Language and Statistics II

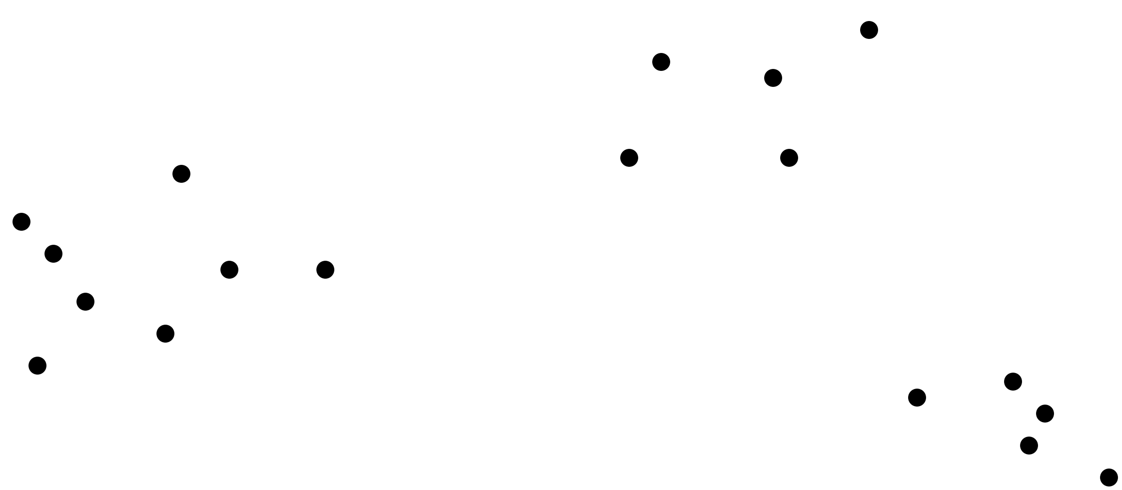
Lecture 18: Clustering

Noah Smith

Clustering

- Given a set of examples, infer classes.
- Class variable has never been observed!
  - So this is **unsupervised** classification.
  - Usual insight: if two examples are very similar, they are probably in the same class.
- In some settings, it's clear how to define the similarity between two examples.
  - But not always (e.g., in NLP).

Clustering $\mathbb{R}$ Data



K-Means

- Given: examples $\{x_i\}$, K

1. Randomly select $m_1, \ldots, m_K$.
2. Assign each x_i to the nearest m_j.

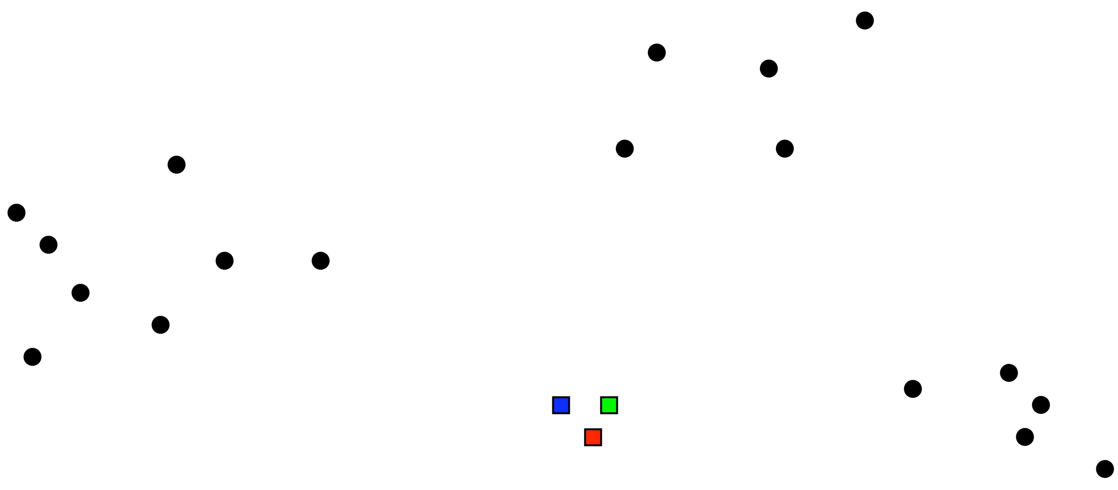
$$\hat{y}_i = \arg\min_{m_j} d\left(x_i, m_j\right)$$

