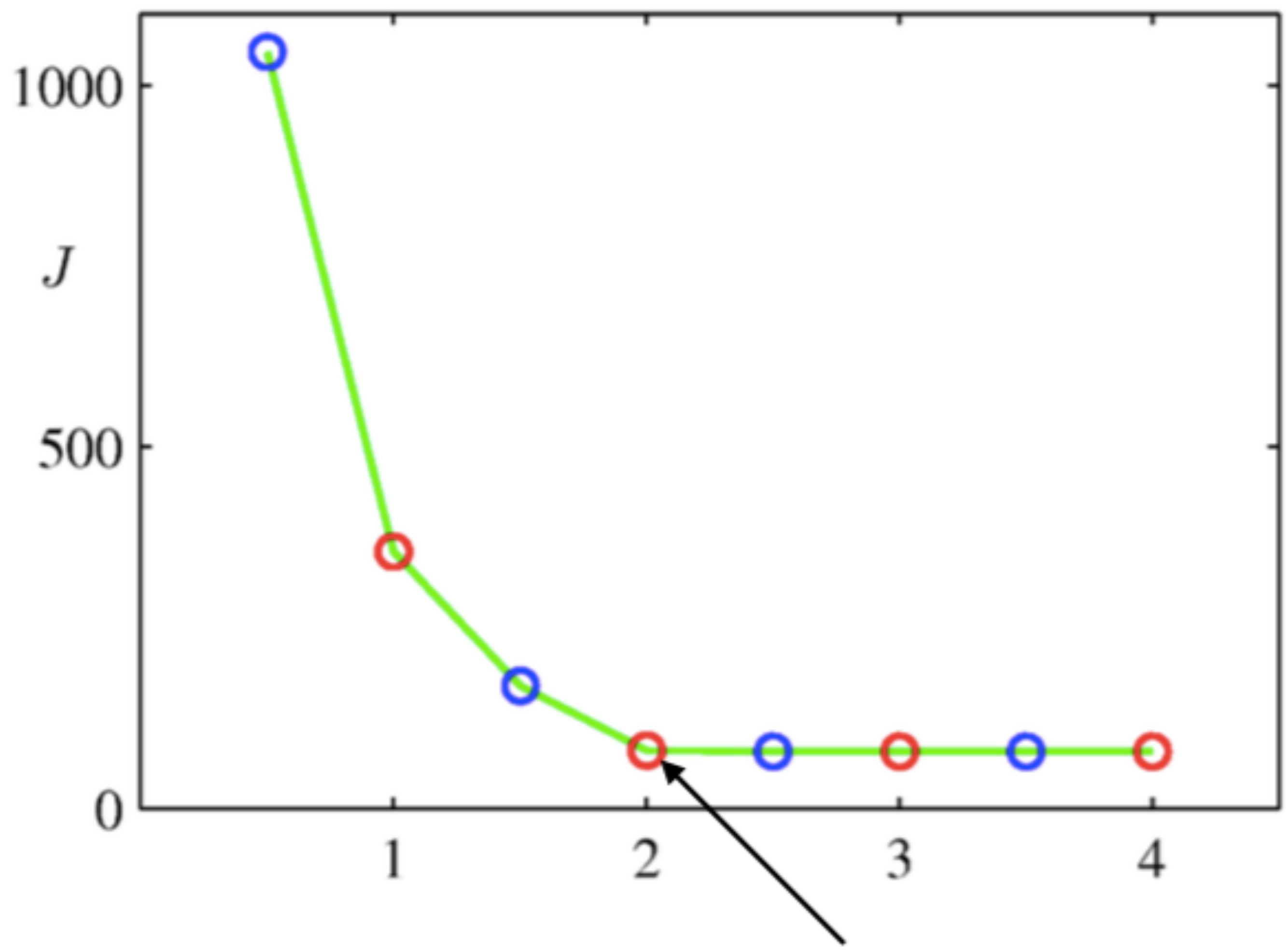
```
1   Data Preprocessing
2   Initialise $\{\boldsymbol{\mu}_k\}$
3   Repeat until convergence or Max Iterations
4       Minimise $J$ w.r.t. $\{r_{nk}\}$ keeping $\{\boldsymbol{\mu}_k\}$ fixed.
5       Minimise $J$ w.r.t. $\{\boldsymbol{\mu}_k\}$ keeping $\{r_{nk}\}$ fixed.
```



Re-compute each cluster centre to be the mean of the points previously assigned.

