# COMP90051 **Statistical Machine Learning**

Semester 2, 2015

Revision

THE UNIVERSITY OF MELBOURNE

# About these Slides

- The aim of these slides is to provide a summary of topics covered in this subject. The slides are NOT a direct indication of the final exam.

# Week 1

- Bayes rule
- Independence
- Expectation
- Bias
- Variance
- Risk analysis (expected loss)

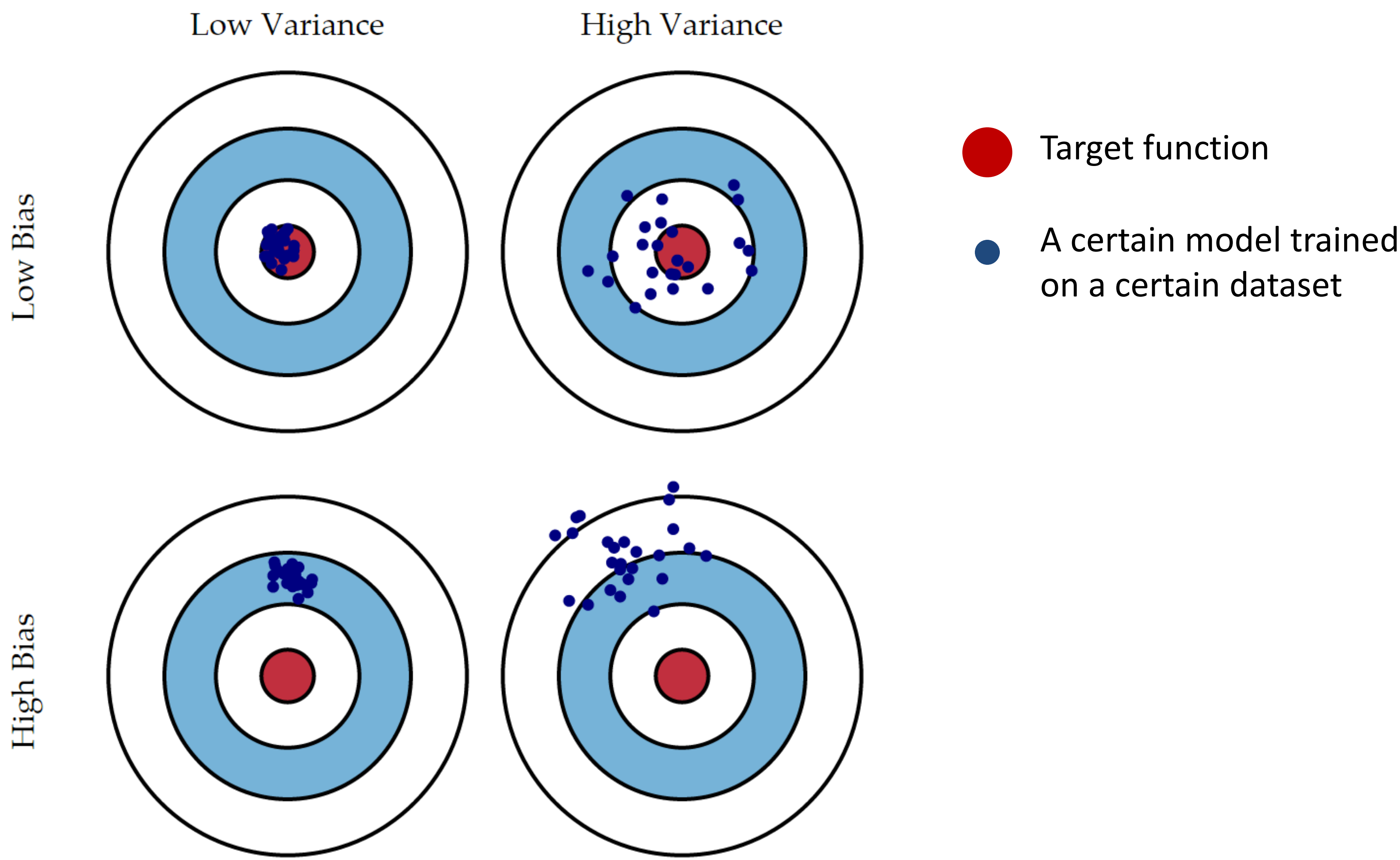
# Week 1 cont…

- Supervised vs. Unsupervised
- Parametric vs. Non-Parametric
- Generative vs. Discriminative
  - Model full joint P(X, Y)
  - Model conditional P(Y|X) only
- Frequentist vs. Bayesian

