Clustering: K-Means

Machine Learning 10-601, Fall 2014

Bhavana Dalvi Mishra
PhD student LTI, CMU

Slides are based on materials from Prof. Eric Xing, Prof. William Cohen and Prof. Andrew Ng

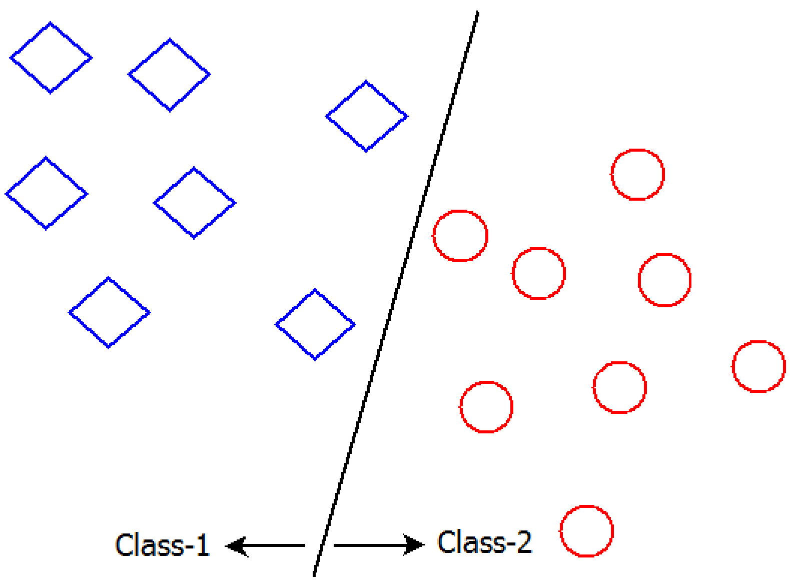
Outline

- What is clustering?
- How are similarity measures defined?
- Different clustering algorithms
  - K-Means
  - Gaussian Mixture Models
- Expectation Maximization
- Advanced topics
  - How to seed clustering?
  - How to choose #clusters
  - Application: Gloss finding for a Knowledge Base

Clustering

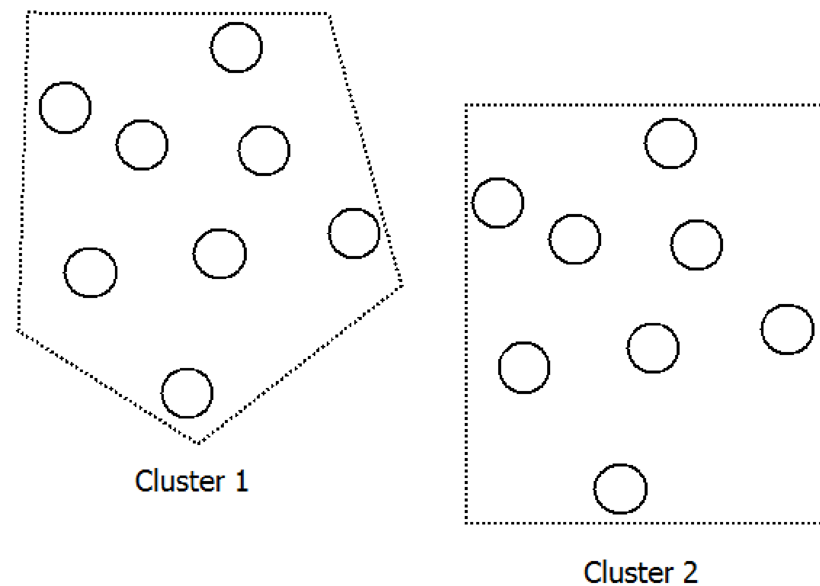
Classification vs. Clustering

Supervision available



Learning from supervised data: example classifications are given

Unsupervised

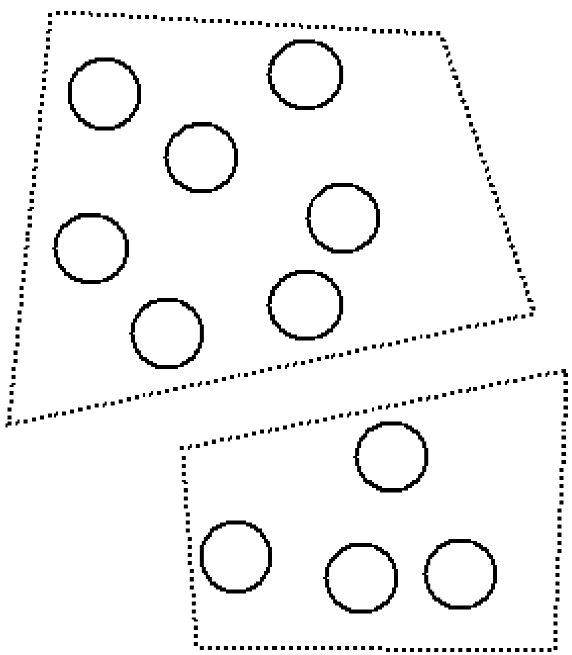


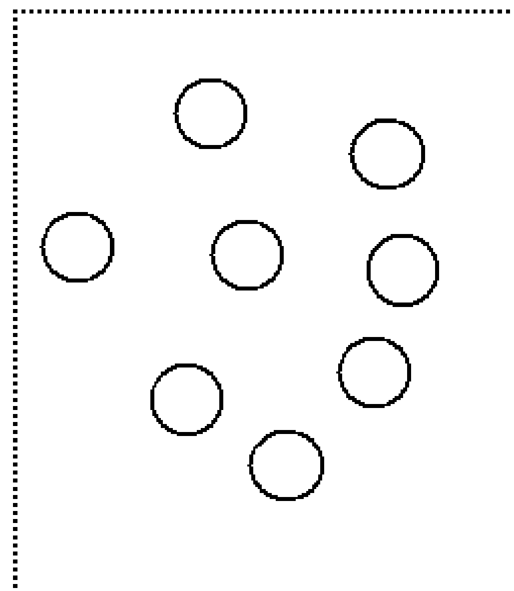
Unsupervised learning: learning from raw (unlabeled) data

Clustering

- The process of grouping a set of objects into clusters
  - high intra-cluster similarity
  - low inter-cluster similarity

How many clusters?
How to identify them?





Applications of Clustering

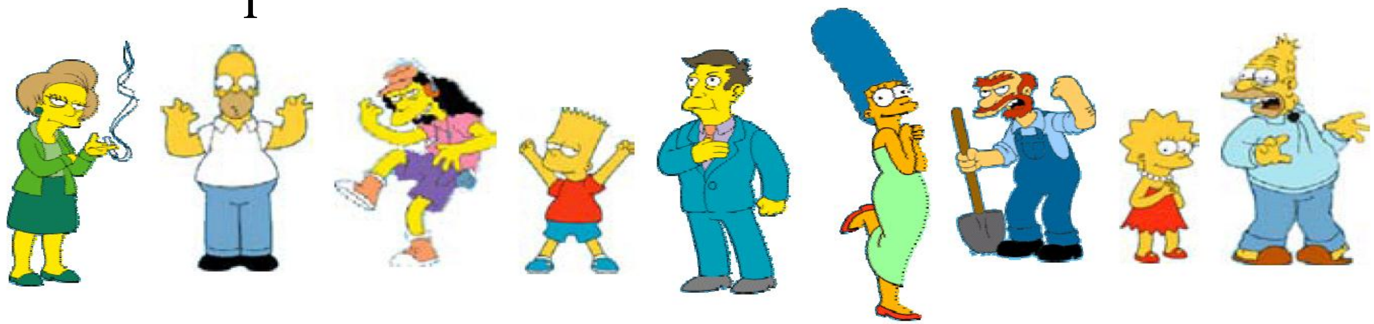
Google news: Clusters news stories from different sources about same event

….

- **Computational biology:** Group genes that perform the same functions
- **Social media analysis:** Group individuals that have similar political views
- **Computer graphics:** Identify similar objects from pictures

Examples

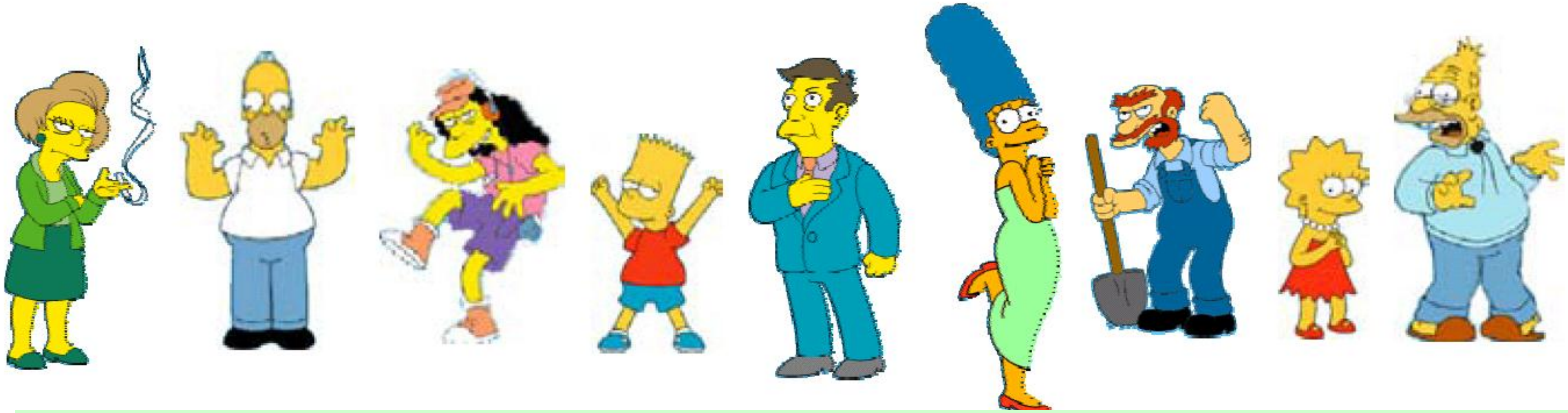
- People



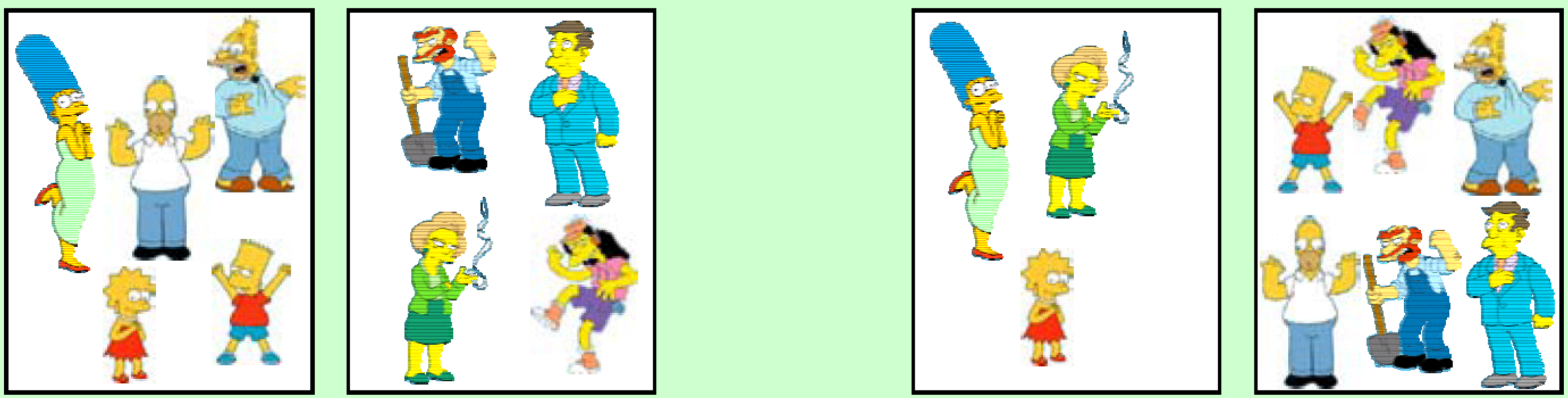
- Images

- Species

What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family | School Employees | Females | Males

Similarity Measures

What is Similarity?

Hard to define!
But *we know it*
when we see it

- The real meaning of similarity is a philosophical question.
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance (rather than similarity) between vectors.

Intuitions behind desirable distance measure properties

- $D(\mathrm{A},\mathrm{B}) = D(\mathrm{B},\mathrm{A})$ *Symmetry*
- $D(\mathrm{A},\mathrm{A}) = 0$ *Constancy of Self-Similarity*
- $D(\mathrm{A},\mathrm{B}) = 0$ IIf $\mathrm{A} = \mathrm{B}$ *Identity of indiscernibles*
- $D(\mathrm{A},\mathrm{B}) \leq D(\mathrm{A},\mathrm{C}) + D(\mathrm{B},\mathrm{C})$ *Triangular Inequality*

Intuitions behind desirable distance measure properties

- $D(A,B) = D(B,A)$ *Symmetry*
  - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"*
- $D(A,A) = 0$ *Constancy of Self-Similarity*
  - *Otherwise you could claim "Alex looks more like Bob, than Bob does"*
- $D(A,B) = 0$ IIf $A = B$ *Identity of indiscernibles*
  - *Otherwise there are objects in your world that are different, but you cannot tell apart.*
- $D(A,B) \leq D(A,C) + D(B,C)$ *Triangular Inequality*
  - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

Distance Measures: Minkowski Metric

- Suppose two object x and y both have p features

$$x = (x_1, x_2, \cdots, x_p)$$
$$y = (y_1, y_2, \cdots, y_p)$$

- The Minkowski metric is defined by

$$d(x, y) = \sqrt[r]{\sum_{i=1}^{p} |x_i - y_i|^r}$$

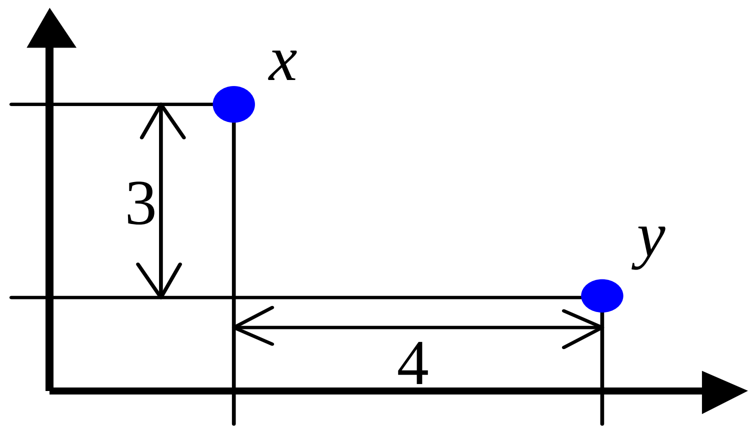
- Most Common Minkowski Metrics

$$r = 2 \text{ (Euclidean distance)} \qquad d(x, y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$

$$r = 1 \text{ (Manhattan distance)} \qquad d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

$$r = +\infty \text{ ("sup" distance)} \qquad d(x, y) = \max_{1 \le i \le p} |x_i - y_i|$$

An Example



$$1:\ \text{Euclidean distance}: \quad \sqrt[2]{4^2+3^2}=5.$$

$$2:\ \text{Manhattan distance}: \quad 4+3=7.$$

$$3:\ \text{"sup" distance}: \quad \max\{4,3\}=4.$$

Hamming distance

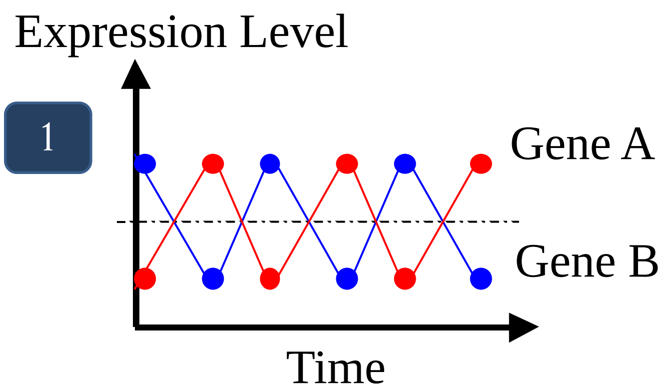
- Manhattan distance is called *Hamming distance* when all features are binary.
  - Gene Expression Levels Under 17 Conditions (1-High,0-Low)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
GeneA	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
GeneB	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

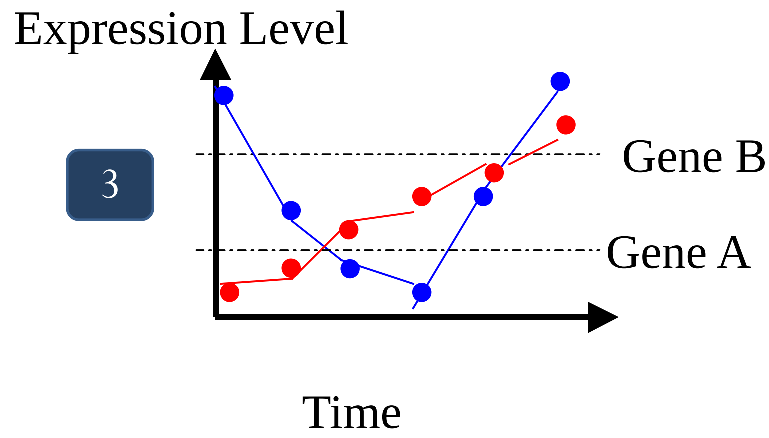
$$\boxed{\text{Hamming Distance}: \#(01) + \#(10) = 4 + 1 = 5.}$$