K-Means Example

Plot of the cost function for each iteration.



Already converged.

EM Algorithm

THE UNIVERSITY OF SYDNEY

1 Initialise means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k.

2 E-step

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3 M-step

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

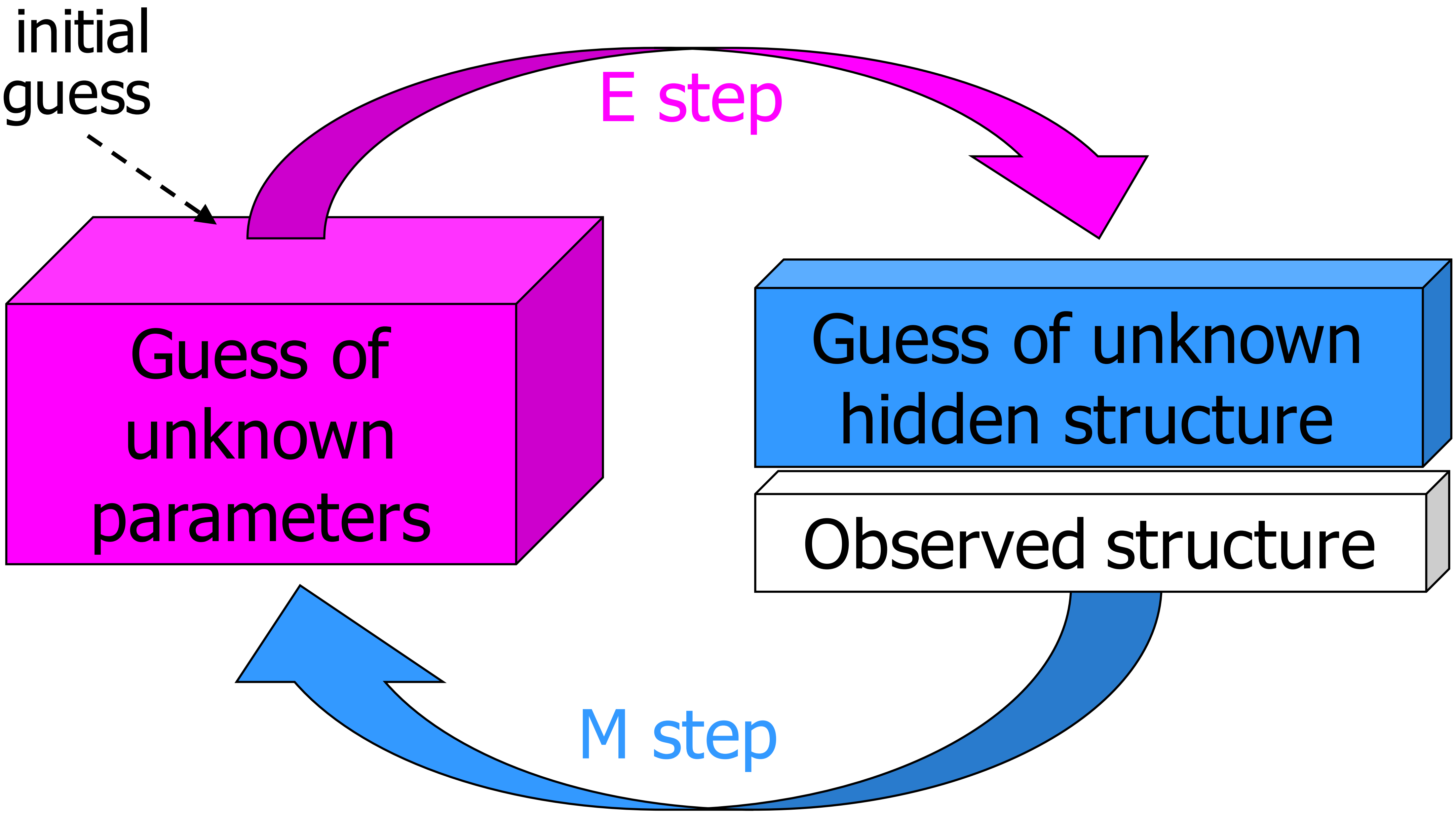
$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