# Week 2

- Linear regression
  - Model representation: $h_w(\boldsymbol{x}) = \sum_{j=0}^{n}(w_j x_j) = \boldsymbol{w}'\boldsymbol{x}$
  - Cost function: $J(\boldsymbol{w}) = \sum_{i=1}^{m}(h_w(\boldsymbol{X_i}) - y_i)^2$
- Logistic regression
  - Model representation
    - $h_w(\boldsymbol{x}) = g(\boldsymbol{w}'\boldsymbol{x})$
    - $g(z) = \frac{1}{1+e^{-z}}$
  - Cost function
    - $y = \{0, 1\}$
    - $J(\boldsymbol{w}) = \sum_{i=1}^{m}[-y_i \log(h_w(X_i)) - (1 - y_i)\log(1 - h_w(X_i))]$

# Bias vs Variance




Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

# Week 2 cont…

- Bias vs Variance
  - Under fitting → High Bias
  - Over fitting → High Variance
- Regularization
- Linear Regression

$$J(\boldsymbol{w}) = \sum_{i=1}^{m}(h_w(X_i) - y_i)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

- Logistic Regression

$$J(\boldsymbol{w}) = \sum_{i=1}^{m}[-y_i \log(h_w(X_i)) - (1 - y_i)\log(1 - h_w(X_i))] + \lambda \sum_{j=1}^{n} w_j^2$$

# Week 3

- Ensemble Learning
  - Reduce variance: results are less dependent on peculiarities of a single training set
  - Reduce bias: a combination of multiple classifiers may learn a more expressive concept class than a single classifier
  - Generally, more diverse → more accurate
- Bagging vs Boosting

# Bagging: Resampling

- Bagging reduces variance by averaging
- Bagging has little effect on bias
  - BUT, it generally won't cause bias.
- Each base classifier is trained on less real data
- Works better with unstable classifiers

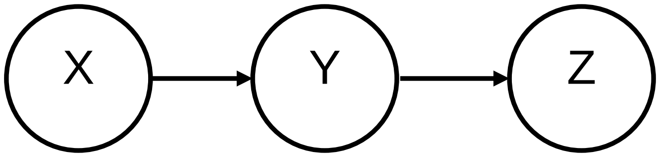
# Boosting

- Require classifiers that can handle weighted instances
  - E.g. C4.5 fractional instances
- “hard” instances have higher weights.
- In Bagging, models are built separately.
- In Boosting, models are built iteratively.

# Week 4

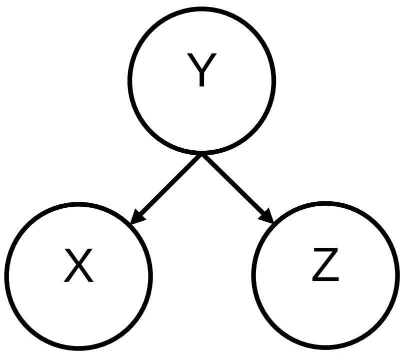
- Conditional independence
- Naïve Bayes
- PGM
  - Representation
  - CPT
  - Conditional independence in PGMs

# Examples



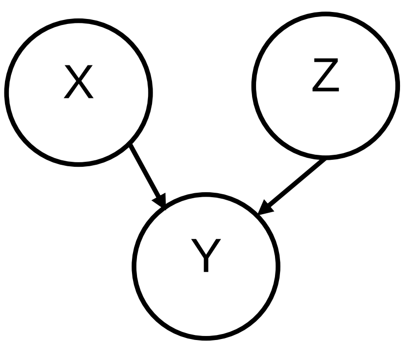

- Are $X$ and $Z$ independent? No
- Is $Z$ independent of $X$ given $Y$? Yes

$$P(Z|X,Y) = P(Z|Y)$$




Common cause

- Are $X$ and $Z$ independent? No
- Are they conditionally independent given $Y$? Yes




Common effect

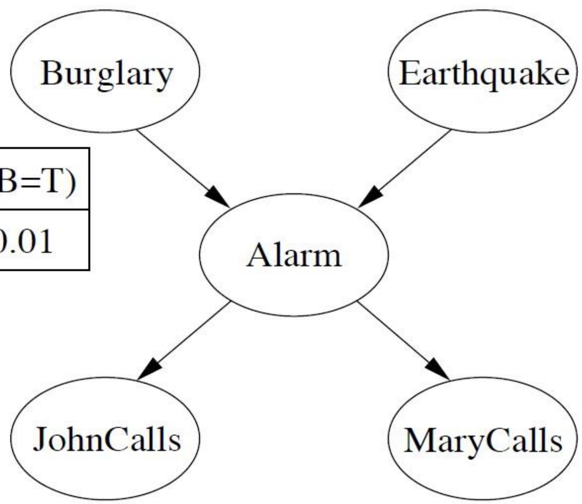
- Are $X$ and $Z$ independent? Yes
- Are they conditionally independent given $Y$? No