Similarity Measures: Correlation Coefficient

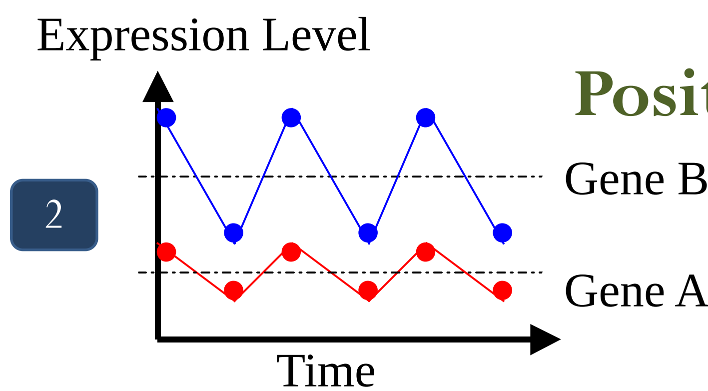
Negatively correlated



Uncorrelated



Positively correlated



Similarity Measures: Correlation Coefficient

- Pearson correlation coefficient

$$s(x, y) = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2 \times \sum_{i=1}^{p}(y_i - \bar{y})^2}}$$

$$\text{where } \bar{x} = \tfrac{1}{p}\sum_{i=1}^{p} x_i \text{ and } \bar{y} = \tfrac{1}{p}\sum_{i=1}^{p} y_i .$$

$$-1 \le s(x, y) \le 1$$

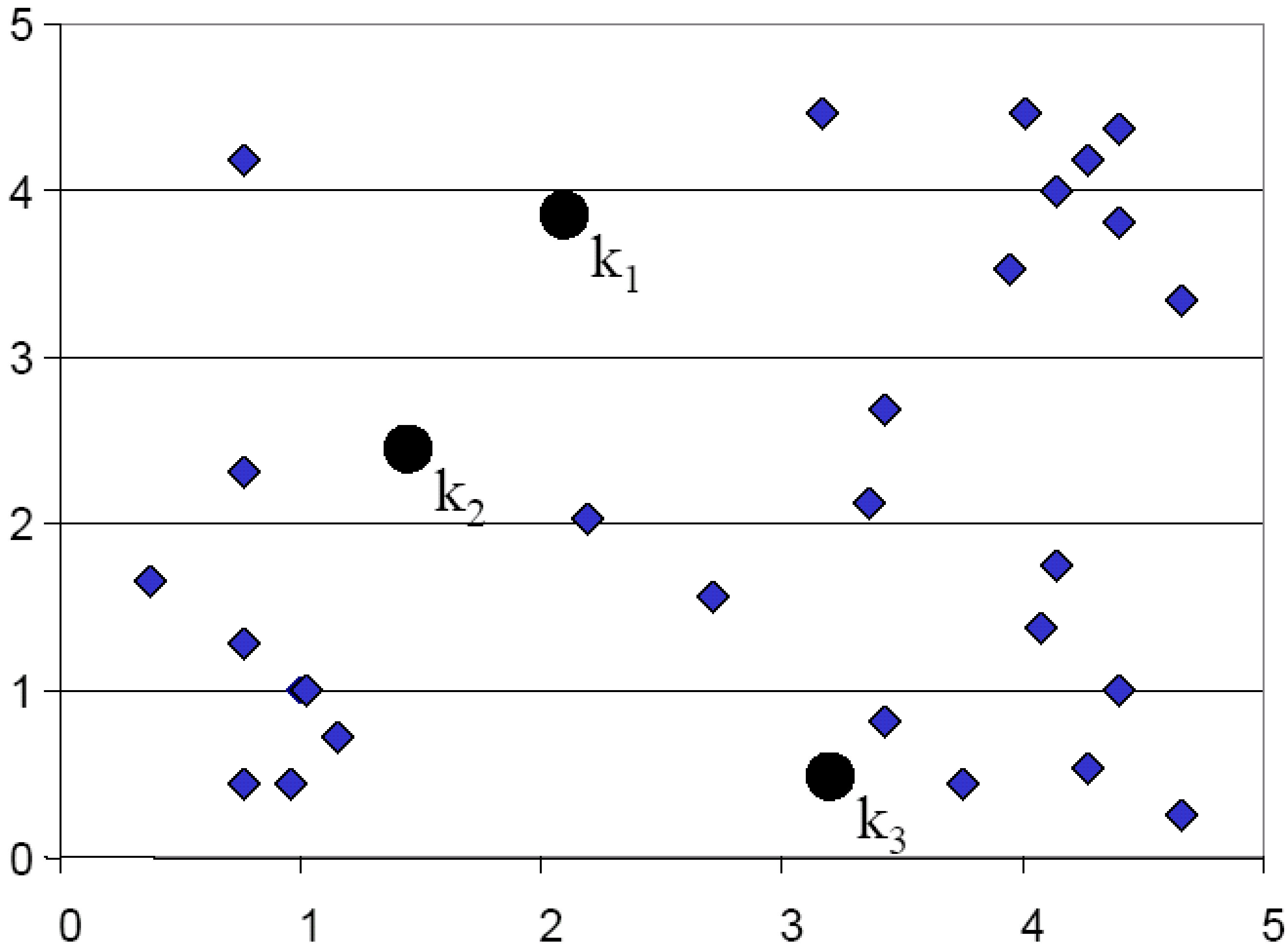
- Special case: cosine distance

$$s(x, y) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

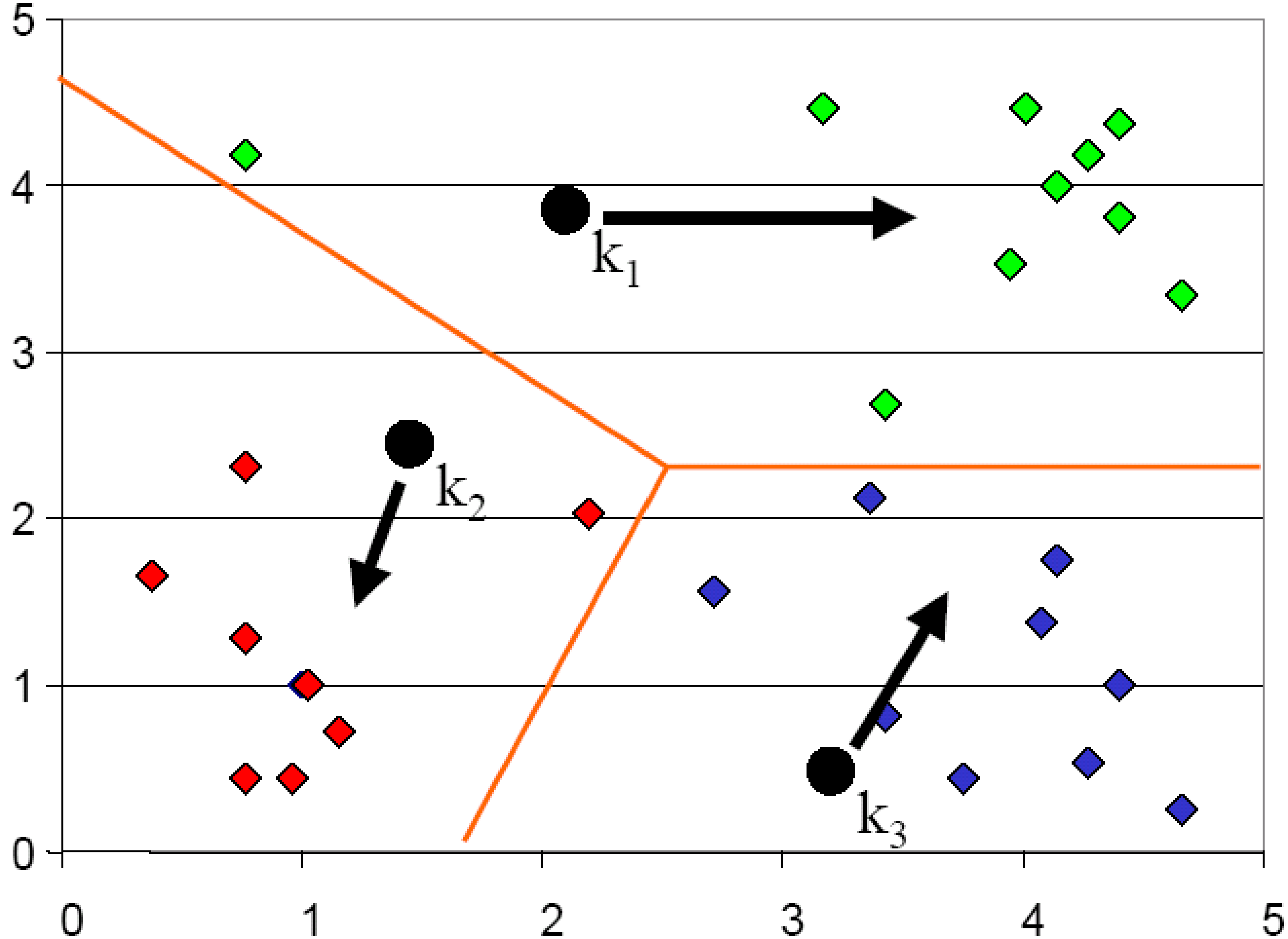
Clustering Algorithm

K-Means

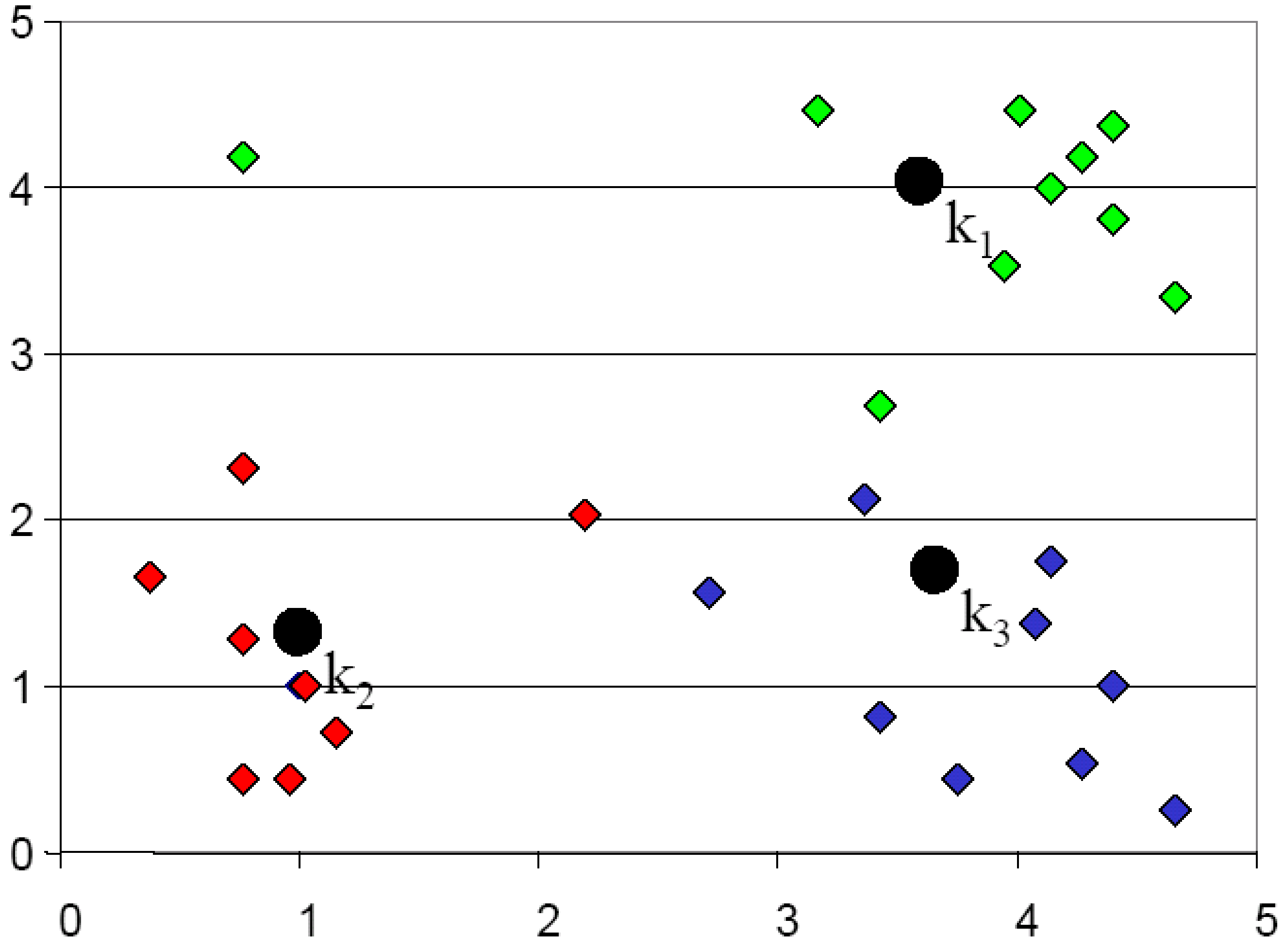
K-means Clustering: Step 1



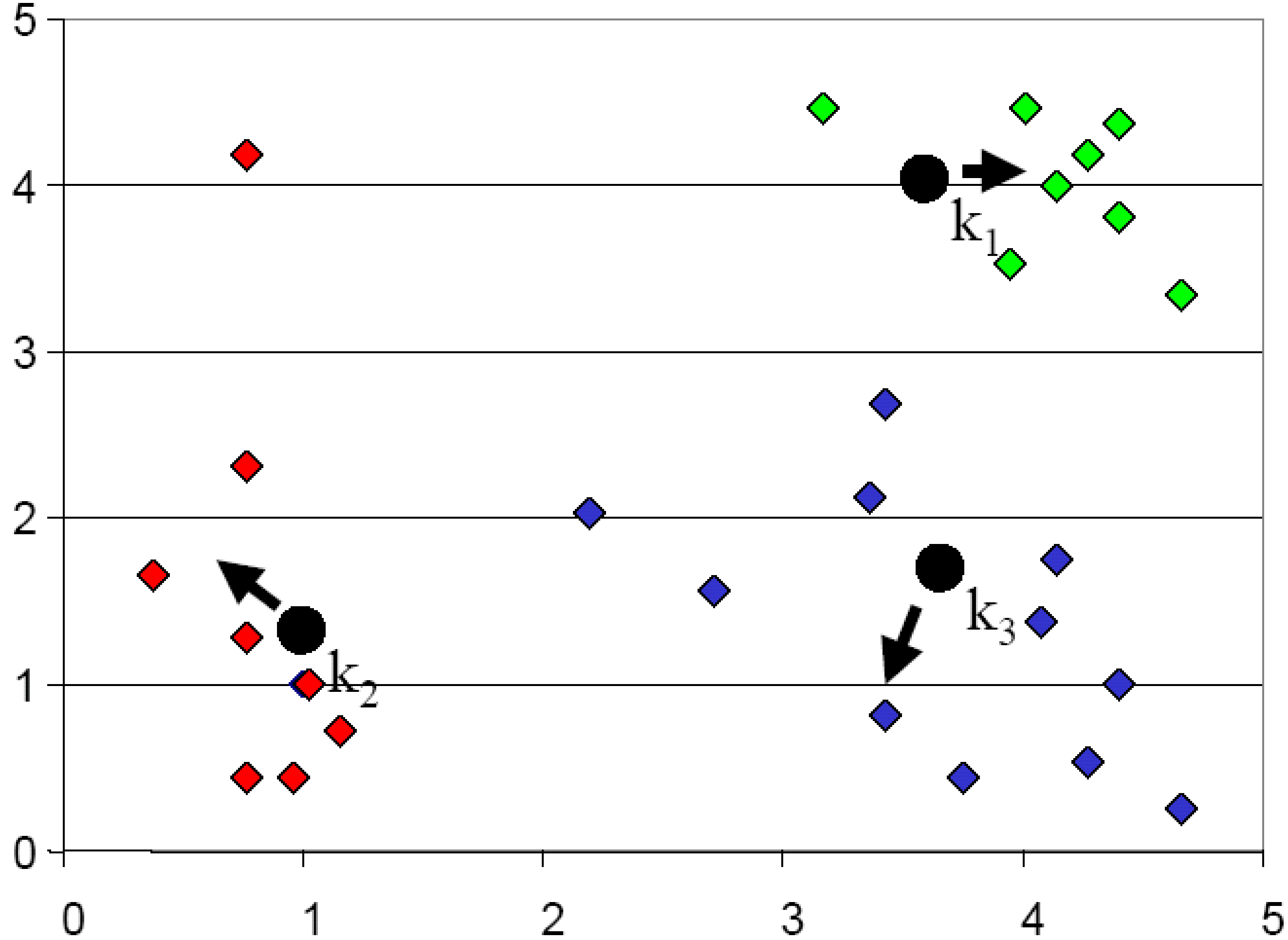
K-means Clustering: Step 2



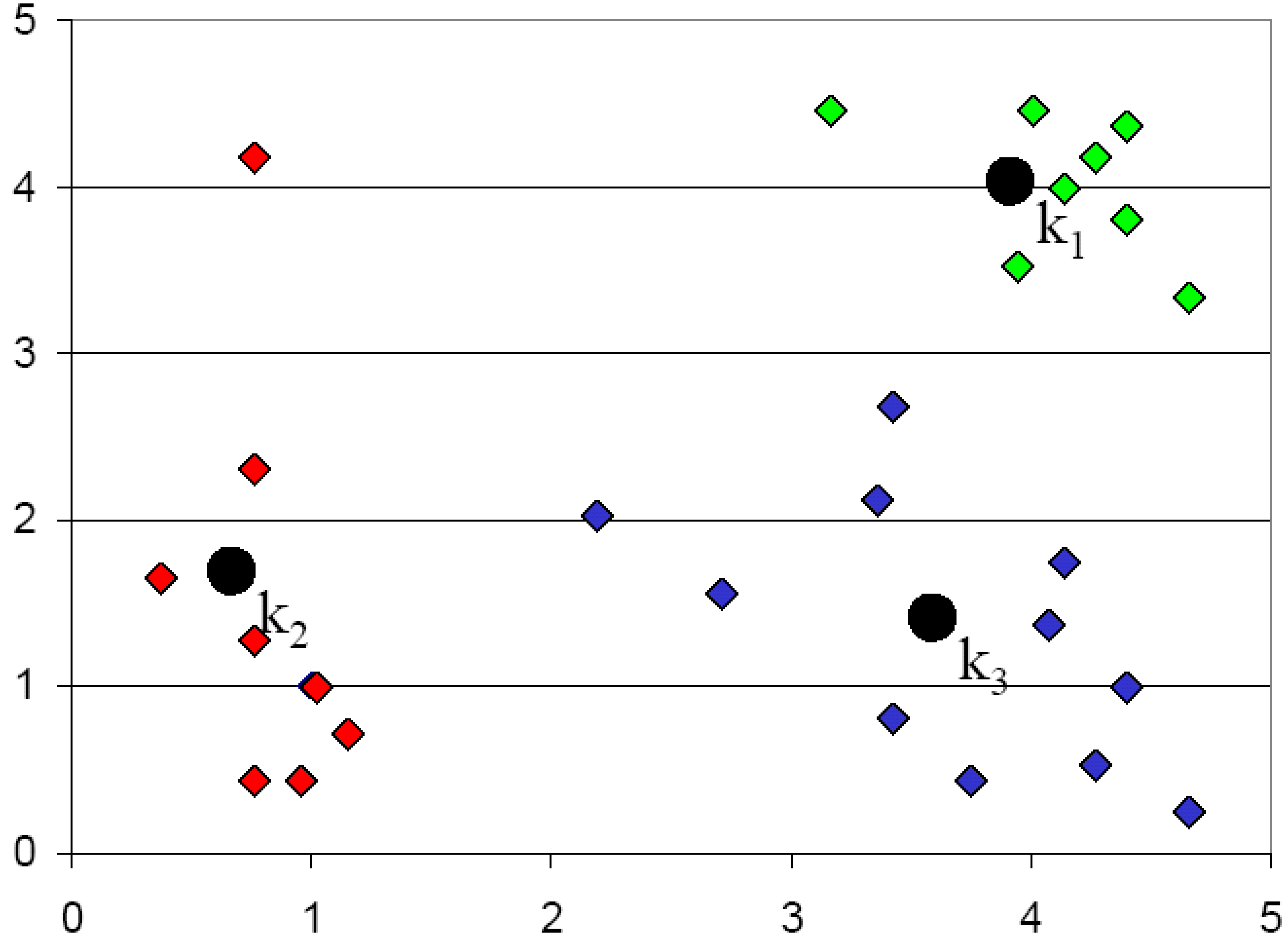
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