3. Select each m_j to be the mean of all x_i assigned to it.

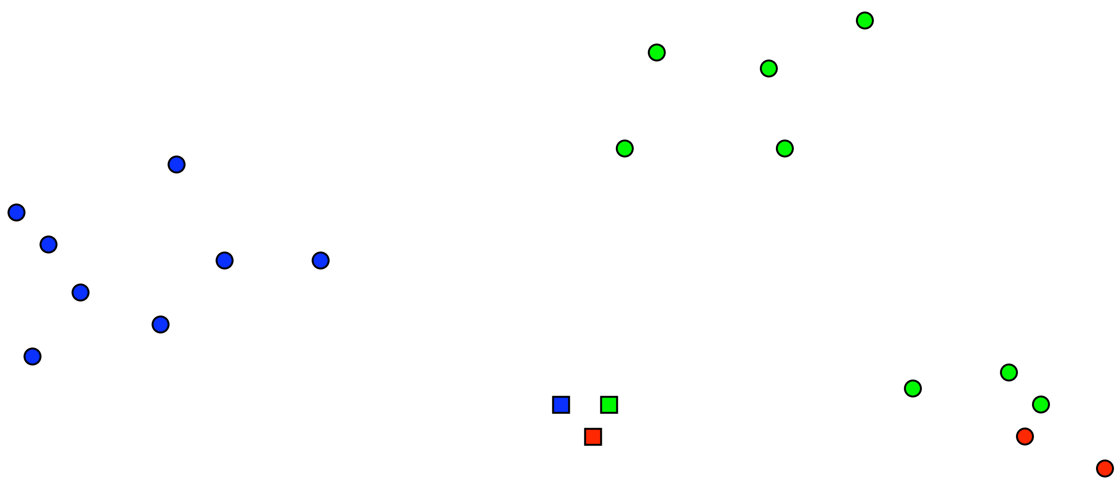
$$m_j = \frac{1}{\left|\left\{i : \hat{y}_i = m_j\right\}\right|} \sum_{i:\hat{y}_i = m_j} x_i$$