# Week 5

- PGM inference
  - Enumeration
  - Variable elimination algorithm
    - Steps
    - Complexity analysis
      - Graph reconstruction
      - Cliques

# PGM: Model Representation

- Directed acyclic graph
- Conditional probability table (parameters)




| P(B=T) |
|---|
| 0.01 |

| P(E=T) |
|---|
| 0.02 |

| B | E | P(A=T\|B,E) |
|---|---|---|
| T | T | 0.95 |
| T | F | 0.94 |
| F | T | 0.29 |
| F | F | 0.001 |

| A | P(J=T\|A) |
|---|---|
| T | 0.90 |
| F | 0.05 |

| A | P(M=T\|A) |
|---|---|
| T | 0.70 |
| F | 0.01 |

- Compact: just 10 rows vs 31 rows in a full joint table!

# Week 5

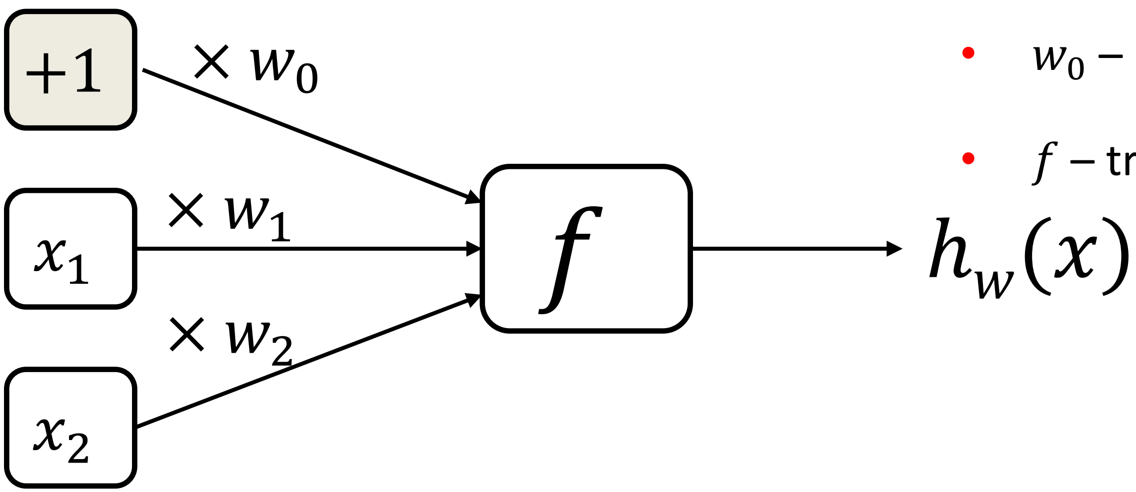
- Undirected PGMs
  - Representation
  - Joint factors as product of clique potentials
  - Normalise joint factors
- Markov property
- Applications
- Directed PGMs vs Undirected PGMs

# Week 6

- Learning ≈ optimisation
  - Find model parameters that minimise discrepancy with training data
- Gradient descent
  - Convergence guaranteed for convex functions
  - Convergence sensitive to learning rate
- Newton-Raphson method
- Regularisation