K-Means: Algorithm

1. Decide on a value for *k*.
2. Initialize the *k* cluster centers randomly if necessary.
3. Repeat till any object changes its cluster assignment
   - Decide the cluster memberships of the *N* objects by assigning them to the nearest ***cluster centroid***

$$cluster(\vec{x}_i) = \arg\min_j d(\vec{x}_i, \vec{\mu}_j)$$

   - Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

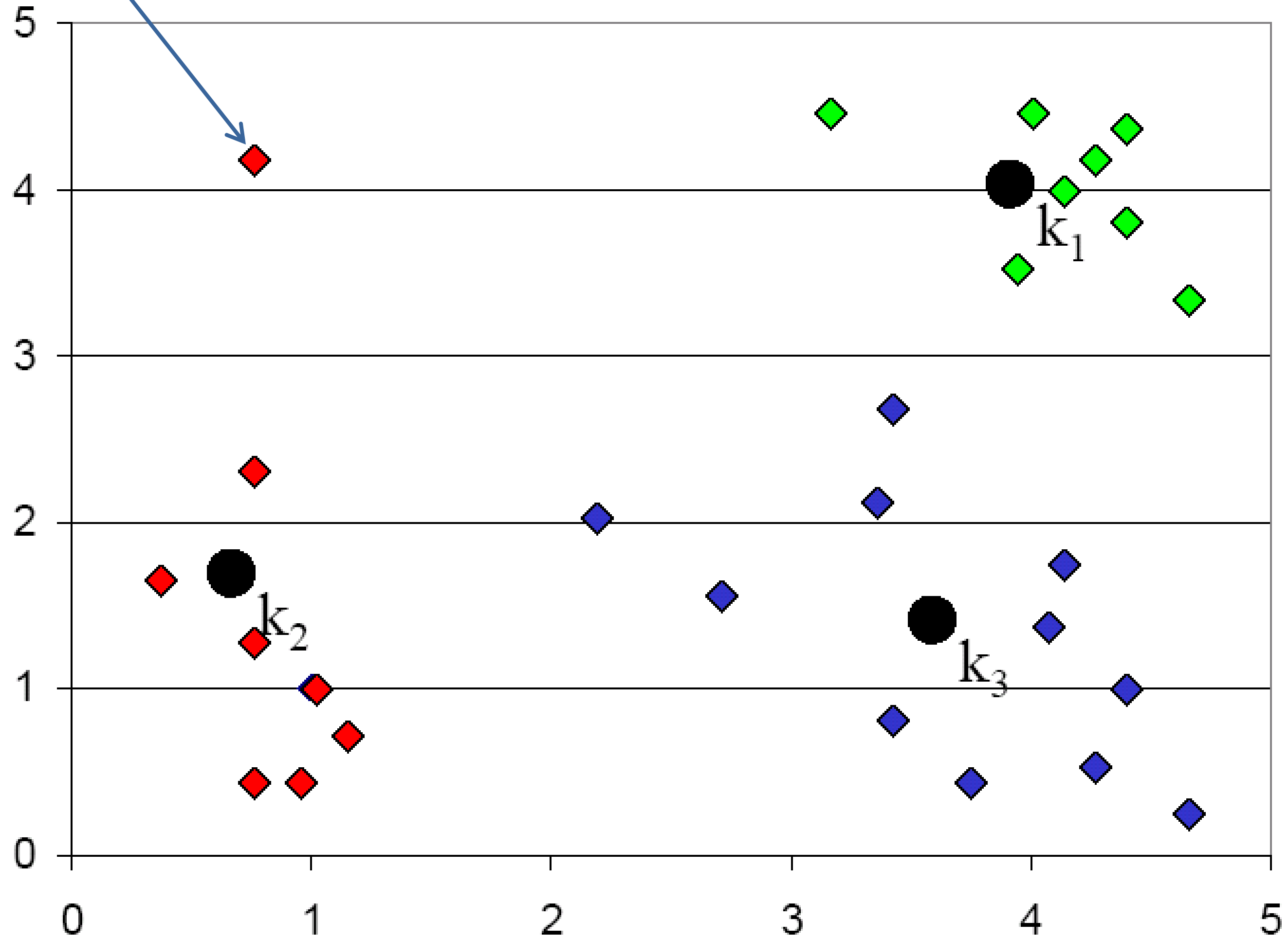
K-Means is widely used in practice

- Extremely fast and scalable: used in variety of applications

- Can be easily parallelized
  - Easy Map-Reduce implementation
  - Mapper: assigns each datapoint to nearest cluster
  - Reducer: takes all points assigned to a cluster, and re-computes the centroids

- Sensitive to starting points or random seed initialization (Similar to Neural networks)
  - There are extensions like K-Means++ that try to solve this problem

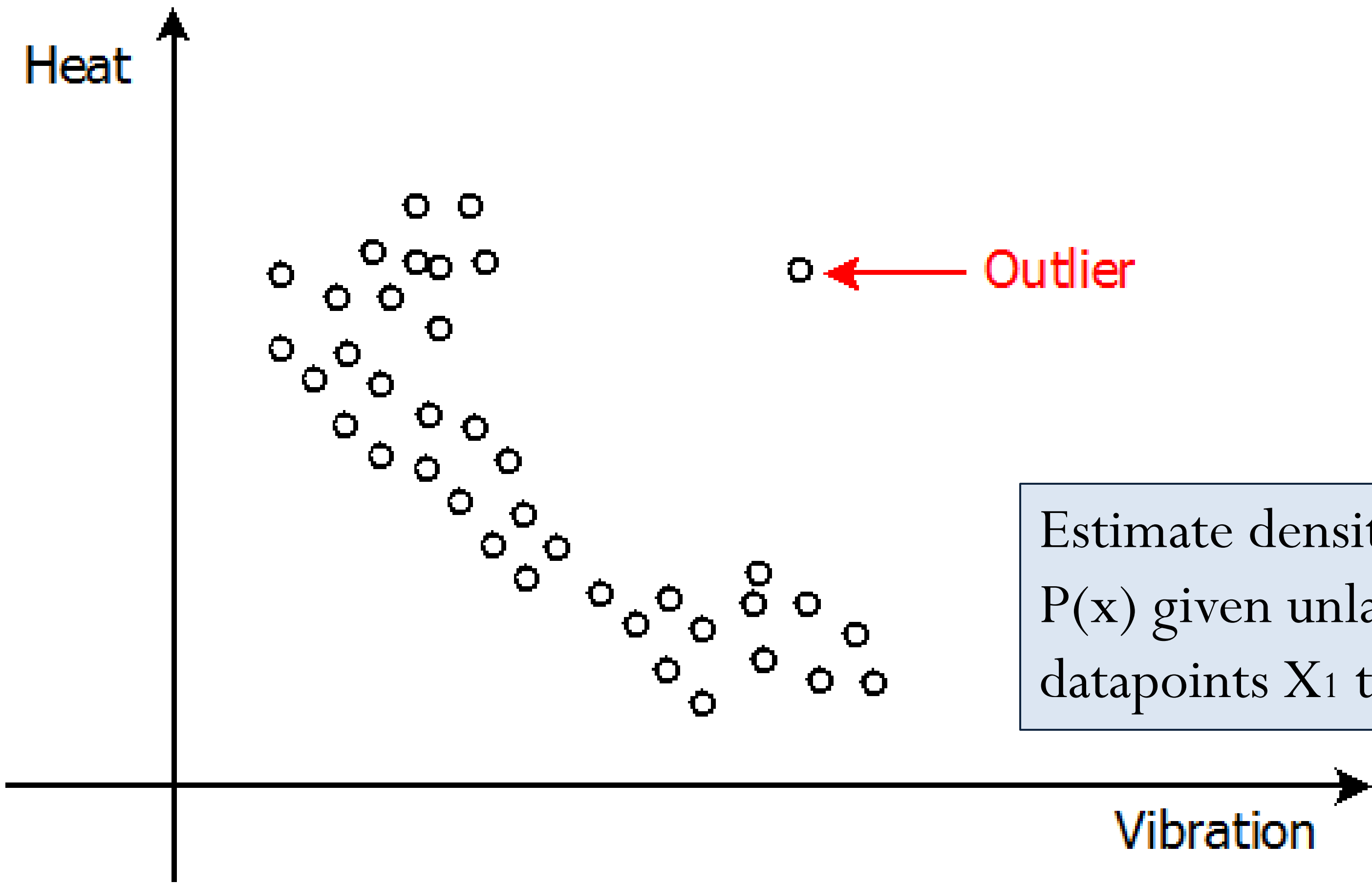
Outliers



Clustering Algorithm

Gaussian Mixture Model

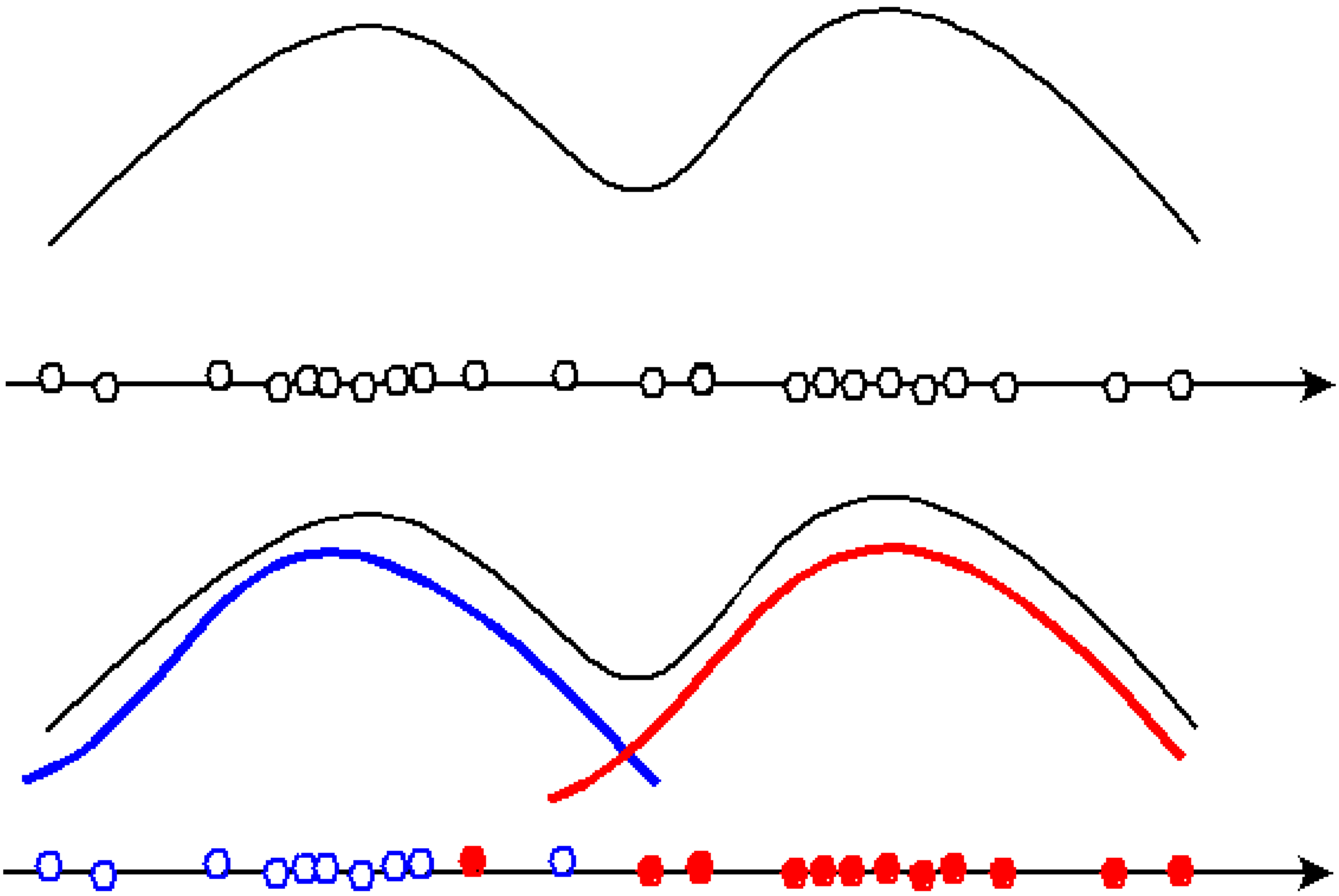
Density estimation



Estimate density function P(x) given unlabeled datapoints X_1 to X_n

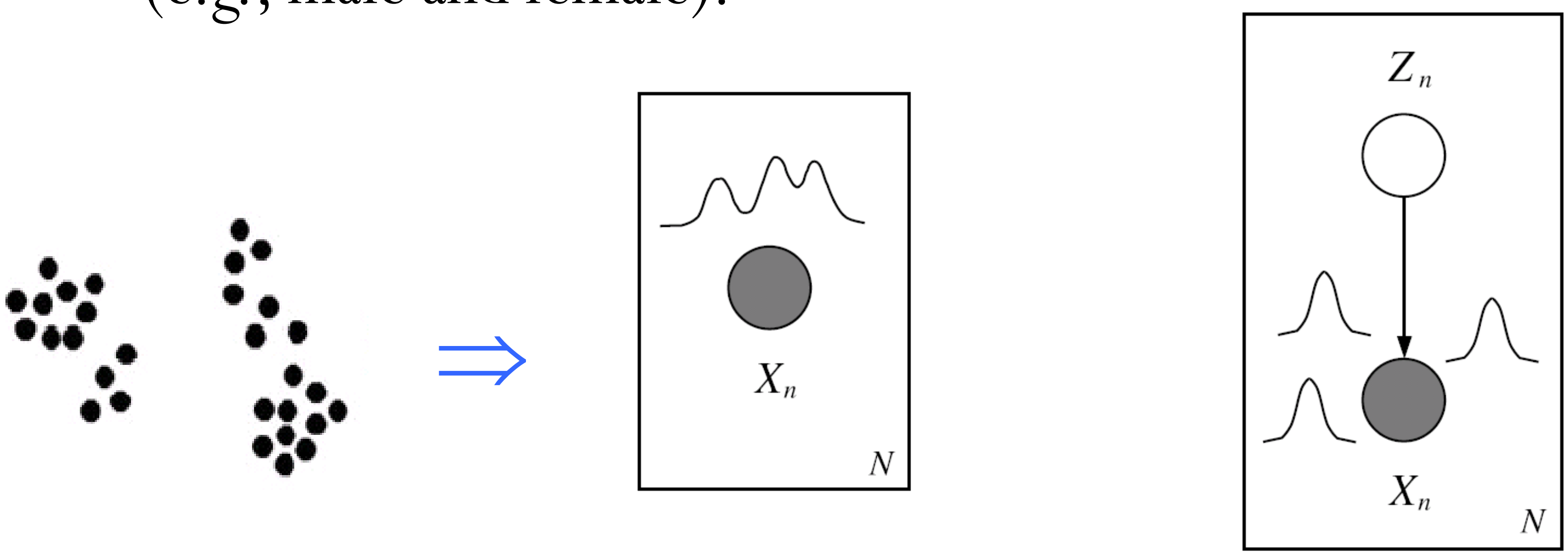
An aircraft testing facility measures Heat and Vibration parameters for every newly built aircraft.

Mixture of Gaussians



Mixture Models

- A density model $p(X)$ may be multi-modal.
- We may be able to model it as a mixture of uni-modal distributions (e.g., Gaussians).
- Each mode may correspond to a different sub-population (e.g., male and female).