4. If all m_j have converged stop; else go to 2.

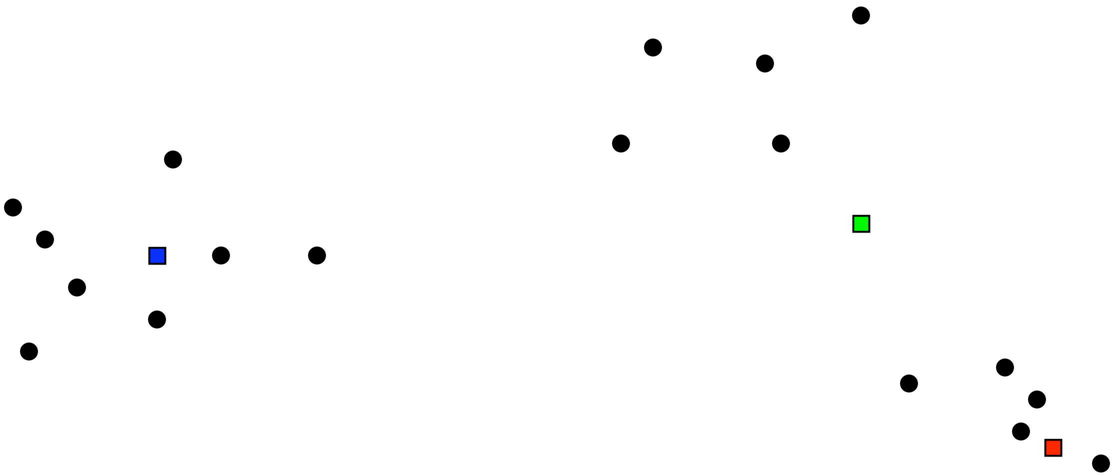
K-Means, Visualized



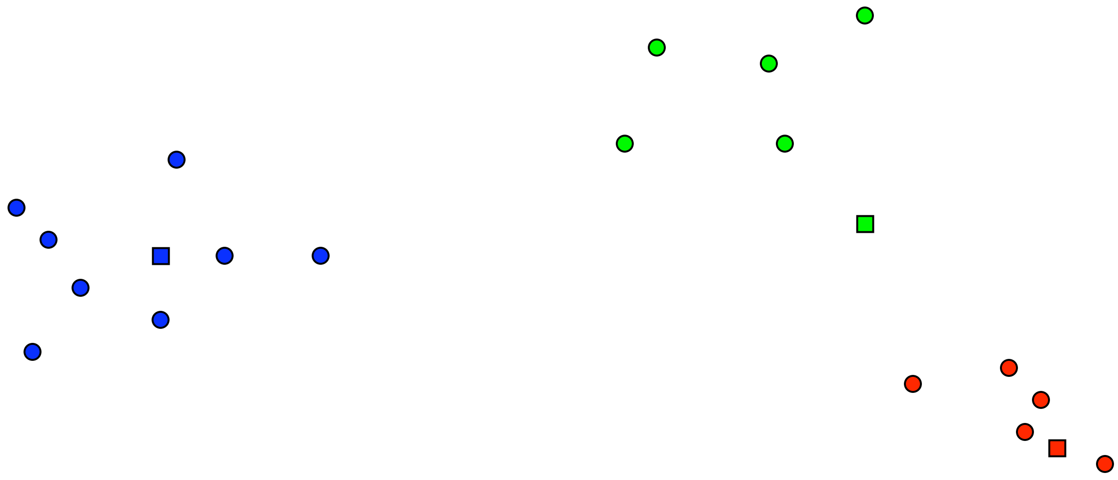
K-Means, Visualized



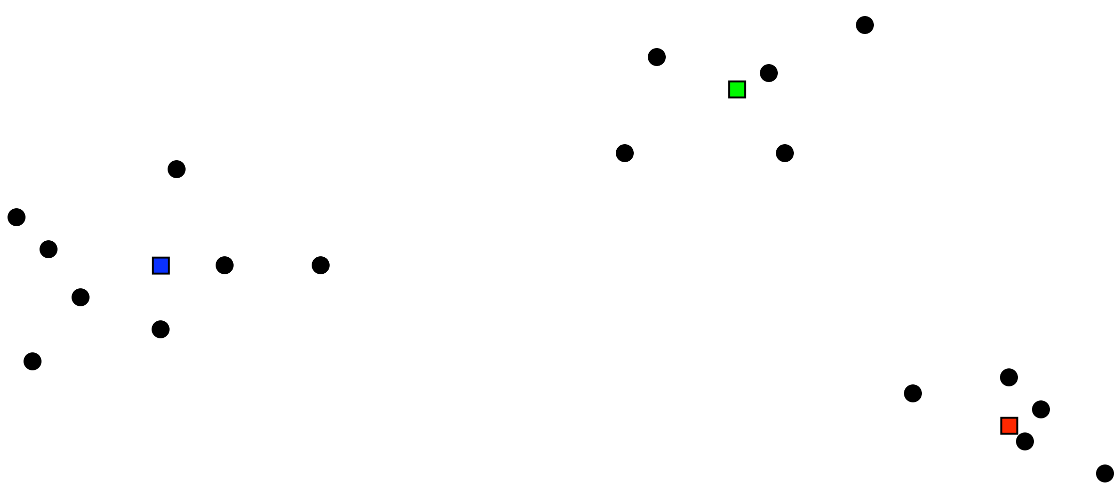
K-Means, Visualized



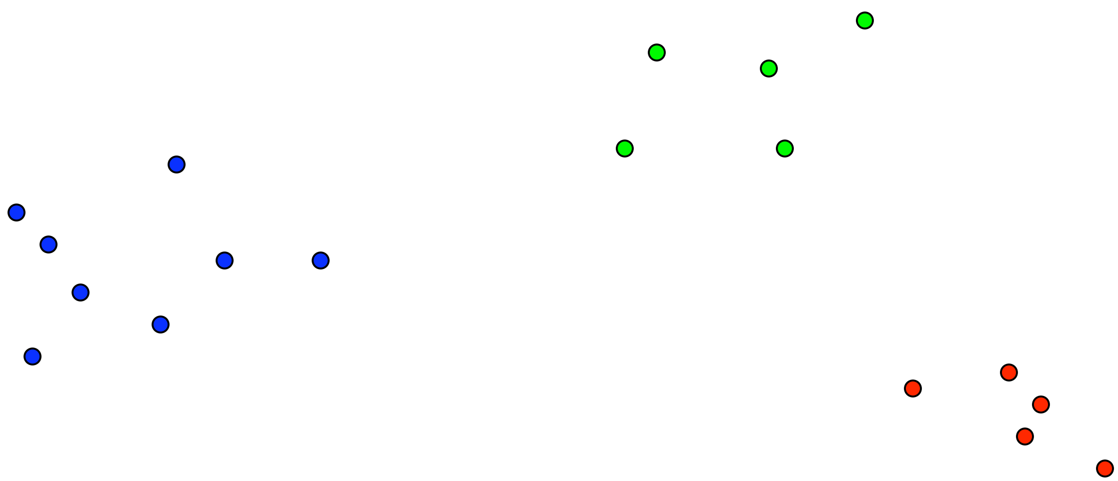
K-Means, Visualized



K-Means, Visualized



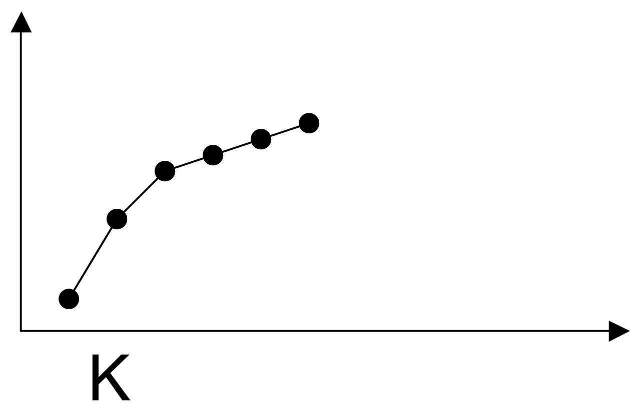
K-Means, Visualized



Questions

- How to choose K?

Try different K; choose the smallest K such that adding another cluster will not explan much variance.



Questions

- How to choose K?
- Does the choice of distance measure matter?
  - Yes!
- Guaranteed to converge?
  - Yes.
- Always to same centroids?
  - No.
- Is there an objective function that is being optimized?
  - Yes (locally).
- Does this have a probabilistic interpretation?
  - Yes.

From K-Means to EM

- Soft K-Means … add a parameter β.

Each x_i gets one vote, which it divides between clusters.

$$V_j(x_i) = \frac{\exp\left[-\beta d\left(x_i, m_j\right)\right]}{\sum_{j'} \exp\left[-\beta d\left(x_i, m_{j'}\right)\right]}$$

portion of x_i's vote going to m_j

Cluster m_j is chosen by a vote among all x_i.

$$m_j = \frac{\sum_i x_i V_j(x_i)}{\sum_i V_j(x_i)}$$

weighted average of x_i (by their votes)