4 Eval Likelihood

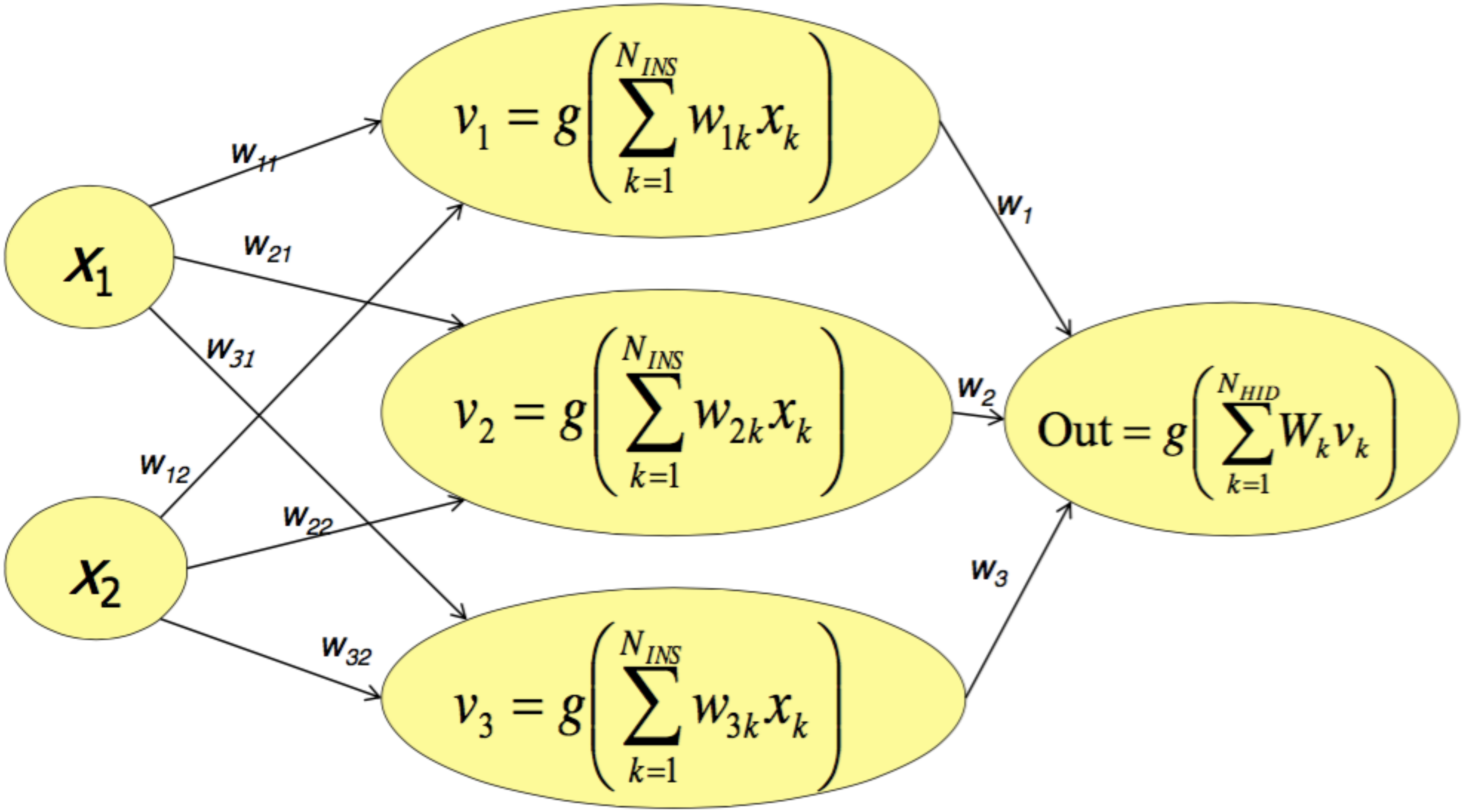
$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

EM Algorithm


initial guess
E step
Guess of unknown parameters
Guess of unknown hidden structure
Observed structure
M step