(a) (b)

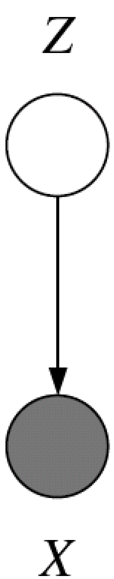
Gaussian Mixture Models (GMMs)

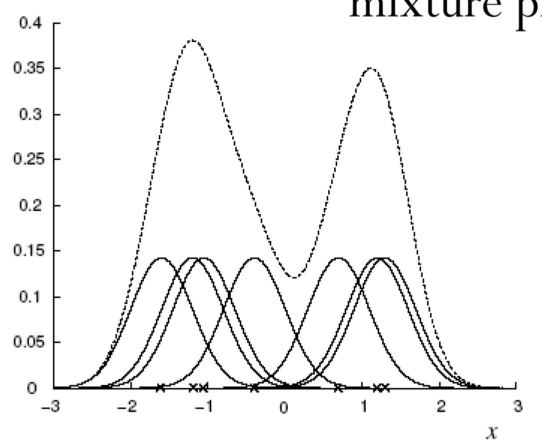
- Consider a mixture of K Gaussian components:

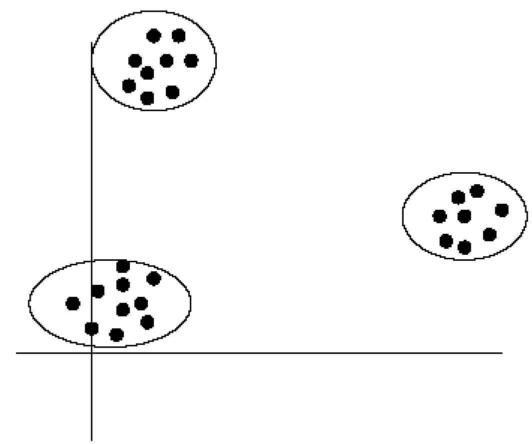
$$\boxed{p(x_n|\mu,\Sigma)=\sum_{i=1}^{K}\pi_i N(x\,|\,\mu_i,\Sigma_i)}$$

mixture component

mixture proportion







- This model can be used for unsupervised clustering.
  - This model (fit by AutoClass) has been used to discover new kinds of stars in astronomical data, etc.

Learning mixture models

- In fully observed iid settings, the log likelihood decomposes into a sum of local terms.

$$\ell_c(\theta; D) = \log p(x, z \mid \theta) = \log p(z \mid \theta_z) + \log p(x \mid z, \theta_x)$$

- With latent variables, all the parameters become coupled together via *marginalization*

$$\ell_c(\theta; D) = \log \sum_z p(x, z \mid \theta) = \log \sum_z p(z \mid \theta_z) p(x \mid z, \theta_x)$$

MLE for GMM

- If we are doing MLE for completely observed data
- Data log-likelihood

$$\ell(\boldsymbol{\theta}; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)$$
$$= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma_k)^{z_n^k}$$
$$= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \frac{1}{2\sigma_k^2}(x_n - \mu_k)^2 + C$$

- MLE

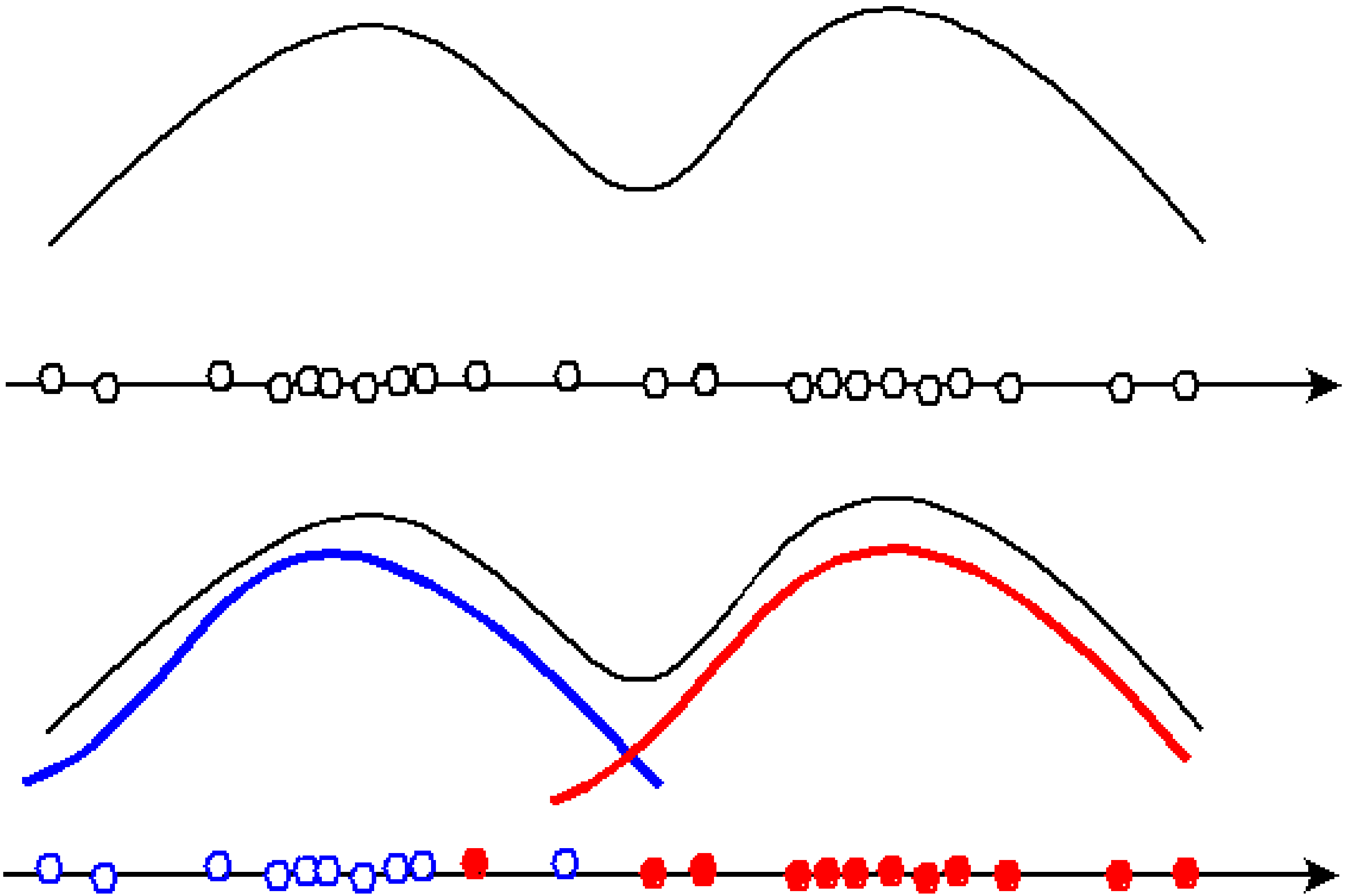
$$\hat{\pi}_{k,MLE} = \arg\max_\pi \ell(\boldsymbol{\theta}; D), \quad \Rightarrow \hat{\pi}_{k,MLE} = \sum_i z_i^k \Big/ \text{Number of datapoints}$$

$$\hat{\mu}_{k,MLE} = \arg\max_\mu \ell(\boldsymbol{\theta}; D) \quad \Rightarrow \quad \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$

$$\hat{\sigma}_{k,MLE} = \arg\max_\sigma \ell(\boldsymbol{\theta}; D)$$

Gaussian Naïve Bayes

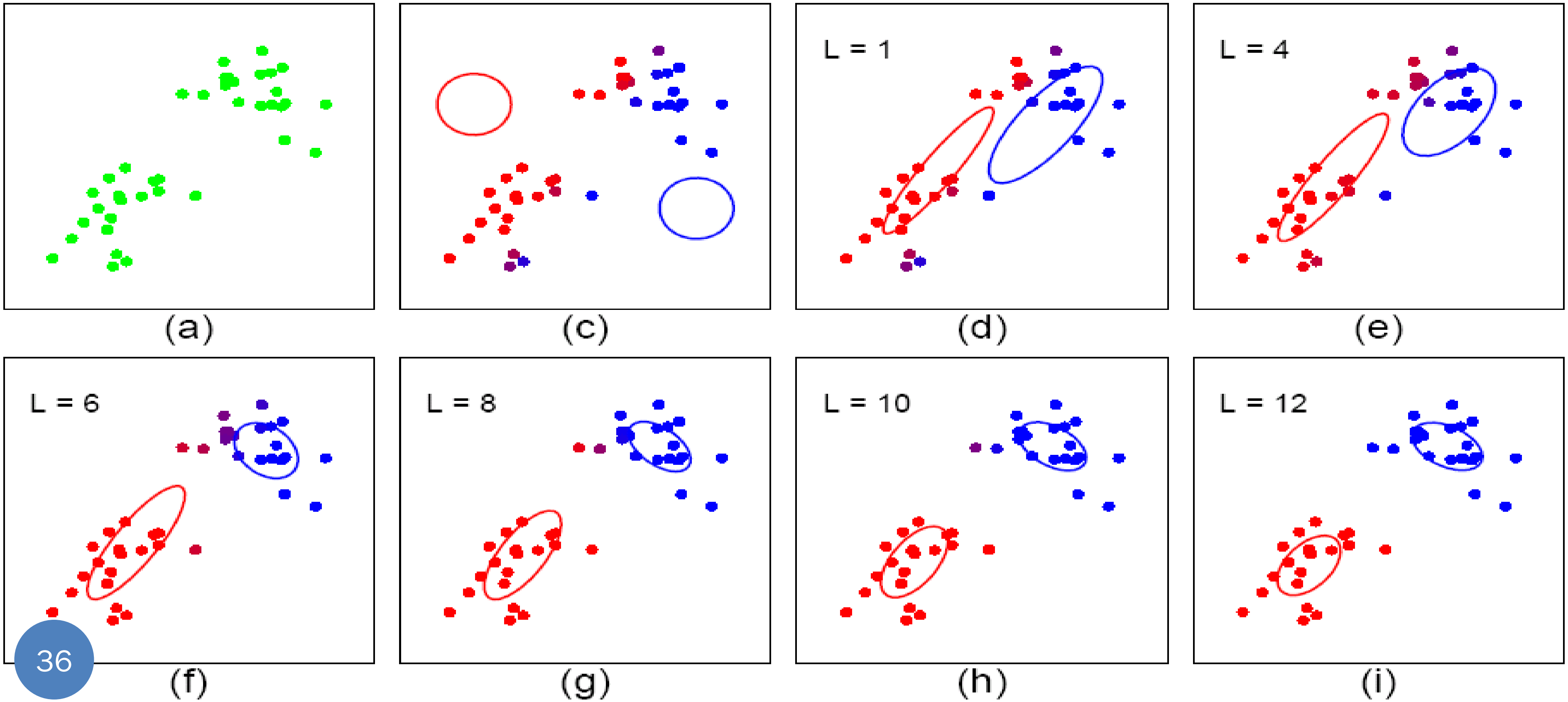
Learning GMM (z's are unknown)

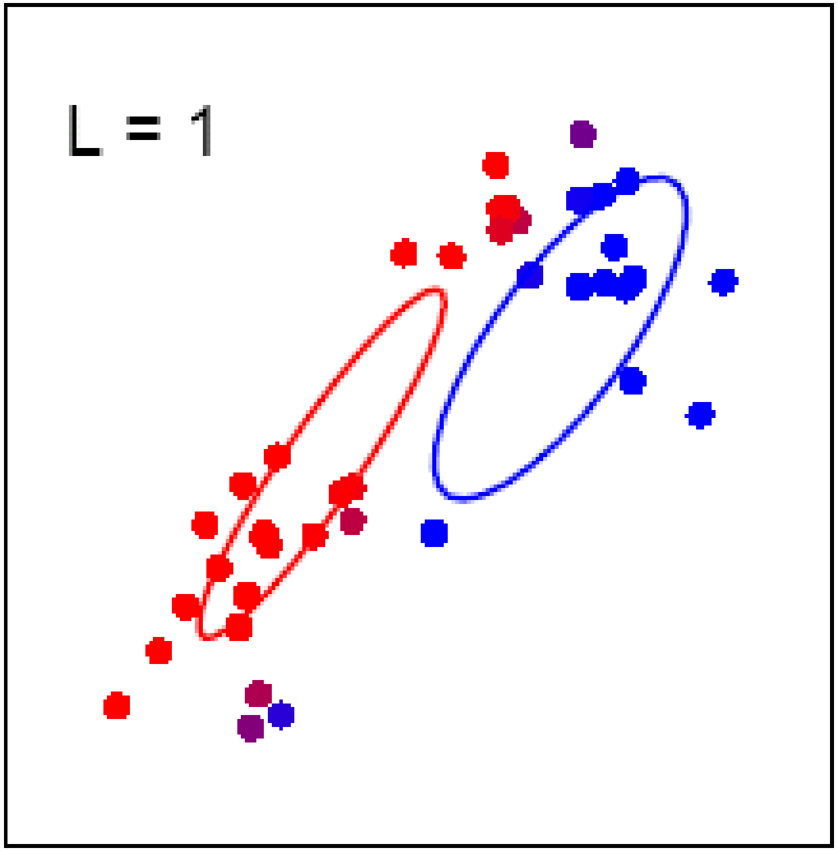


Expectation Maximization (EM)

Expectation-Maximization (EM)

- Start: "Guess" the mean and covariance of each of the K gaussians
- Loop

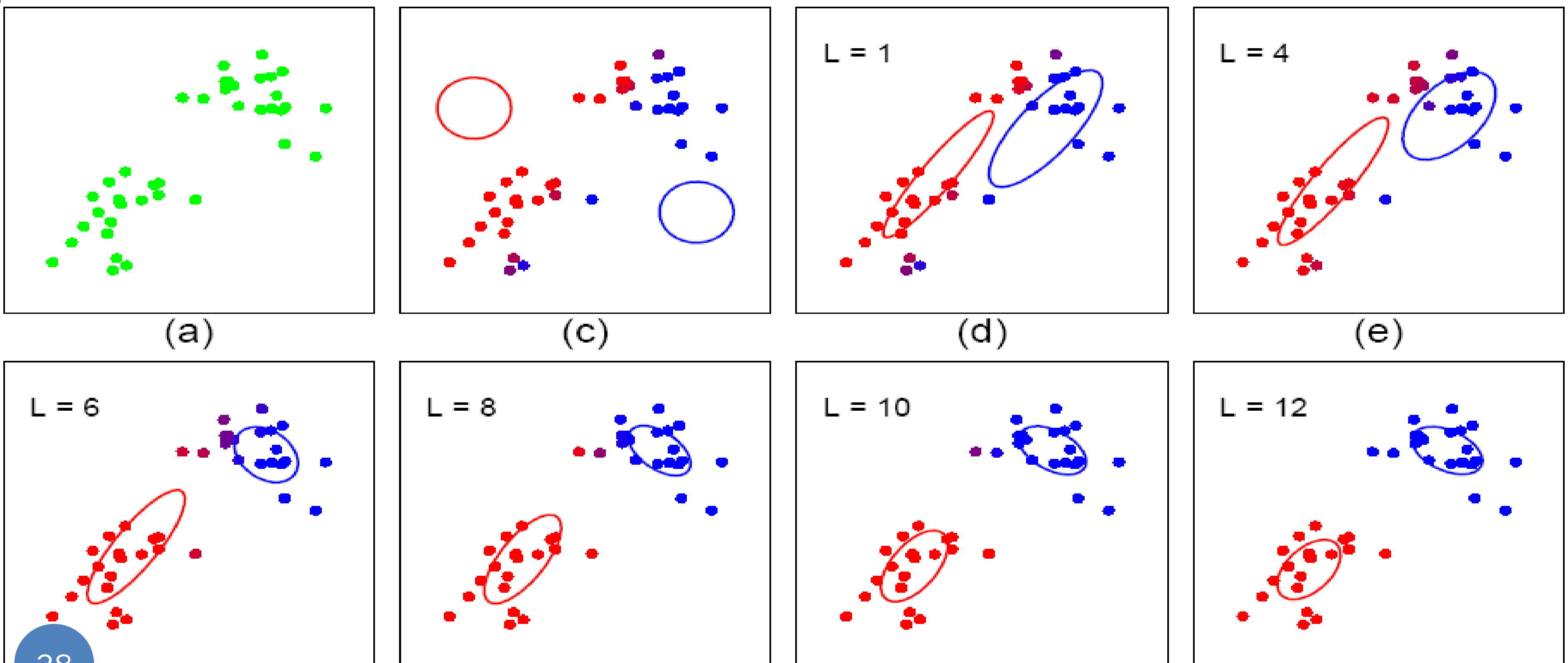




(d)

Expectation-Maximization (EM)

- Start: "Guess" the centroid and covariance of each of the K clusters
- Loop



(a) (c) (d) (e)

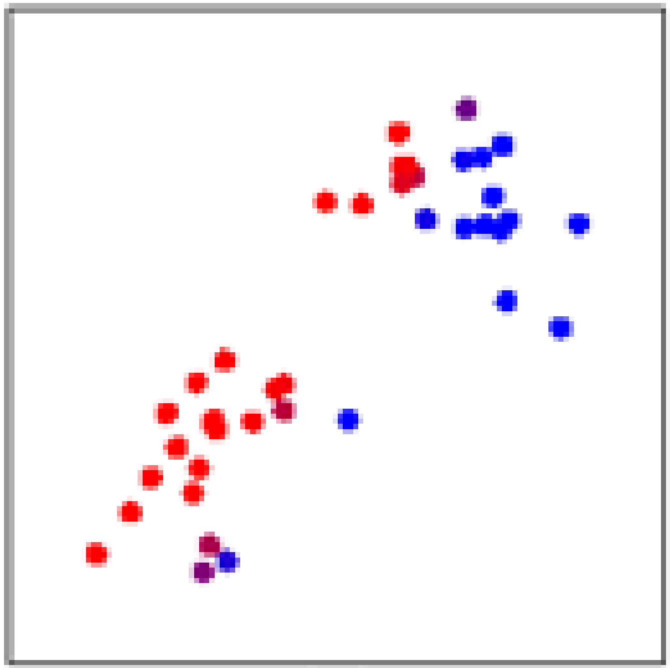
(f) (g) (h) (i)