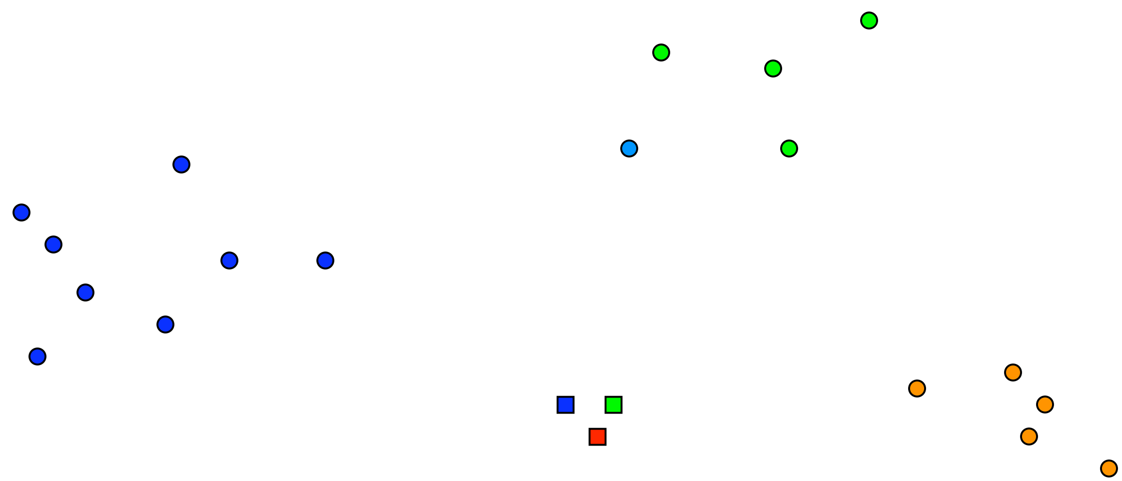
From K-Means to EM

- Soft K-Means … add a parameter β.
  - β is "stiffness" - it controls how much variance the clusters can have.
  - β → ∞ approaches hard K-Means!

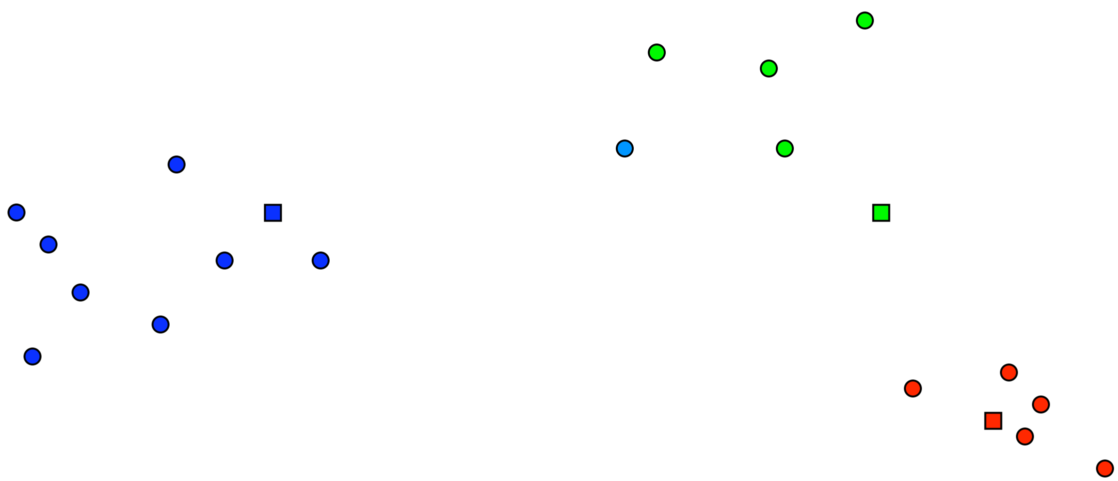
$$V_j(x_i) = \frac{\exp\left[-\beta d\left(x_i, m_j\right)\right]}{\sum_{j'} \exp\left[-\beta d\left(x_i, m_{j'}\right)\right]} \xrightarrow{\beta \to \infty} \begin{cases} 1 & \text{if } m_j = \arg\min_m d\left(x_i, m\right) \\ 0 & \text{otherwise} \end{cases}$$