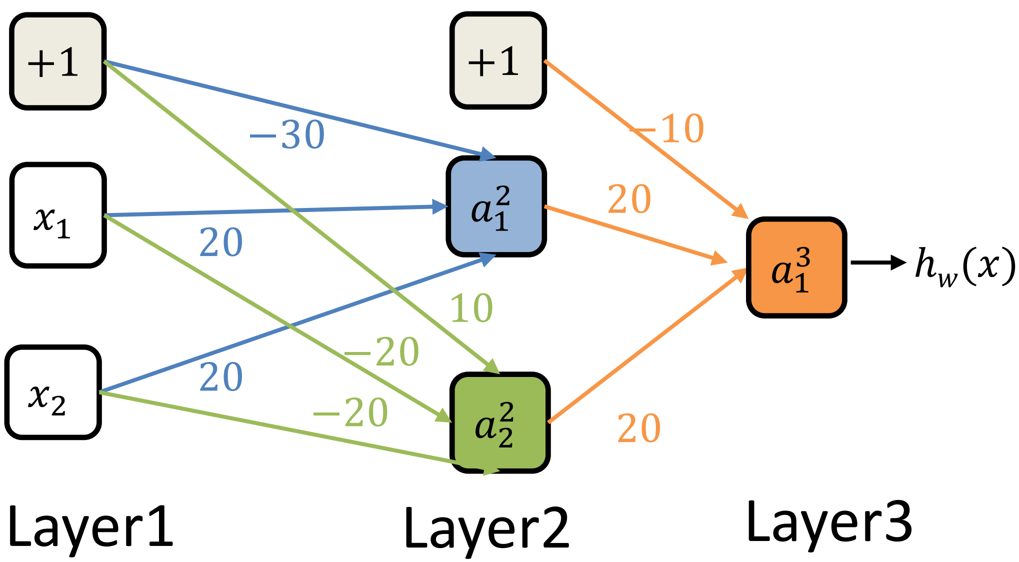
# Week 6 and Week 7

- Artificial Neural Network
  - Model representation
    - Perceptron Model




- $x_1, x_2$ – inputs
- $w_1, w_2$ – synaptic weights
- $w_0$ – bias weight
- $f$ – transfer function

# Model Representation




$a_i^j$ = activation of unit $i$ in layer $j$

$W^j$ = matrix of weights from layer $j$ to $j$+1

$$a_1^2 = f(W_{10}^1 x_0 + W_{11}^1 x_1 + W_{12}^1 x_2)$$

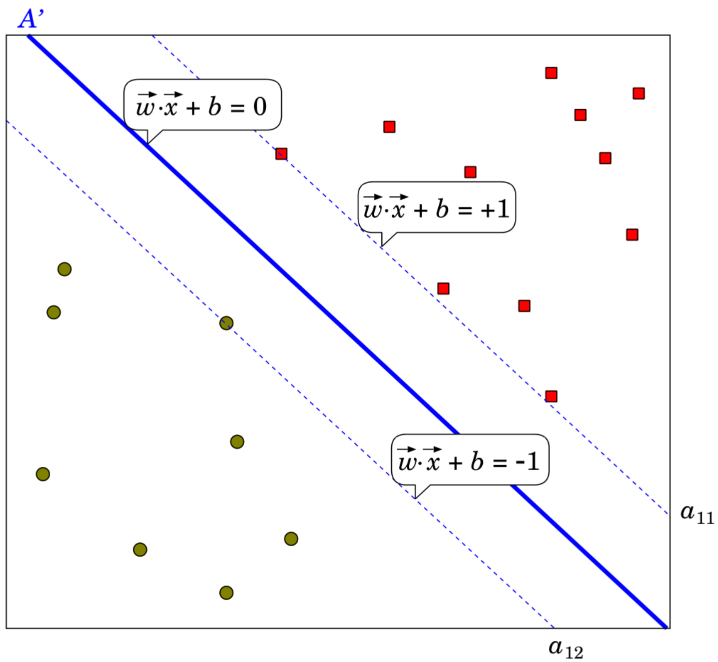
$$a_2^2 = f(W_{20}^1 x_0 + W_{21}^1 x_1 + W_{22}^1 x_2)$$

$$a_3^2 = f(W_{10}^2 a_0^2 + W_{11}^2 a_1^2 + W_{12}^2 a_2^2)$$

If #units in layer $j = s_j$
#units in layer $j + 1 = s_{j+1}$
then, $W^j = (s_j+1) \times s_{j+1}$

# Week 8

- SVMs
  - Maximum Margin
- A hyperplane is characterised by a normal $\vec{w}$ and offset $b$:

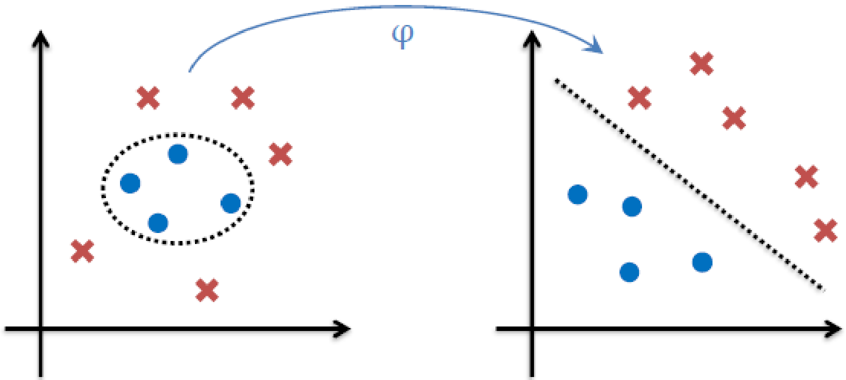



$$\text{margin} = \frac{2}{||\vec{w}||}$$

$$f(\vec{x}) = \begin{cases} +1 & \text{if } \vec{w} \cdot \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \cdot \vec{x} + b \leq -1 \end{cases}$$

# Non-Linear SVM

- Attribute transformation




- Kernel trick
  - Kernel = Similarity function
  - Computing similarity in the transformed space using the original attributes

# Regularisation and Parameters

- $C = \frac{1}{\lambda}$
  - Large C: Lower bias, high variance
  - Small C: Higher bias, low variance.
- $\sigma^2$
  - Large $\sigma^2$ : Features vary more smoothly. Higher bias, lower variance.
  - Small $\sigma^2$ : Features vary less smoothly. Lower bias, higher variance.