The Expectation-Maximization (EM) Algorithm

- **E Step: Guess values of Z's**

$$w_j^{i(t)} = p(z^i = j \mid x^i, \pi^{(t)}, \mu^{(t)}, \Sigma^{(t)})$$

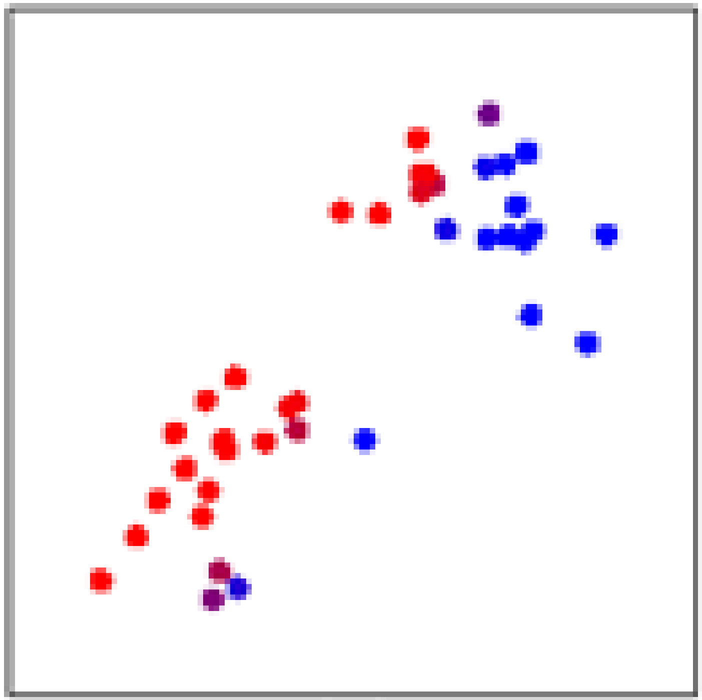
$$= \frac{p(x^i \mid z^i = j) \times P(z^i = j)}{\sum_{l=1}^{k} p(x^i \mid z^i = l) \times P(z^i = l)}$$



$$p(x^i \mid z^i = j) = N(\boxed{\mu_j^{(t)}}, \boxed{\Sigma_j^{(t)}})$$

$$\boxed{\pi_k^{(t)}} = P(Z^i = k)$$

The Expectation-Maximization (EM) Algorithm



- **M Step: Update parameter estimates**

$$\pi_{\mathrm{k}}^{(\mathrm{t}+1)} = P(Z^i = k) = \frac{\sum_n w_n^{k(t)}}{\#\,\mathrm{datapoints}}$$

$$\mu_k^{(t+1)} = \frac{\sum_n w_n^{k(t)} x_n}{\sum_n w_n^{k(t)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_n w_n^{k(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_n w_n^{k(t)}}$$

EM Algorithm for GMM

- **E Step: Guess values of Z's**

$$w_j^{i(t)} = p(z^i = j \mid x^i, \boxed{\pi^{(t)}, \mu^{(t)}, \Sigma^{(t)}})$$

$$= \frac{p(x^i \mid z^i = j) \times P(z^i = j)}{\sum_{l=1}^{k} p(x^i \mid z^i = l) \times P(z^i = l)}$$

- **M Step: Update parameter estimates**

$$\pi_k^{(t+1)} = P(Z^i = k) = \sum_n \boxed{w_n^{k(t)}} \Big/ N$$

$$\mu_k^{(t+1)} = \frac{\sum_n w_n^{k(t)} x_n}{\sum_n w_n^{k(t)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_n w_n^{k(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_n w_n^{k(t)}}$$

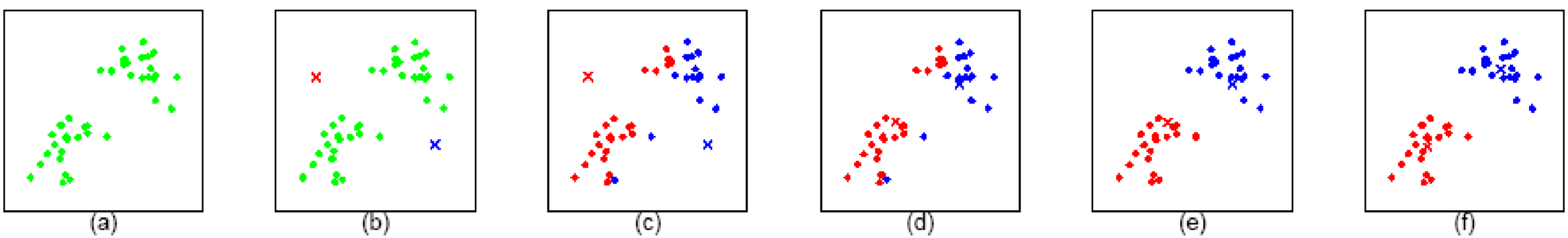
K-means is a hard version of EM

- In the K-means "E-step" we do hard assignment:

$$z_n^{(t)} = \boxed{\arg\max_k} (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

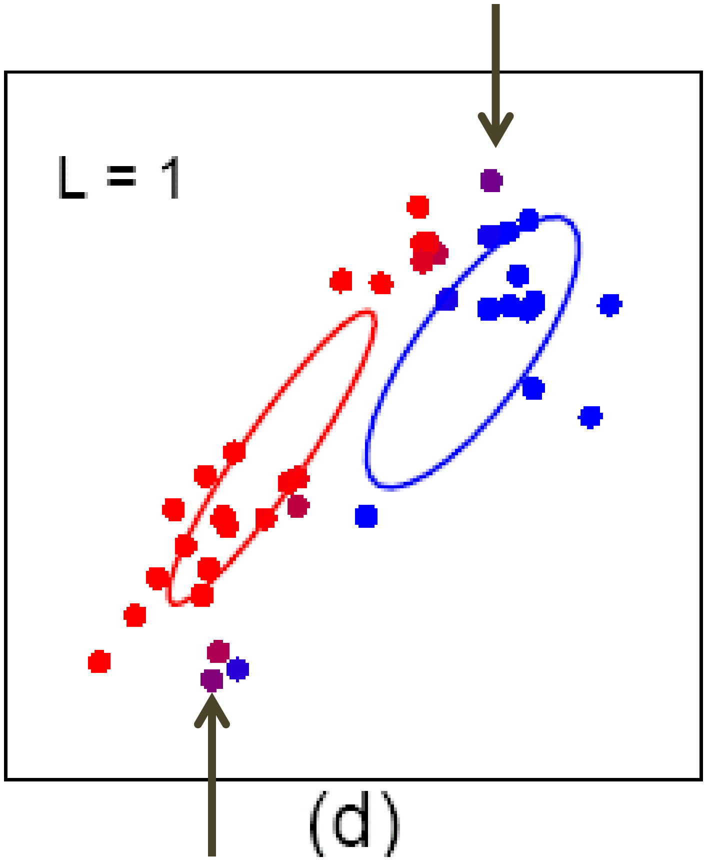
- In the K-means "M-step" we update the means as the weighted sum of the data, but now the weights are 0 or 1:

$$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)}$$

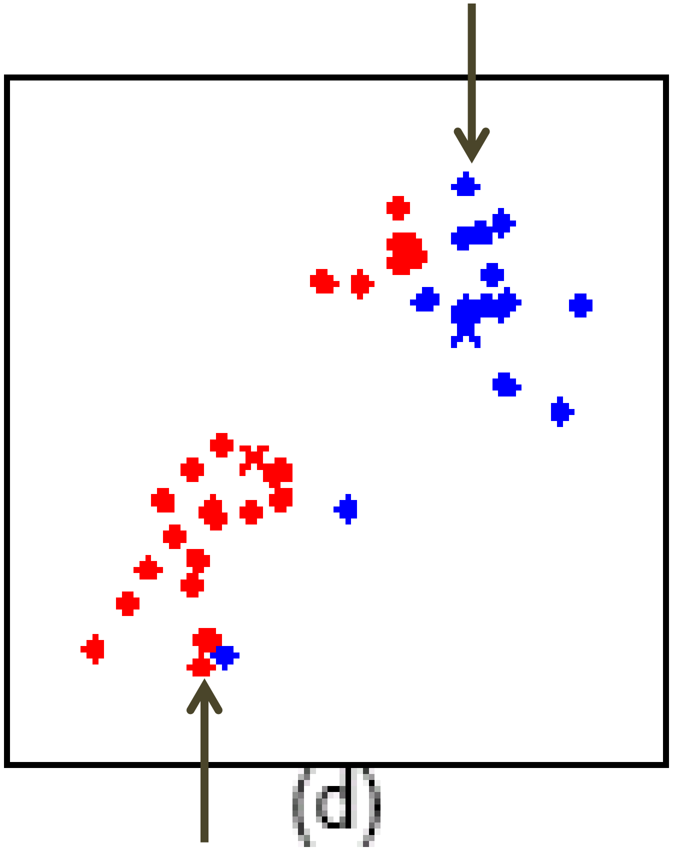


Soft vs. Hard EM assignments

- GMM



- K-Means



Theory underlying EM

- What are we doing?

- Recall that according to MLE, we intend to learn the model parameters that would maximize the likelihood of the data.

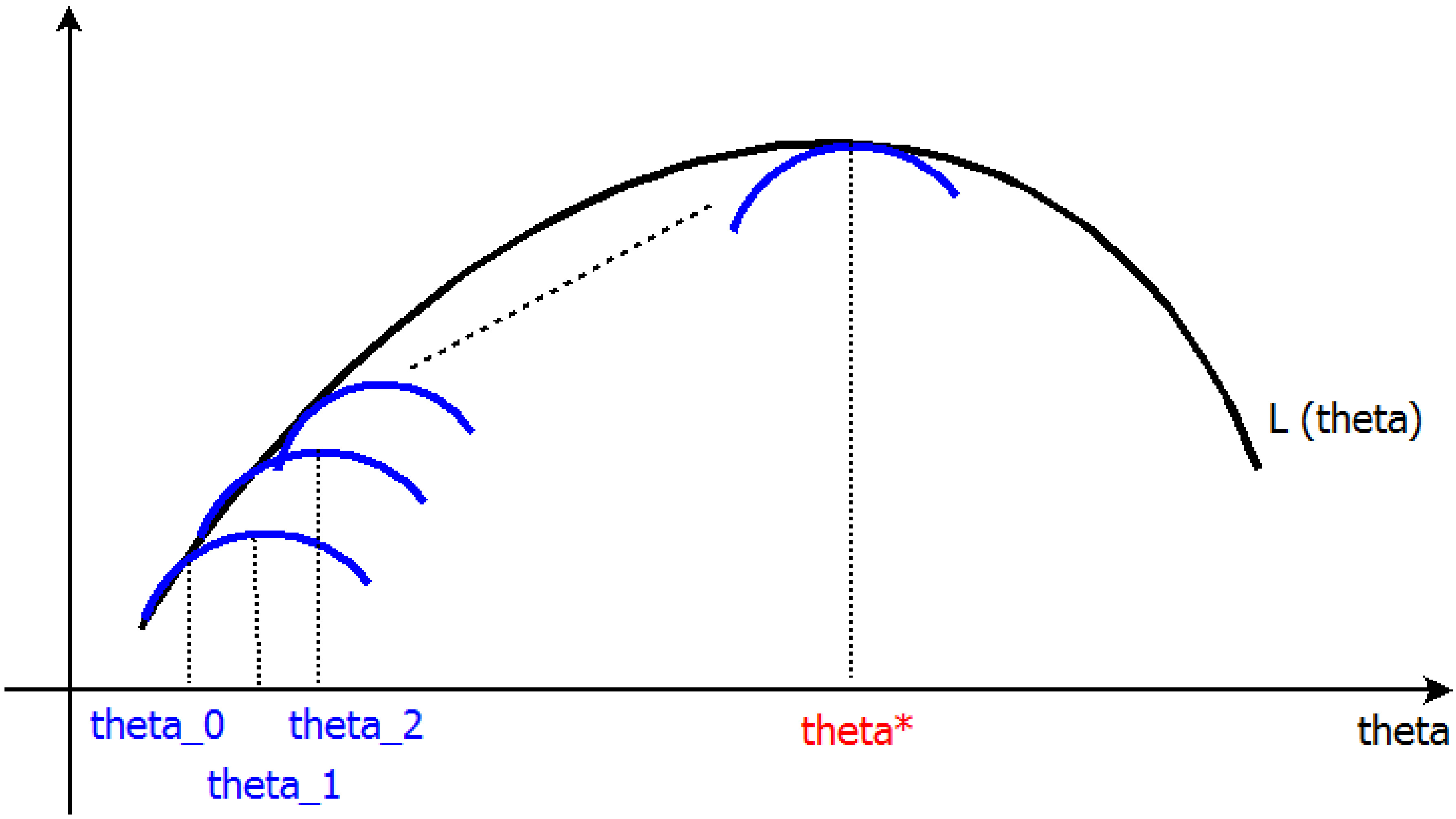
- But we do not observe z, so computing

$$\ell_c(\theta; D) = \log \sum_z p(x, z \mid \theta) = \log \sum_z p(z \mid \theta_z)\, p(x \mid z, \theta_x)$$

  is difficult!

- What shall we do?

Intuition behind the EM algorithm



Jensen's Inequality

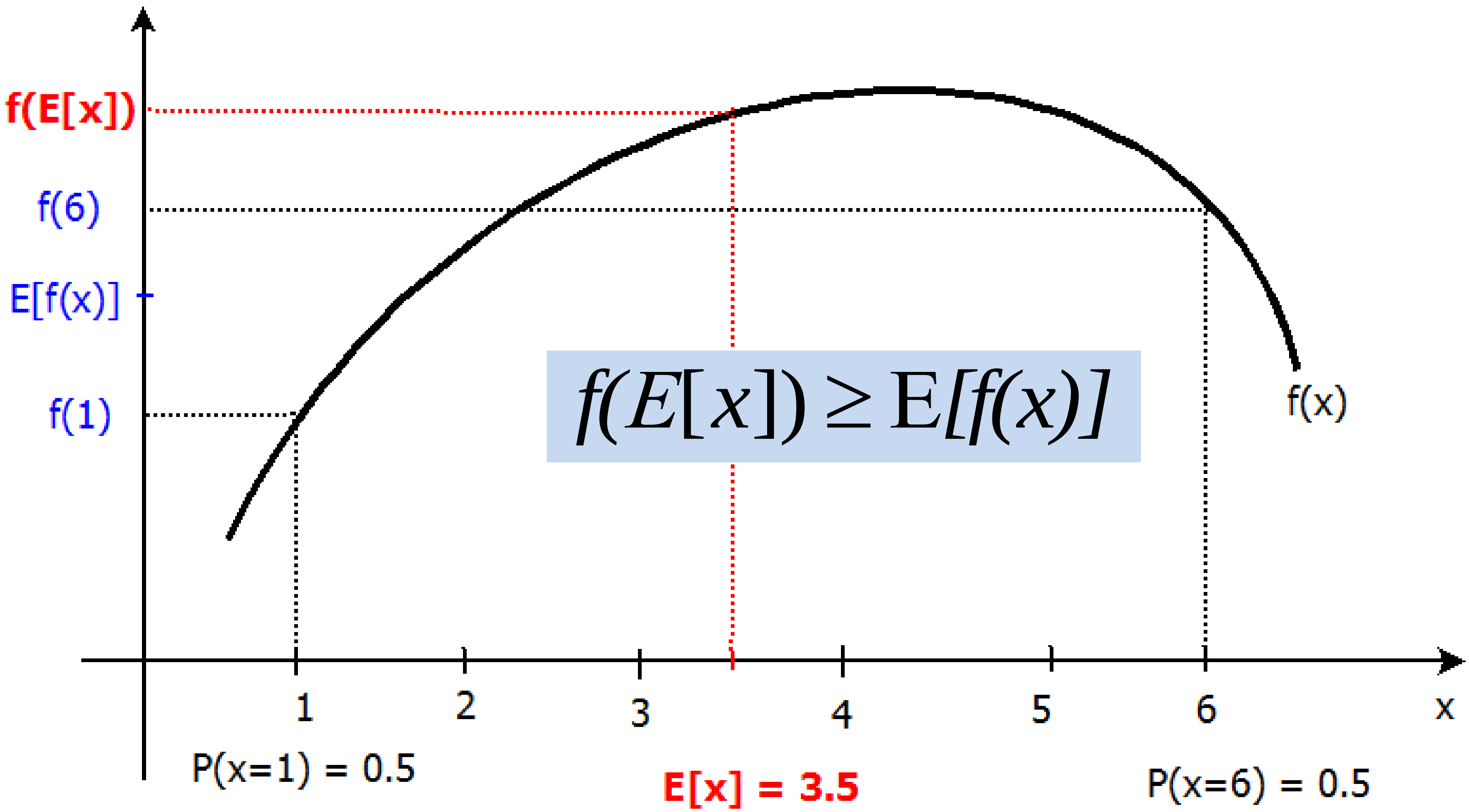
- For a convex function f(x)

$$f(E[x]) \leq \mathrm{E}[f(x)]$$

- Similarly, for a concave function f(x)