$$m_j = \frac{\sum_i x_i V_j(x_i)}{\sum_i V_j(x_i)}$$

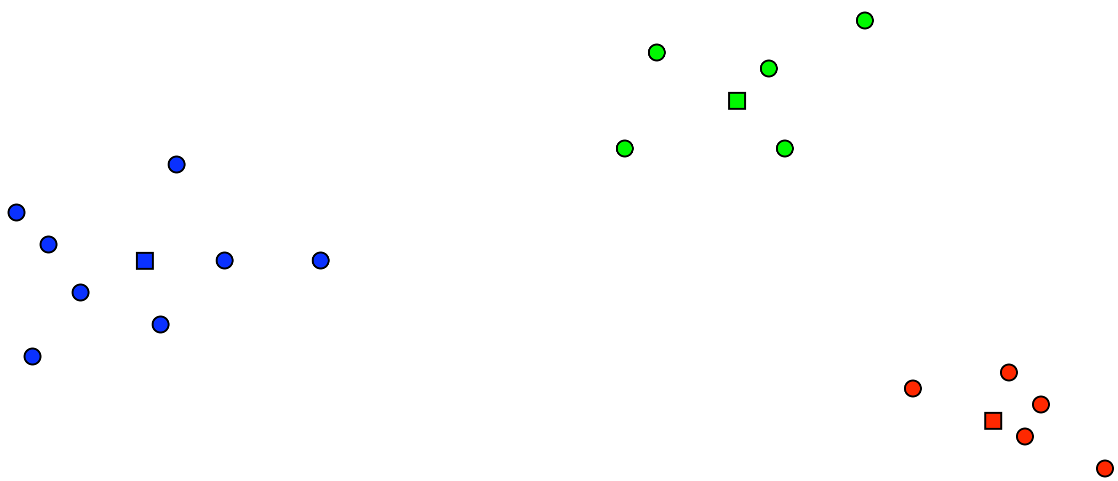
Soft K-Means, Visualized



Soft K-Means, Visualized



Soft K-Means, Visualized



From K-Means to EM

- Soft K-Means … add a parameter β.
  - β is "stiffness" - it controls how much variance the clusters can have.
  - $\beta \to \infty$ approaches hard K-Means!

- Claim: this is the EM algorithm, for a particular log-linear model!

$$p(X = x, M = m) \propto \exp\left[-\beta d\left(x, m\right)\right]$$

From K-Means to EM

- If d(x, y) is squared Euclidean distance, clusters are equiprobable *a priori*, all clusters have same variance, and $\beta = 2\sigma^2$ …

$$p(X = x, M = m) = p(x|m)\,p(m) = \frac{1}{K}\,p(x|m)$$

$$= \frac{1}{K\sqrt{|\Sigma|(2\pi)^D}}\exp\left(-\frac{1}{2}(x - m)'\,\Sigma^{-1}(x - m)\right)$$

$$\propto \exp\left(-\beta(x - m)^2\right)$$

$$p(X = x, M = m) \propto \exp\left[-\beta d(x, m)\right]$$

What is this EM?

- EM is many things.
  - Class of alternating minimization algorithms
  - Likelihood maximization technique for hidden variables (like clusters)
  - Approximate inference technique

- For now, think of it as a soft clustering method with two alternating steps:
  - E (expectation or "election") step
  - M (maximization or "model-fitting") step

E (Election) Step

- Each example x_i decides how much of its vote to give to each cluster.
- To allocate x_i's vote, consider the **posterior** probability that x_i came from m_j:

$$q(m_j | x_i) \propto e^{-\beta d(x_i, m_j)}$$

  - The closer m_j is, the more of x_i's vote it gets.
- For squared Euclidean distance, you can tell this generative story:
  - Pick a centroid j uniformly.
  - Sample X according to a Gaussian at mean m_j.