# Week 9

- Unsupervised learning
  - Clustering
  - Association rules mining
- Dimensionality reduction
  - PCA
    - Reduce #features
    - k-principal components
    - select dimensions that max variance
  - MDS

# Week 10

- Clustering analysis
  - Hard clustering: Each document belongs to exactly one cluster
  - Soft clustering: A document can belong to more than one cluster.
- Major Clustering Approaches
  - Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
  - Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Density-based algorithms: based on connectivity and density functions
  - Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other
  - Probabilistic Clustering: Rather than identifying clusters by "nearest" centroids, fit a Set of *k* Gaussians to the data

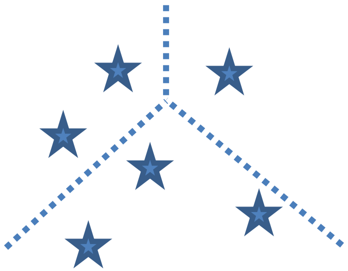
# Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based algorithms: based on connectivity and density functions
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other
- Probabilistic Clustering: Rather than identifying clusters by "nearest" centroids, fit a Set of *k* Gaussians to the data

# Exclusive vs. overlapping clustering

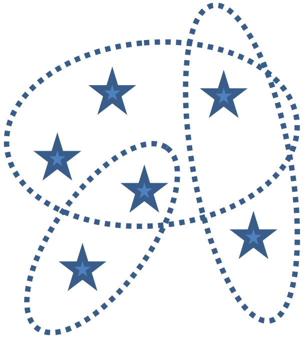
Exclusive

(hard clustering, partition of a set)



Overlapping

(fuzzy clustering, soft clustering)



Deterministic clustering is a combinatorial problem

# Deterministic vs. probabilistic clustering

Overlapping deterministic

| Object | Cluster membership |
|---|---|
| 1 | 2 |
| 2 | 1, 3 |
| 3 | 4 |
| … | |

Exclusive deterministic

| Object | Cluster membership |
|---|---|
| 1 | 2 |
| 2 | 1 |
| 3 | 4 |
| … | |

Probabilistic

| Object | Cluster | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 1 | 0.01 | 0.87 | 0.12 | 0.00 |
| 2 | 0.05 | 0.25 | 0.67 | 0.03 |
| 3 | 0.00 | 0.98 | 0.02 | 0.00 |
| … | | | | |

# Week 11

- Social network analysis
- Community detection algorithms
  - Edge betweenness
  - Modularity score
  - Clique percolation

# Edge Betweenness

***Girvan-Newman Method***

- Remove the edges of **highest betweenness** first.
- Repeat the same step with the remainder graph.
- Continue this until the graph breaks down into individual nodes.

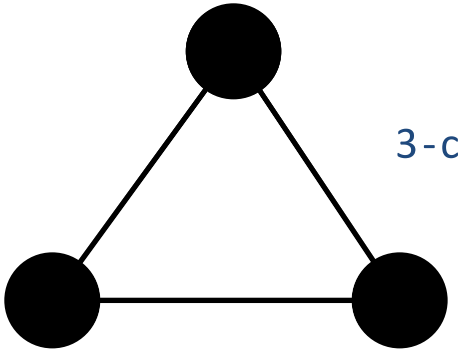
As the graph breaks down into pieces, the tightly knit community structure is exposed.

# Modularity based community detection

- *Modularity* is a measure that indicates how unexpected a set of communities are
  - The more unexpected, the more likely those communities are inherent ones
- Note that any random arrangement of graph will result in some form of communities
- Modularity measures the extent of deviation from randomness

# Clique Percolation Method CPM?

- Method to find **overlapping** communities
- Based on concept:
  - internal edges of community likely to form cliques
  - Intercommunity edges unlikely to form cliques
- Clique: Complete graph
  - k-clique: Complete graph with k vertices



# Week 12

- Semi-supervised learning
  - *Training with instances, some of which are labelled*
  - Self-training
  - Co-training
- Active-learning
  - *Iteratively request labels and use the model (trained thus far) to do so*
  - Sampling strategies
  - Query strategies