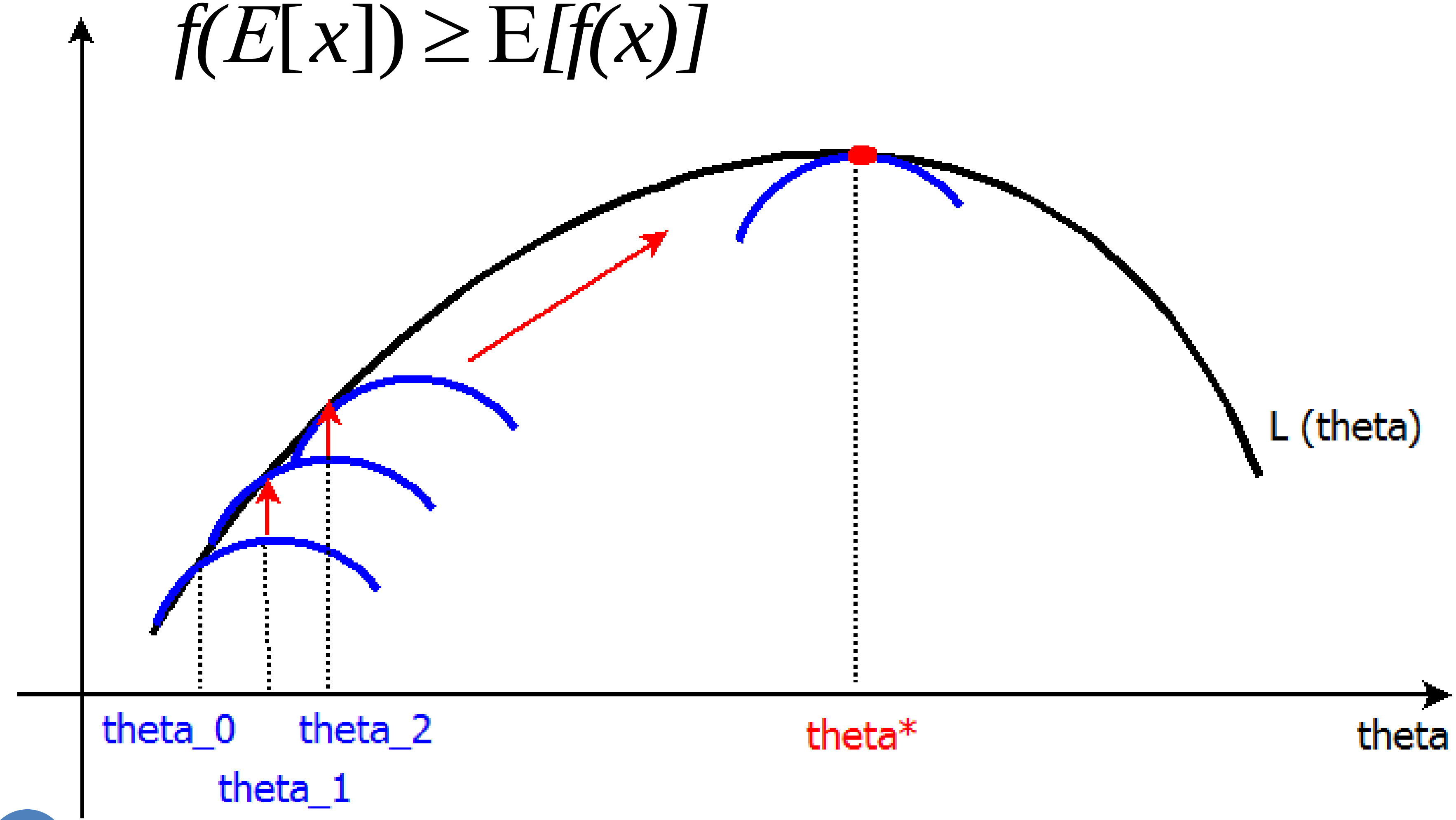
$$f(E[x]) \geq \mathrm{E}[f(x)]$$

Jensen’s Inequality: concave f(x)



EM and Jensen's Inequality

$$f(E[x]) \geq \mathrm{E}[f(x)]$$



Advanced Topics

How Many Clusters?

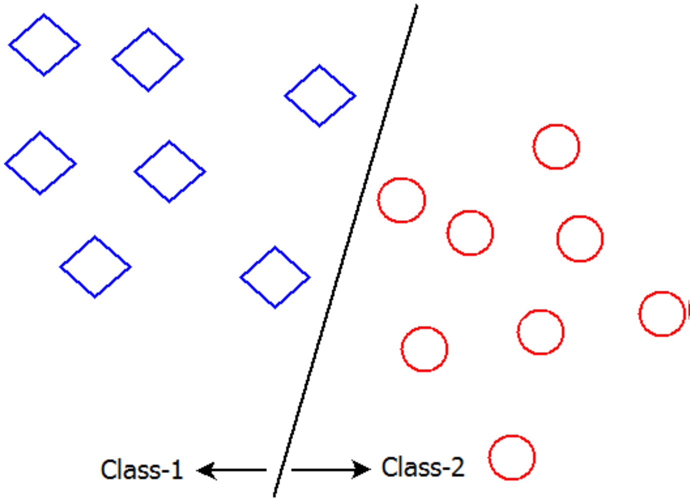
- Number of clusters K is given
  - Partition 'n' documents into predetermined #topics
- Solve an optimization problem: penalize #clusters
  - Information theoretic approaches: AIC, BIC criteria for model selection
  - Tradeoff between having clearly separable clusters and having too many clusters

Seed Choice: K-Means++

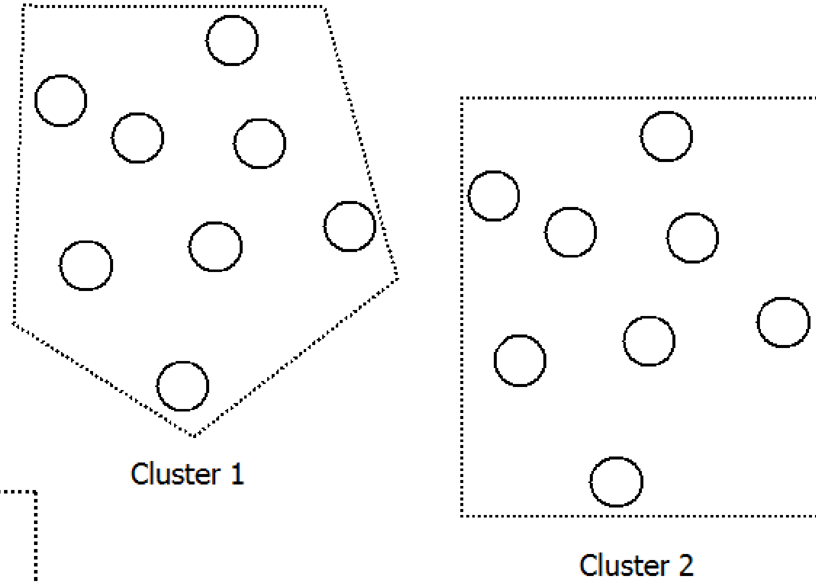
- K-Means results can vary based on random seed selection.
- K-Means++
  - Choose one center uniformly at random among given datapoints.
  - For each data point x, compute $D(x)$
    $D(x) = \text{distance}(x, \ \text{nearest center})$
  - Choose one new data point at random as a new center
    $P(x) \propto D(x)^2$.
  - Repeat Steps 2 and 3 until k centers have been chosen.
- Run standard K-Means with this centroid initialization.

Semi-supervised K-Means

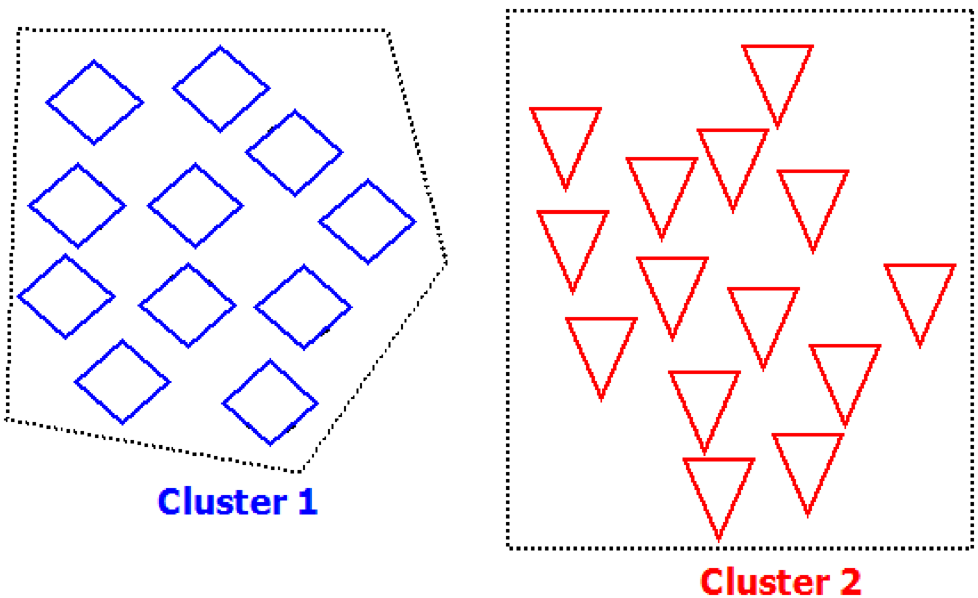
Supervised Learning



Unsupervised Learning



Semi-supervised Learning



Automatic Gloss Finding for a Knowledge Base

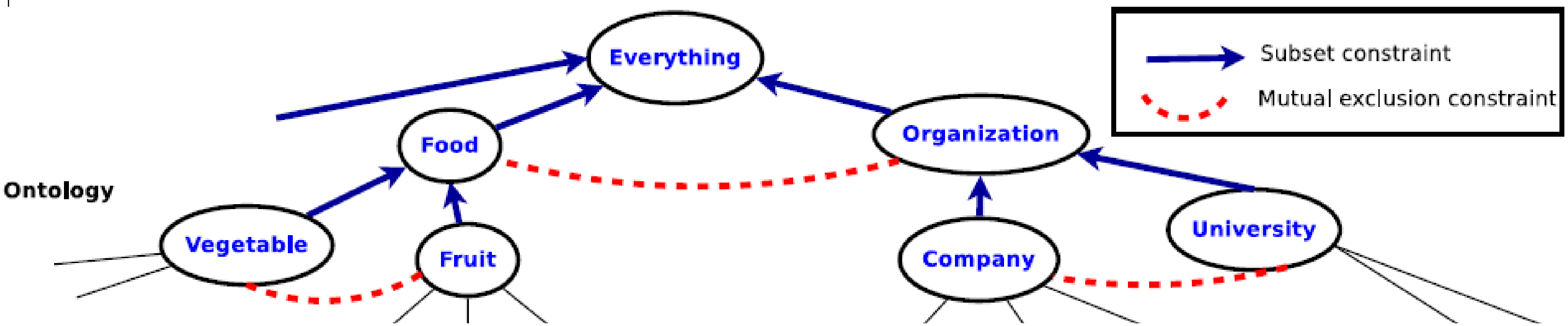
- **Glosses:** Natural language definitions of named entities.

  *E.g.“**Microsoft**” is an American multinational corporation headquartered in Redmond that develops, manufactures, licenses, supports and sells computer software, consumer electronics and personal computers and services …*

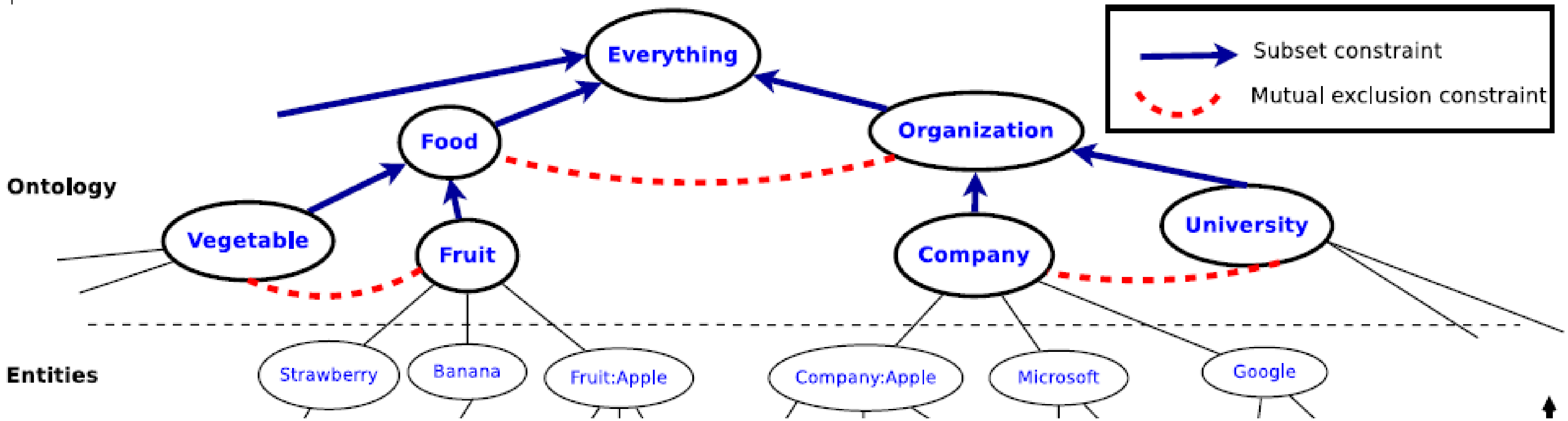
- **Input:** Knowledge Base i.e. a set of concepts (e.g. company) and entities belonging to those concepts (e.g. Microsoft), and a set of potential glosses.
- **Output:** Candidate glosses matched to relevant entities in the KB.
  “Microsoft is an American multinational corporation headquartered in Redmond …”
  is mapped to **entity “Microsoft” of type “Company”**.

*[**Automatic Gloss Finding for a Knowledge Base using Ontological Constraints**, Bhavana Dalvi Mishra, Einat Minkov, Partha Pratim Talukdar, and William W. Cohen, 2014, Under submission]*