Multi-Layer Neural Networks

- There are many ways to connect perceptrons into a network. One standard way is multi-layer neural nets
- 1 Hidden layer: we can't see the output; 1 output layer


x_1
x_2
w_{11}
w_{21}
w_{31}
w_{12}
w_{22}
w_{32}
$v_1 = g\left(\sum_{k=1}^{N_{INS}} w_{1k} x_k\right)$
$v_2 = g\left(\sum_{k=1}^{N_{INS}} w_{2k} x_k\right)$
$v_3 = g\left(\sum_{k=1}^{N_{INS}} w_{3k} x_k\right)$
w_1
w_2
w_3
$\text{Out} = g\left(\sum_{k=1}^{N_{HID}} W_k v_k\right)$

Learning a neural network

- Again we will minimise the error (K outputs):

$$E(W) = \tfrac{1}{2} \sum_{i=1..N} \sum_{c=1..K} (o_{ic}-Y_{ic})^2$$

- i: the i-th training point
- o_{ic}: the c-th output for the i-th training point
- Y_{ic}: the c-th element of the i-th label indicator vector
- Our variables are all the weights w on all the edges

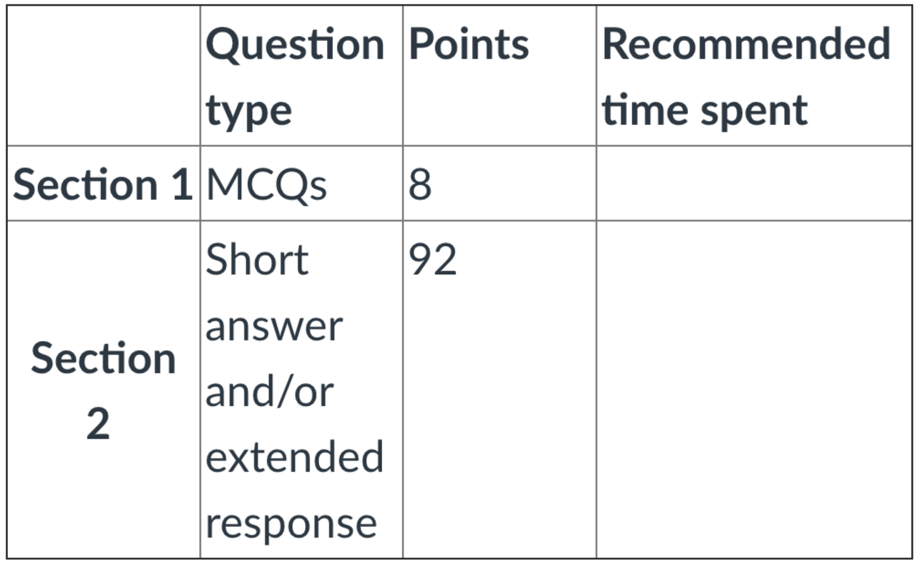
- Apparent difficulty: we don't know the 'correct' output of hidden units
- It turns out to be OK: we can still do gradient descent. The trick you need is the chain rule
- The algorithm is known as **back-propagation**

Exam Details

- Date: 4 December at 1:00 PM
- Location: Online (Canvas), Open-book
- **Duration: Your exam is 2 hours and 10 minutes long (130 minutes).** This includes 10 minutes of reading time, but you can start writing whenever you are ready– you are strongly encouraged to use this time to carefully plan and structure your response before you start writing.

Exam Details

- **Please keep track of your time.** Your timer may not update if you have an Internet connection issue. Set a timer on your watch or mobile phone so that you always know how much time you have left. Only questions completed within the exam time will be marked.
- **Format:** This exam is a Canvas Quiz with 15 questions. You should attempt all questions and follow the instructions for each question carefully.



	Question type	Points	Recommended time spent
Section 1	MCQs	8	
Section 2	Short answer and/or extended response	92	

Exam Details

- The marks assigned to the question should give you a sense of how much time to allocate
- Questions will be involve either derivations, calculations or a short discussion

Exam: what to expect?