M (Model-Fitting) Step

- Each cluster conforms to its constituents!
- I.e., given a set of (possibly fractional) examples, carry out MLE for m_j:

$$\hat{m}_j = \arg\max_m \prod_{i=1}^{n} p(x_i|m)^{\overbrace{q(m_j|x_i)}^{\text{fractional count of } x_i}} = \arg\max_m \sum_{i=1}^{n} q(m_j|x_i) \log p(x_i|m)$$

$$= \arg\max_m \sum_{i=1}^{n} q(m_j|x_i) \log e^{-\beta d(x_i,m)}$$

$$= \arg\min_m \sum_{i=1}^{n} q(m_j|x_i) d(x_i,m)$$

Another View of EM

- If we knew the m_j, we could say how strongly each x_i belongs to each m_j. (Easily!)

- If we knew how strongly each x_i belongs to each m_j, we could guess where the m_j **are**. (Easily!)

Another View of EM

- If we knew the m_j, we could say how strongly each x_i belongs to each m_j. (Easily!)

This is the E step.

- If we knew how strongly each x_i belongs to each m_j, we could guess where the m_j **are**. (Easily!)

This is the M step.

The Model

- Two random variables: X and Y
- Each x_i is observed (the data)
- Each Y_i is **hidden** or **latent**
- $-d(x, y)$ is a similarity (negative distance) feature
- β is the weight of that feature
- The possible values of the y_j (the possible values for each Y_i) are the model parameters. We know there are K vectors, $m_1, \ldots, m_K$.

(This model really only makes sense in a continuous space where we can take weighted averages!)

In General …

- EM can be applied to any probabilistic model.
  - But it’s much easier to apply to some models than to others!

- There’s always a “winner-take-all” variant.
  - You should think of this as an approximation.

EM in General

- E step:

$$\forall i, y, q\left(y \middle| x_i\right) \leftarrow p_{\vec{\theta}^{(t)}}\left(y \middle| x_i\right) = \frac{p_{\vec{\theta}^{(t)}}\left(x_i, y\right)}{\sum_{y'} p_{\vec{\theta}^{(t)}}\left(x_i, y'\right)}$$

soft assignment or voting

- M step:

$$\vec{\theta}^{(t+1)} \leftarrow \arg\max_{\vec{\theta}} \sum_{x,y} \underbrace{\tilde{p}(x)q\left(y \middle| x\right)}_{\text{"pretend" } \tilde{p}(x,y)} \log p_{\vec{\theta}}(x, y)$$

fully-observed data MLE

Aside: EM ≈ Gibbs Sampling

- Alternative view: we have two hidden variables, Θ (the parameters) and Y.
- Randomized approach to inference: sample each hidden variable in turn, given all the others.
  - Sample Y given X, Θ. (E step: exact inference)
  - Sample Θ given X, Y. (M step: take the mode)

Claims

- EM is trying to maximize the likelihood of the data.
  - The observed part: $\{x_i\}$
  - The hidden part, Y, is marginalized over.
- EM converges to a **local** optimum.
  - Which local optimum depends on the initial parameters (or posterior).
  - EM can take many iterations to converge.

Clustering Words

- Brown et al. (1992)
- Pereira et al. (1993)
- Schütze (1993)

Brown et al., 1992

- Motivation: improved language modeling.
- Class-based language model:

$$p\left(s_i \middle| s_{i-m} \ldots s_{i-1}\right) = p\left(s_i \middle| c_i\right) p\left(c_i \middle| c_{i-m} \ldots c_{i-1}\right)$$

- Classes are **hard clusters**.
- Greedy search algorithm …

Brown et al., 1992

- Input: vocabulary of V words, K

1. Initialize with each word in its own class.
2. For t = 1 to V - K:
    1. Compute the average mutual information between each class pair.
    2. Merge the class pair that will result in the smallest loss in average mutual information.

*Some implementation tricks required!