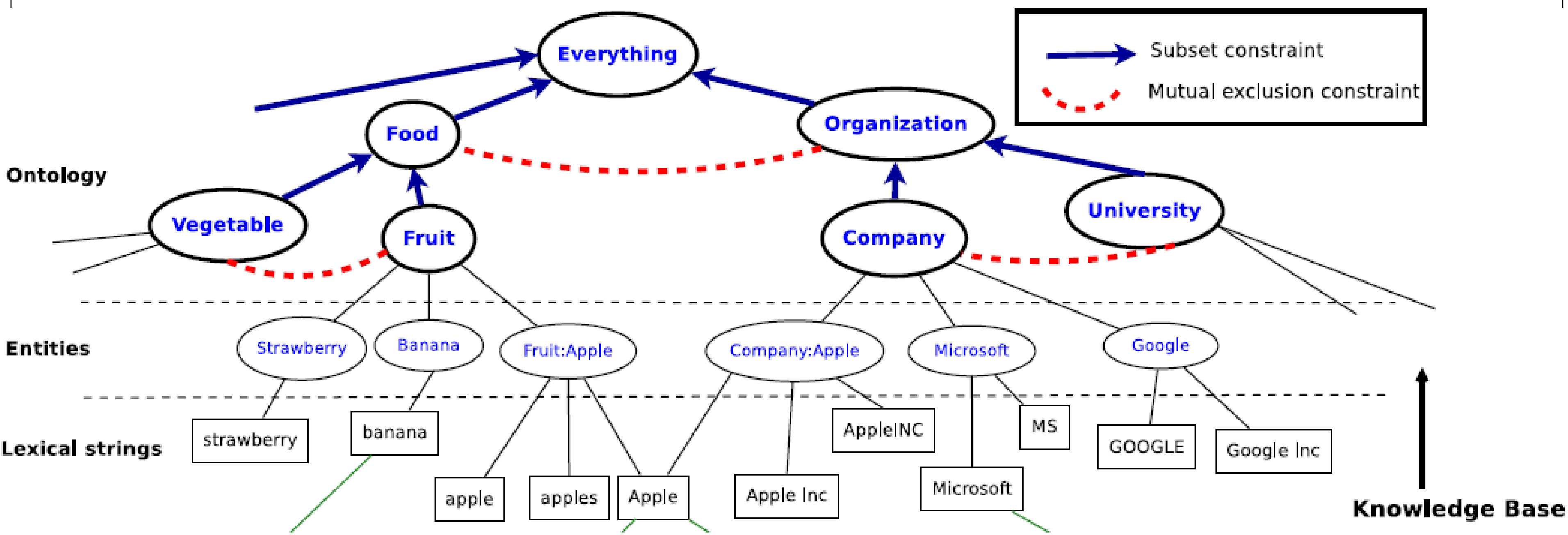
Example: Gloss finding



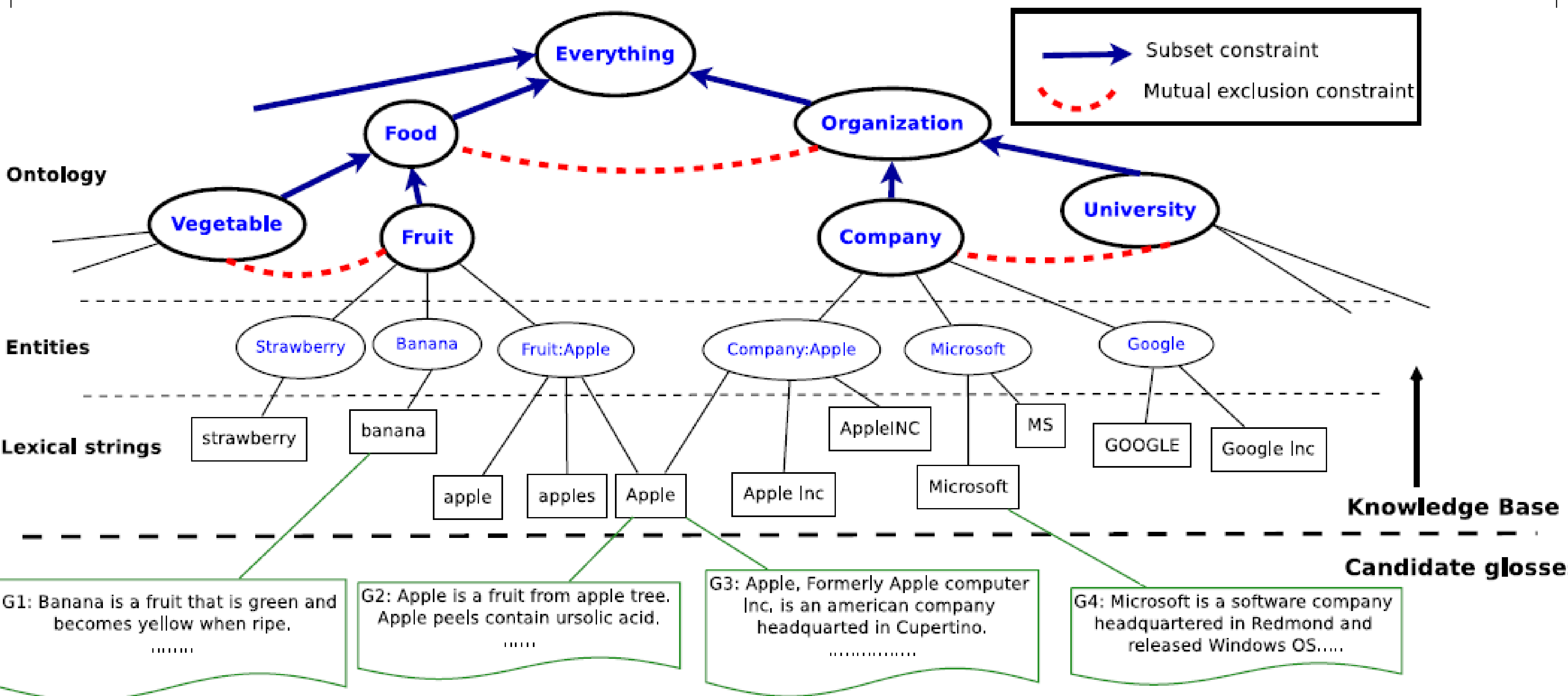
Example: Gloss finding



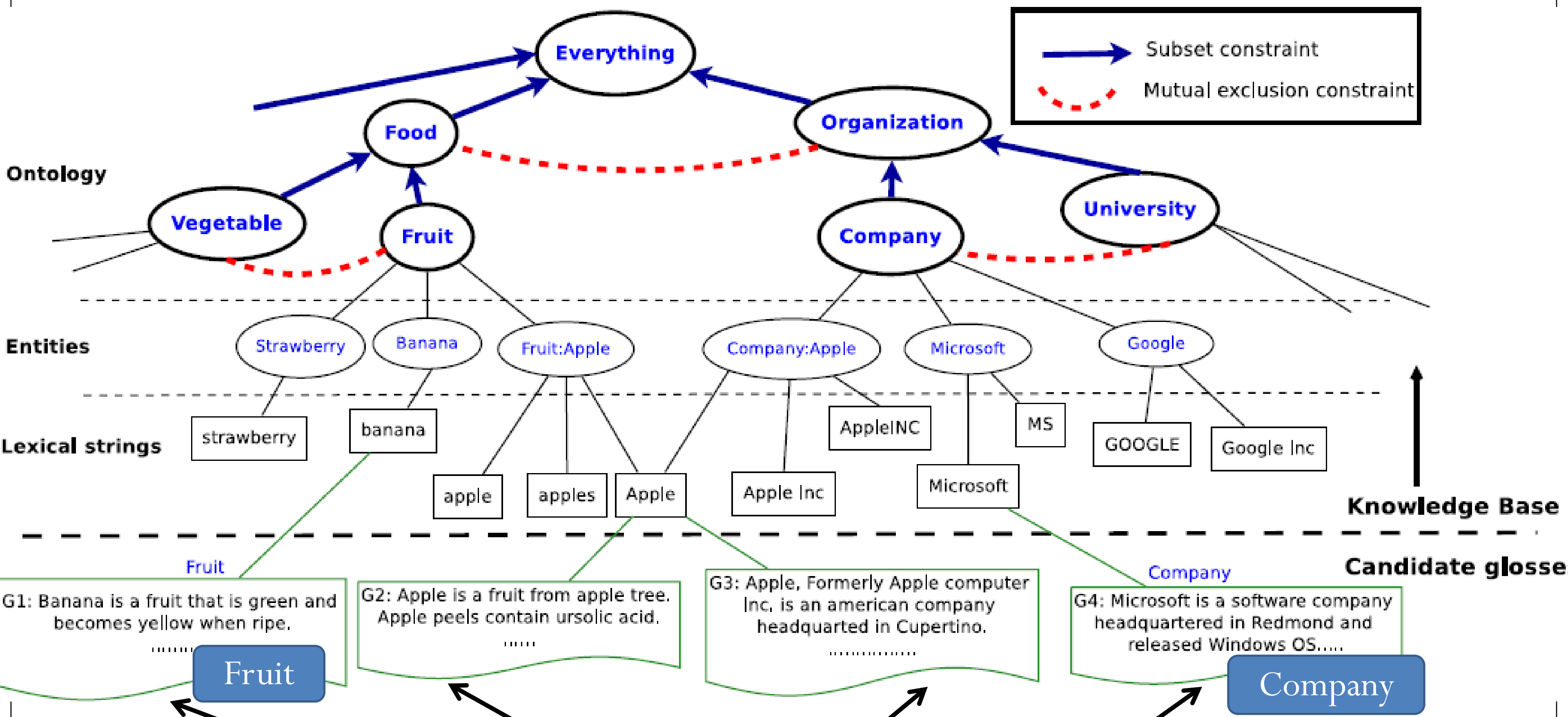
Example: Gloss finding



Example: Gloss finding



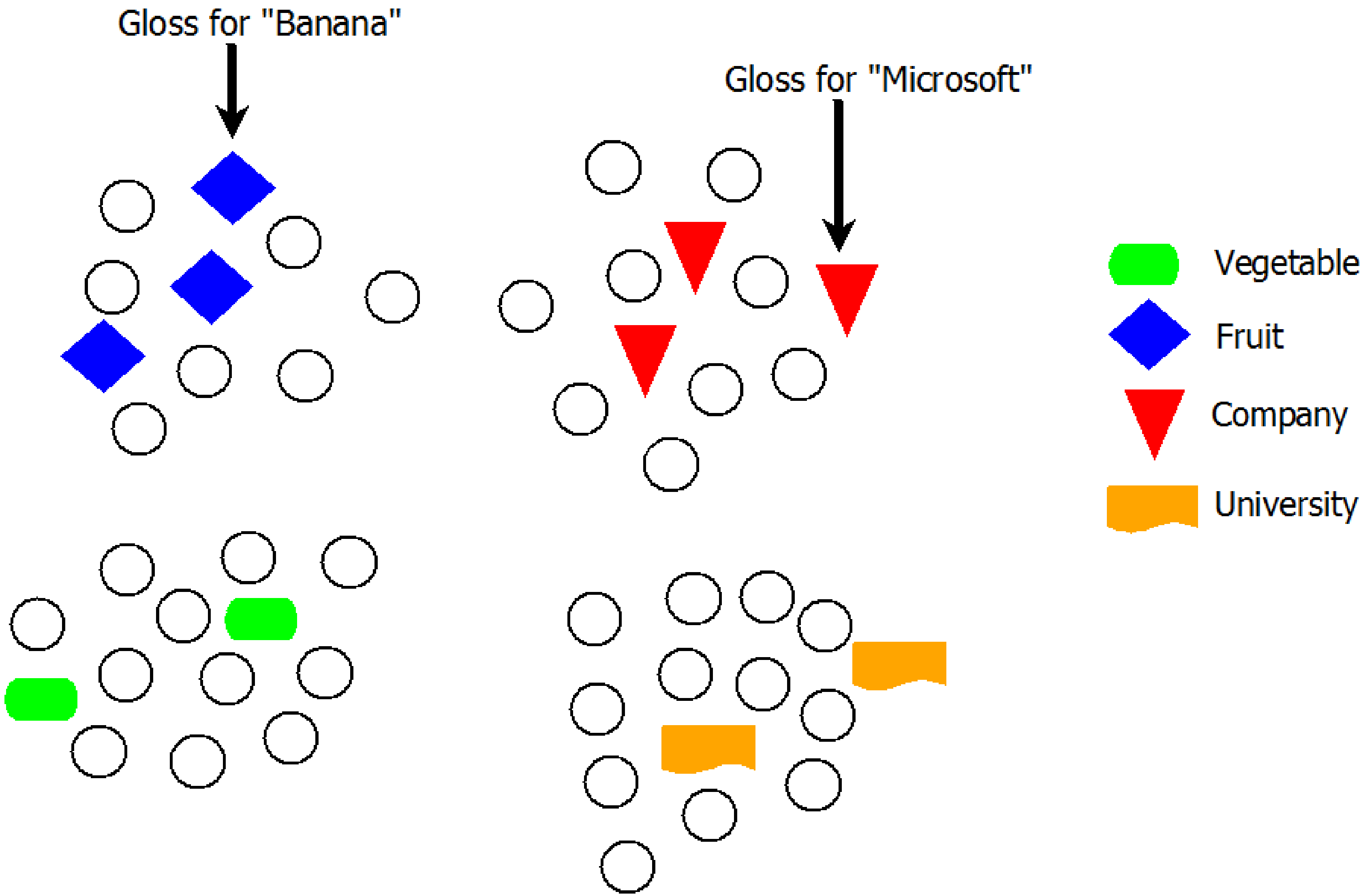
Training a clustering model



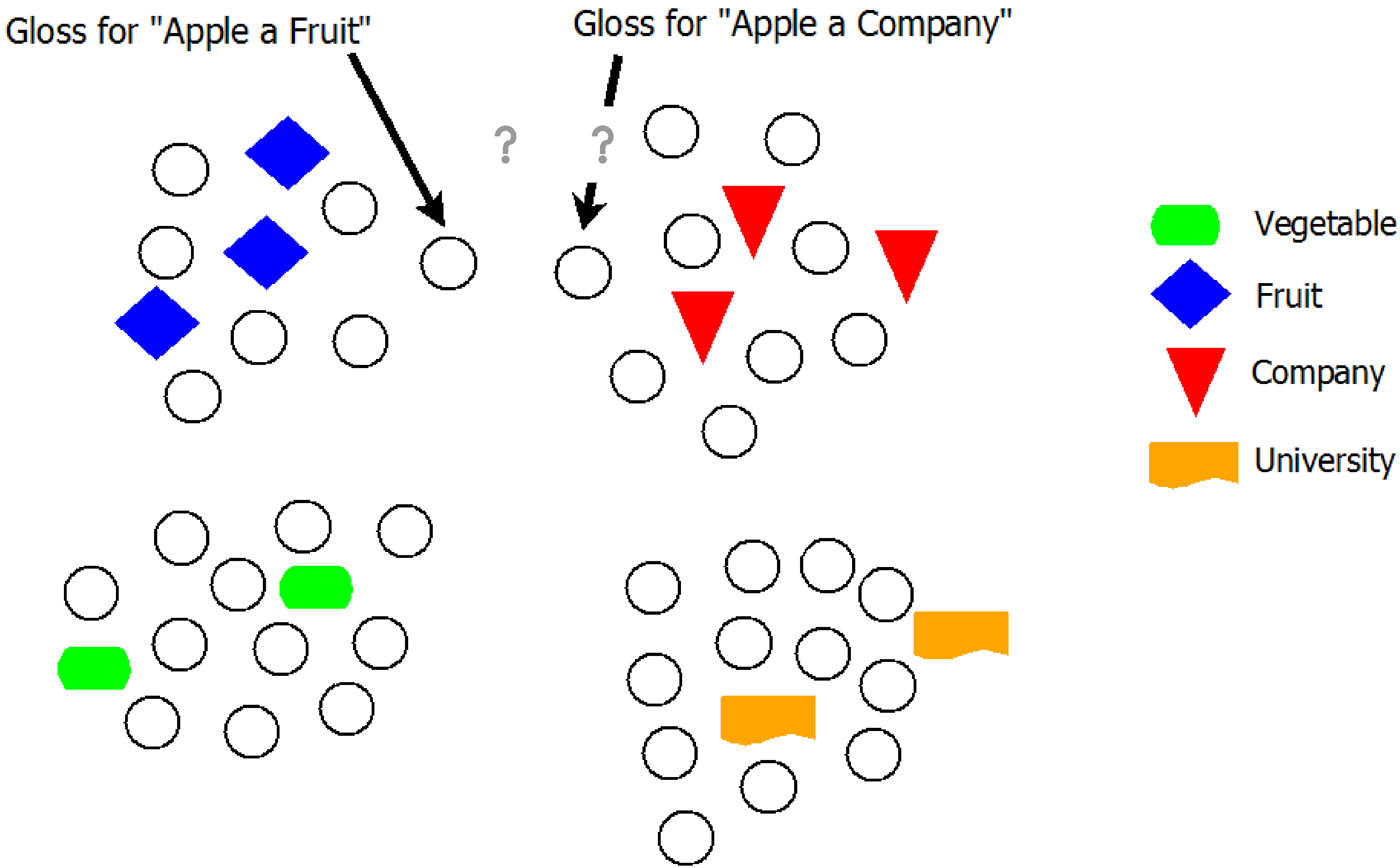
Test: Ambiguous glosses

Train: Unambiguous glosses

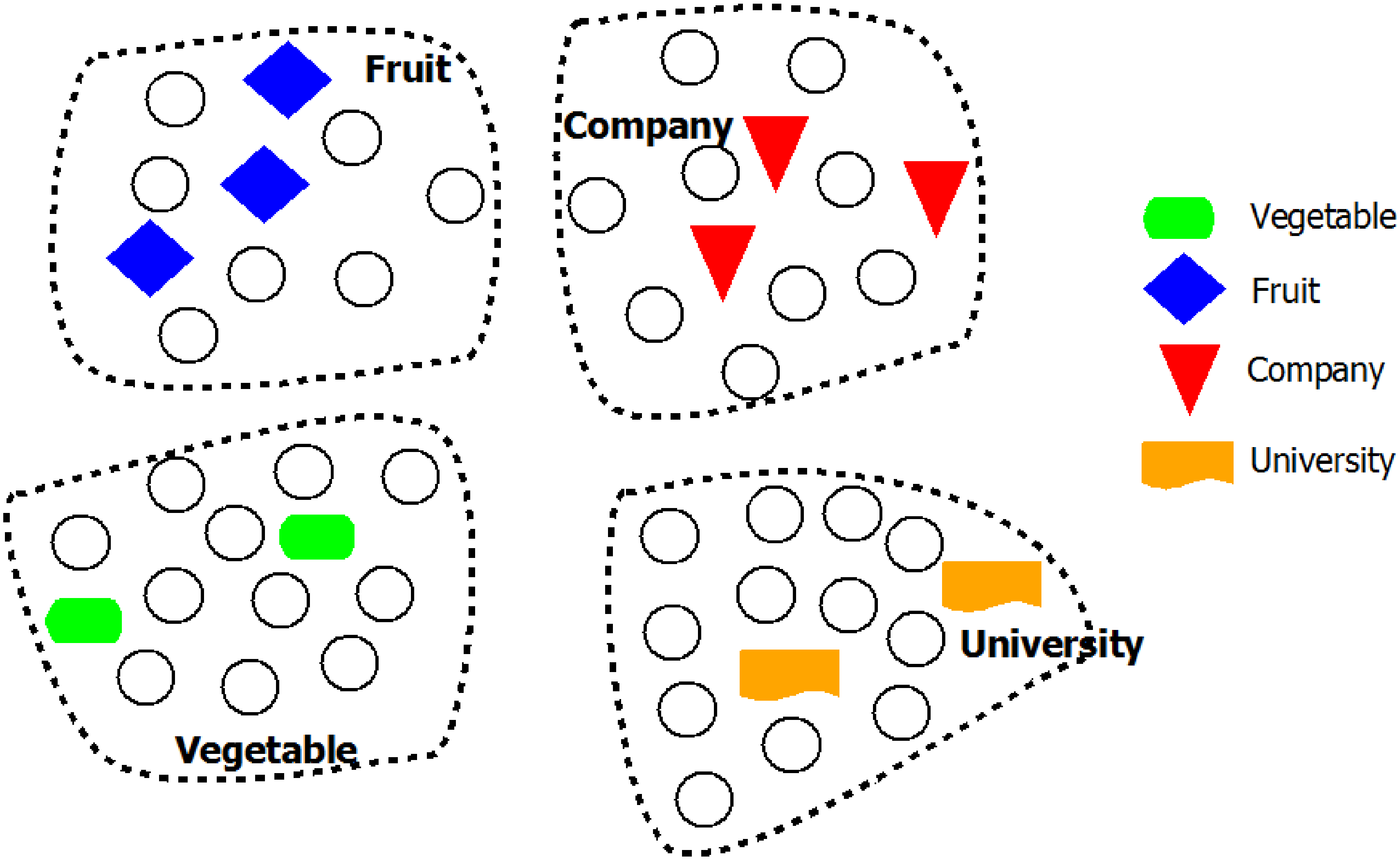
GLOFIN: Clustering glosses



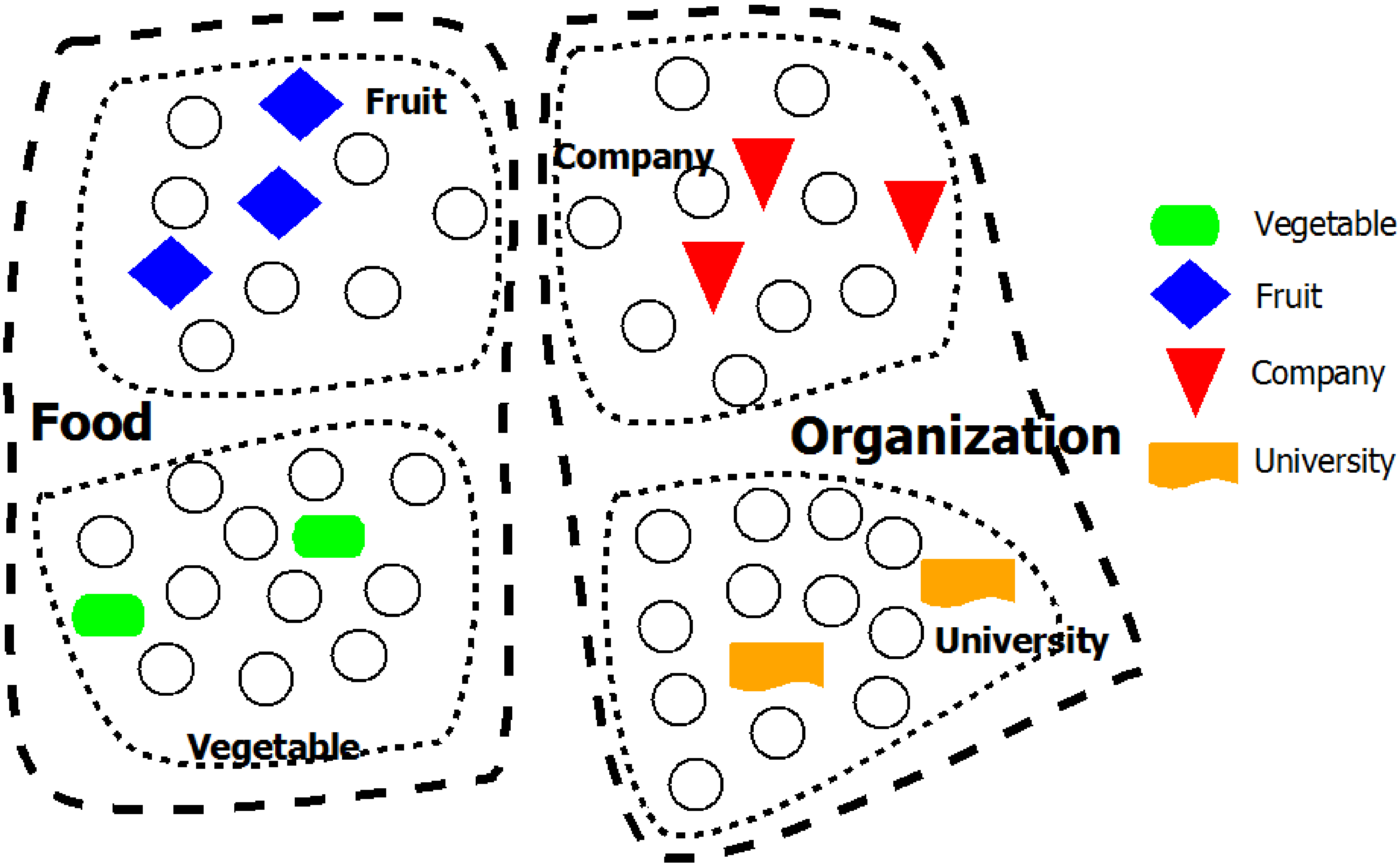
GLOFIN: Clustering glosses



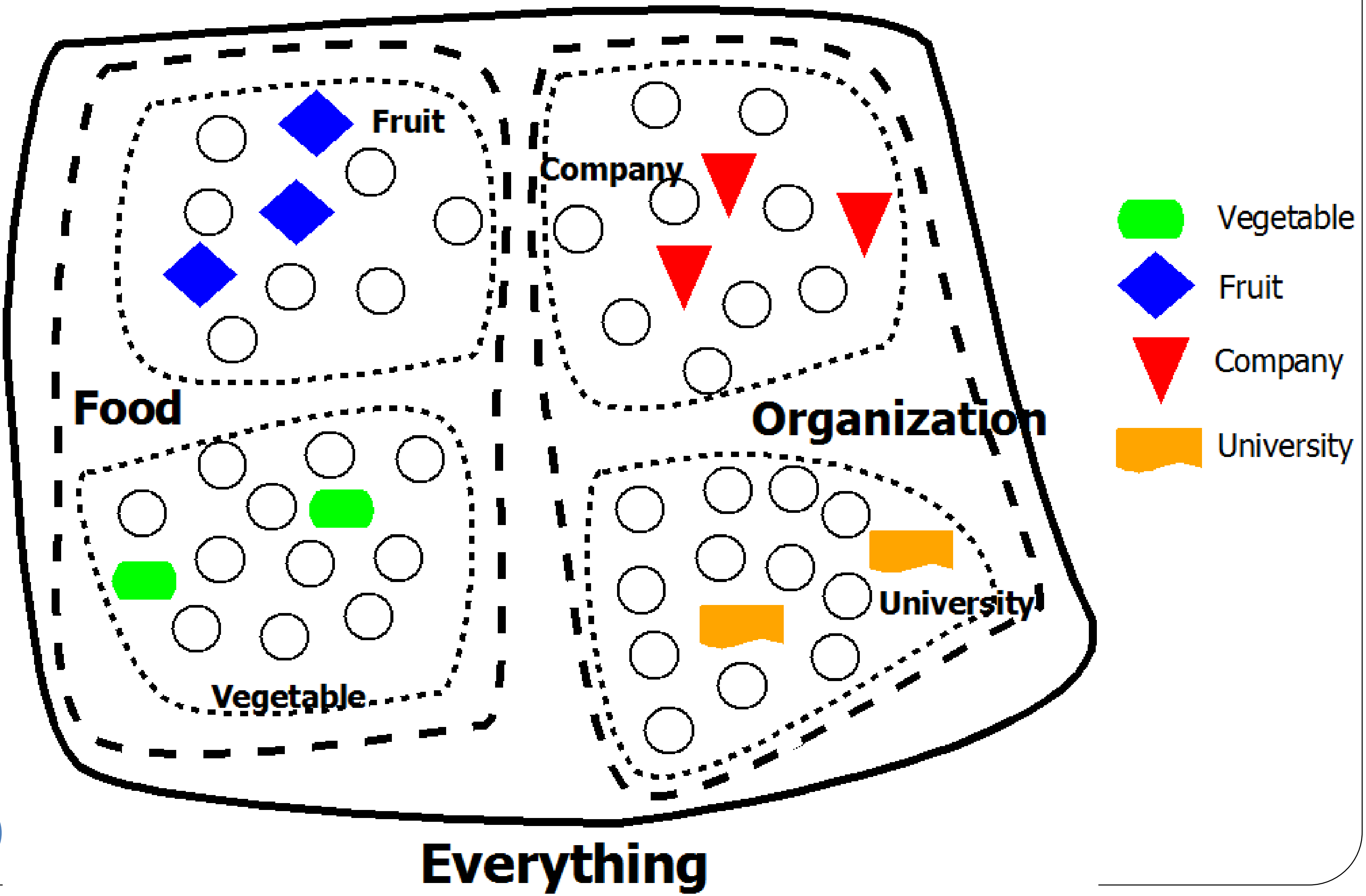
GLOFIN: Clustering glosses



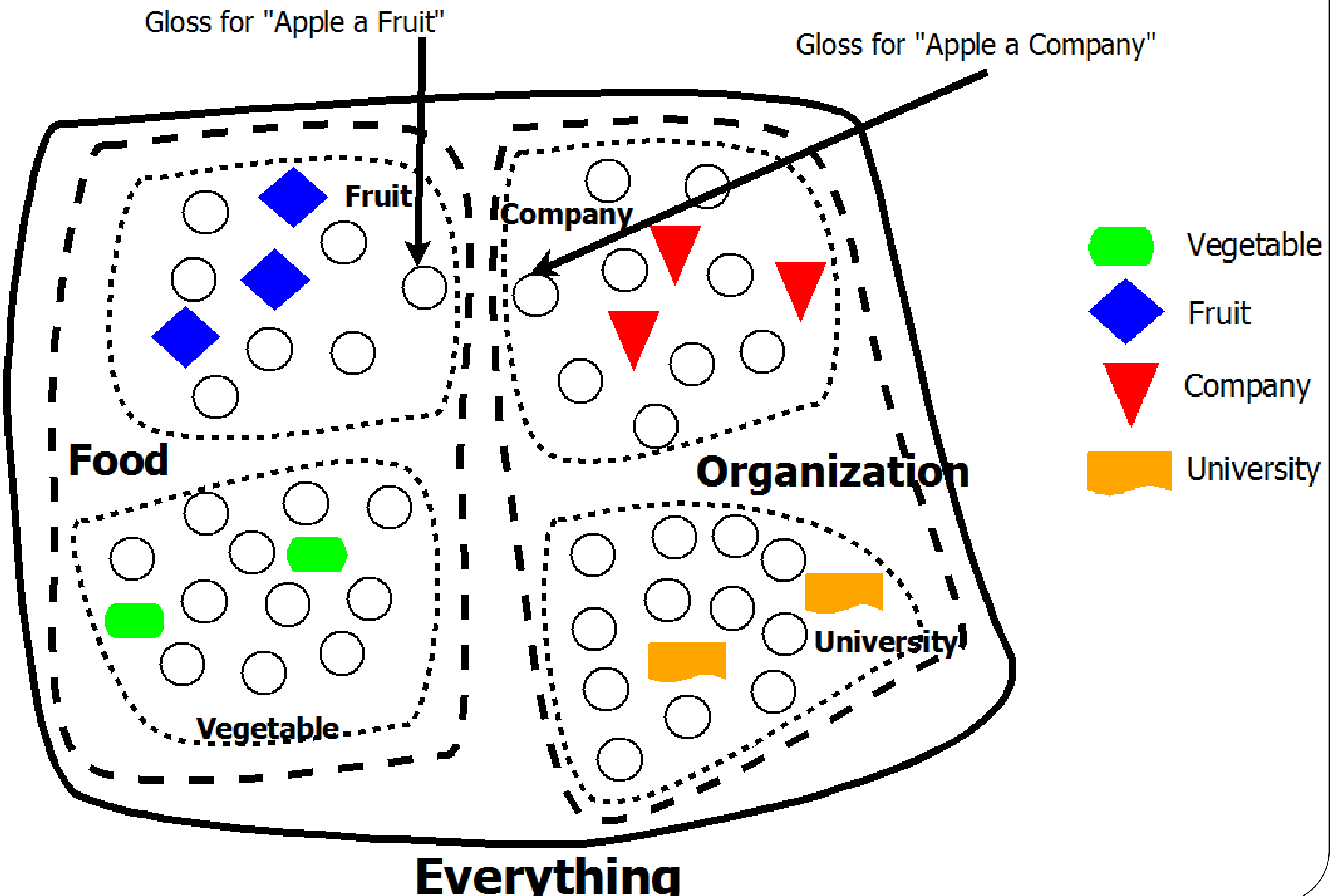