- Numerical questions, e.g., what is probability of X … given … Y; Calculating distance between vectors; Given tiny data set, find weight vector and bias of SVM; Calculating the posterior probability of Naive Bayes classifiers; Calculating k-means update….
- Theoretical on the main concepts, e.g., SVD; gradient descent update; generative or discriminative; activation function in neural networks; ….
- Maths will be simple; no need to memorise equations
- Pros vs Cons for different methods

Advanced machine learning

If you want to learn more about machine learning, consider COMP5328: advanced machine learning, and COMP5329: Deep Learning

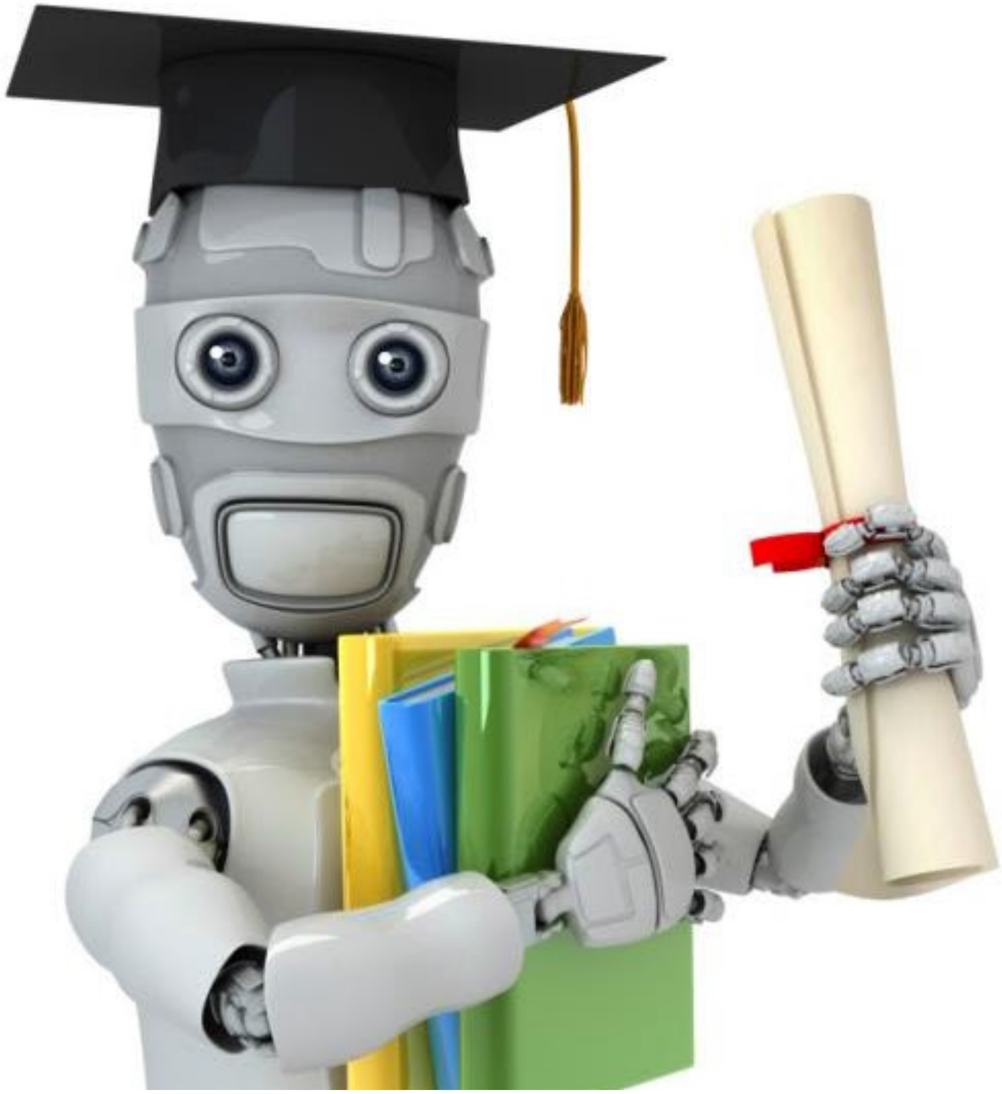
Check the questions here: https://www.analyticsvidhya.com/blog/2016/09/40-interview-questions-asked-at-startups-in-machine-learning-data-science/

- For deep learning, see these: https://www.analyticsvidhya.com/blog/2017/01/must-know-questions-deep-learning/

Unit of Study Surveys (USS)

https://student-surveys.sydney.edu.au/students/

Wish you a successful machine learning and data mining career!



Q&A