GLOFIN: Clustering glosses



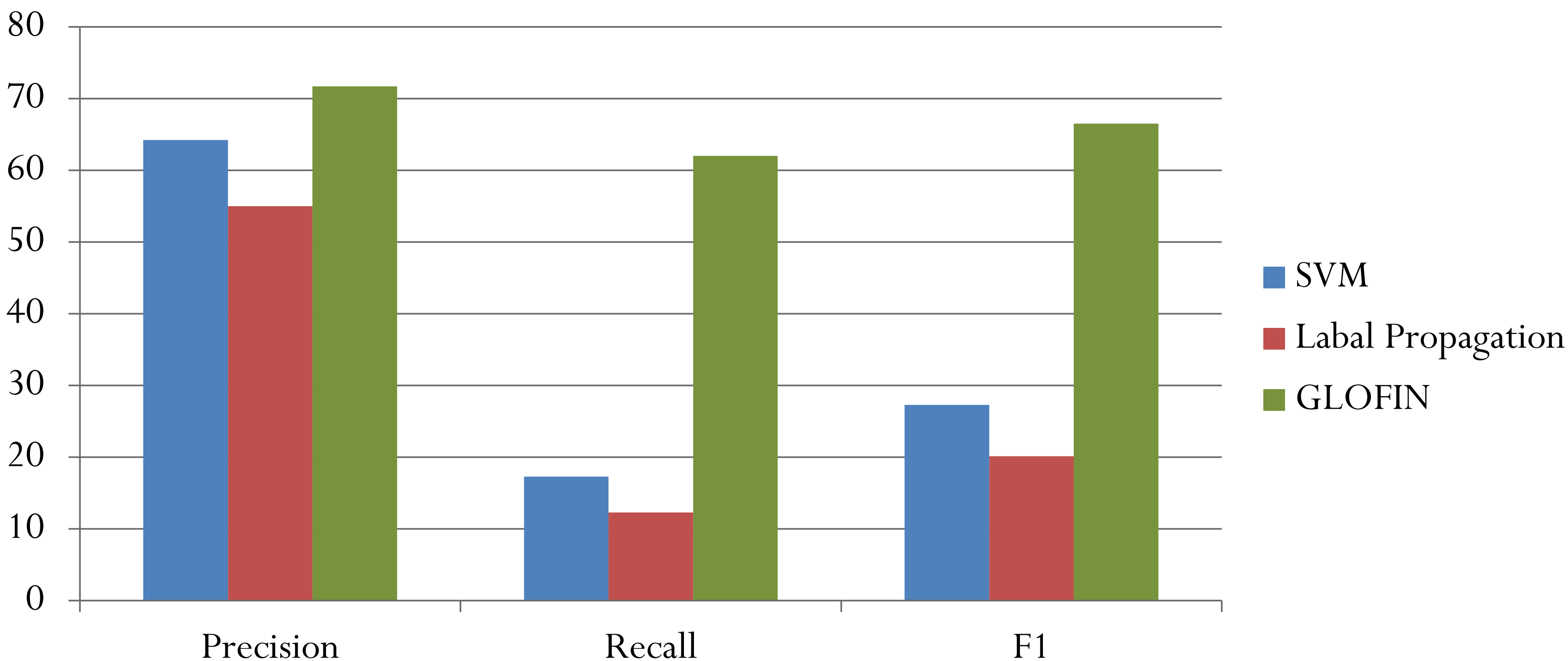
GLOFIN: Clustering glosses



GLOFIN: Clustering glosses

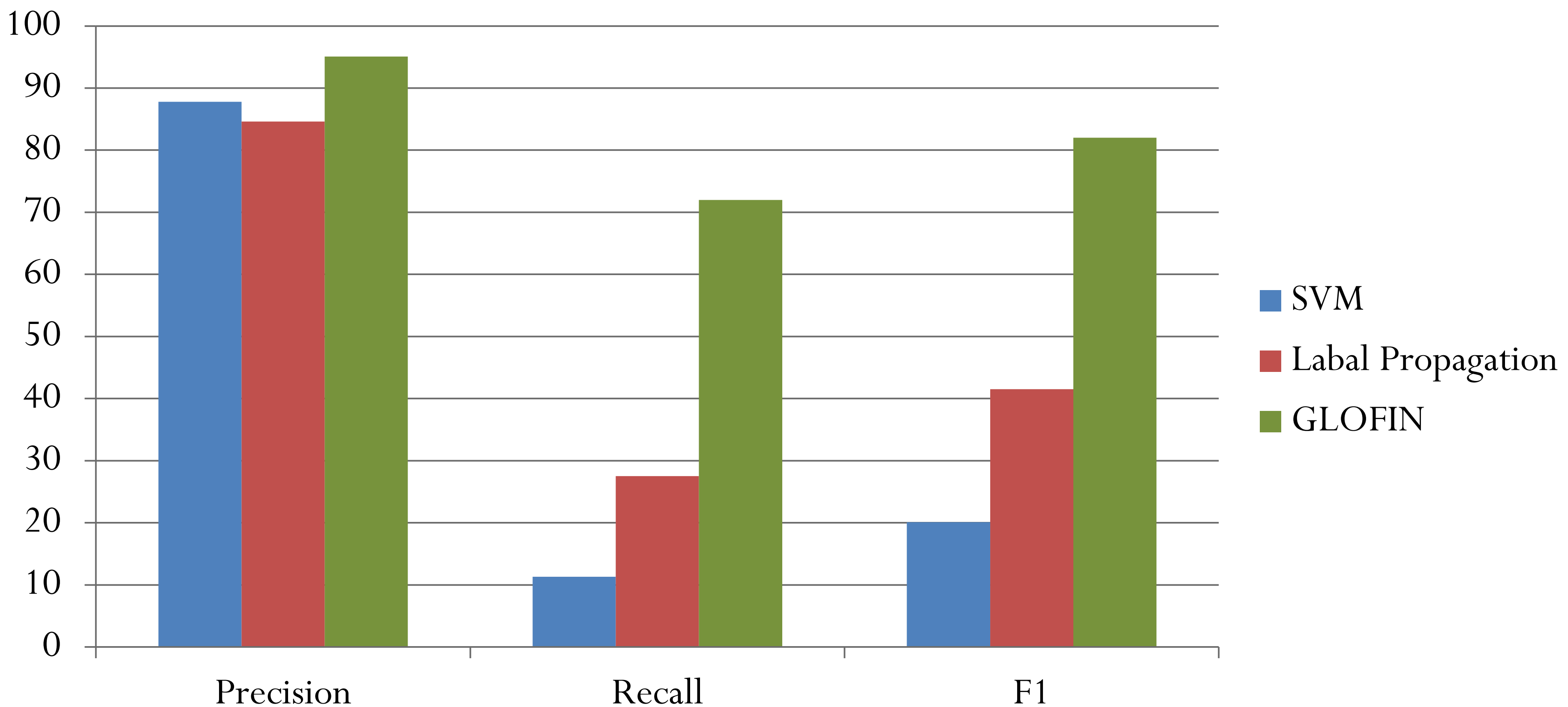


GLOFIN on NELL Dataset



275 categories, 247K candidate glosses, #train=20K, #test=227K

GLOFIN on Freebase Dataset



66 categories, 285K candidate glosses, #train=25K, #test=260K

Summary

- What is clustering?
- What are similarity measures?
- K-Means clustering algorithm
- Mixture of Gaussians (GMM)
- Expectation Maximization
- Advanced Topics
  - How to seed clustering
  - How to decide #clusters
- Application: Gloss finding for a Knowledge Bases

Thank You

Questions?