Average Mutual Information

- Likelihood of the data:

$$\frac{1}{N}\sum_{i=1}^{N}\log\left(p\left(w_i \middle| c_i\right)p\left(c_i \middle| c_{i-1}\right)\right) = \mathbf{E}_{\tilde{p}}\left[\log\left(p\left(W \middle| C\right)p\left(C \middle| C'\right)\right)\right]$$

$$= \mathbf{E}_{\tilde{p}}\left[\log\left(\frac{p\left(W \middle| C\right)p\left(C \middle| C'\right)p(C)}{p(C)}\right)\right]$$

$$= \mathbf{E}_{\tilde{p}}\left[\log\left(\frac{p\left(C \middle| C'\right)}{p(C)}\right) + \log\left(p\left(W \middle| C\right)p\left(C\right)\right)\right]$$

$$= \mathbf{E}_{\tilde{p}}\left[\log\left(\frac{p\left(C',C\right)}{p\left(C'\right)p\left(C\right)}\right) + \log\left(p\left(W\right)\right)\right]$$

Comparison

K-Means	Brown et al., 1992
• Hard classes	• Hard classes
• Distance feature (similarity model)	• Bigram features (bigram class model)
• Fixed # classes K	• # classes: $V \rightarrow K$
• Winner-take-all EM (optimize “extreme” likelihood)	• Greedy search based on MI (optimize likelihood)

Both can be seen as trying to optimize likelihood.

Pereira et al., 1993

*Warning: this is a very confusing paper because it introduces lots of new ideas.

- **Soft** clustering of **nouns** based on the **verbs** that take them as objects.
- The model: $$p(v,n) = \sum_c p(c)\,p(v|c)\,p(n|c)$$
- Like in K-Means, there is a distance feature: it is the KL divergence between two distributions:

$$d(n,c) = D\left(\tilde{p}(V|n) \,\|\, p(V|c)\right)$$

- Unlike the other methods discussed so far, K is not fixed. It starts at 1, and they gradually increase it by **splitting** clusters.
- To make this happen, they manipulate β …

Deterministic Annealing and Phase Transitions

- Recall:

$$q\left(m_j \middle| x_i\right) \propto e^{-\beta d\left(x_i, m_j\right)} \qquad q\left(c \middle| n\right) \propto e^{-\beta d(n,c)}$$

- When β is close to 0, every noun is in every cluster with about the same strength.
- As β increases, model commits more.
- Can think of β as a Lagrange multiplier controlling the entropy of the posterior! $F = E_{p(C|N)}\left[d(N,C)\right] - \frac{1}{\beta} H\left(p\left(C \middle| N\right)\right)$

Deterministic Annealing and Phase Transitions

- Recall:

$$q\left(m_j \middle| x_i\right) \propto e^{-\beta d\left(x_i, m_j\right)} \qquad\qquad q\left(c \middle| n\right) \propto e^{-\beta d\left(n, c\right)}$$

- When β is close to 0, every noun is in every cluster with about the same strength.
- As β increases, model commits more.
- Can think of β as a Lagrange multiplier controlling the entropy of the posterior! $F = E_{p(C|N)}\left[d(N,C)\right] - \frac{1}{\beta} H\left(p\left(C \middle| N\right)\right)$
- Physical analogy: β = 1/temperature.
  - At high temperatures, the system is equally likely to be in any state.
  - As system cools (β gets large), system commits to one state.
  - Goal of annealing in metalworking is to **find** a stable configuration (low free energy).

Deterministic Annealing and Phase Transitions

- Recall:

$$q(m_j|x_i) \propto e^{-\beta d(x_i, m_j)} \qquad q(c|n) \propto e^{-\beta d(n,c)}$$

- When β is close to 0, every noun is in every cluster with about the same strength.
- As β increases, model commits more
- Can think of β as a Lagrange multiplier controlling the entropy of the posterior! $F = E_{p(C|N)}\left[d(N,C)\right] - \frac{1}{\beta} H\left(p(C|N)\right)$

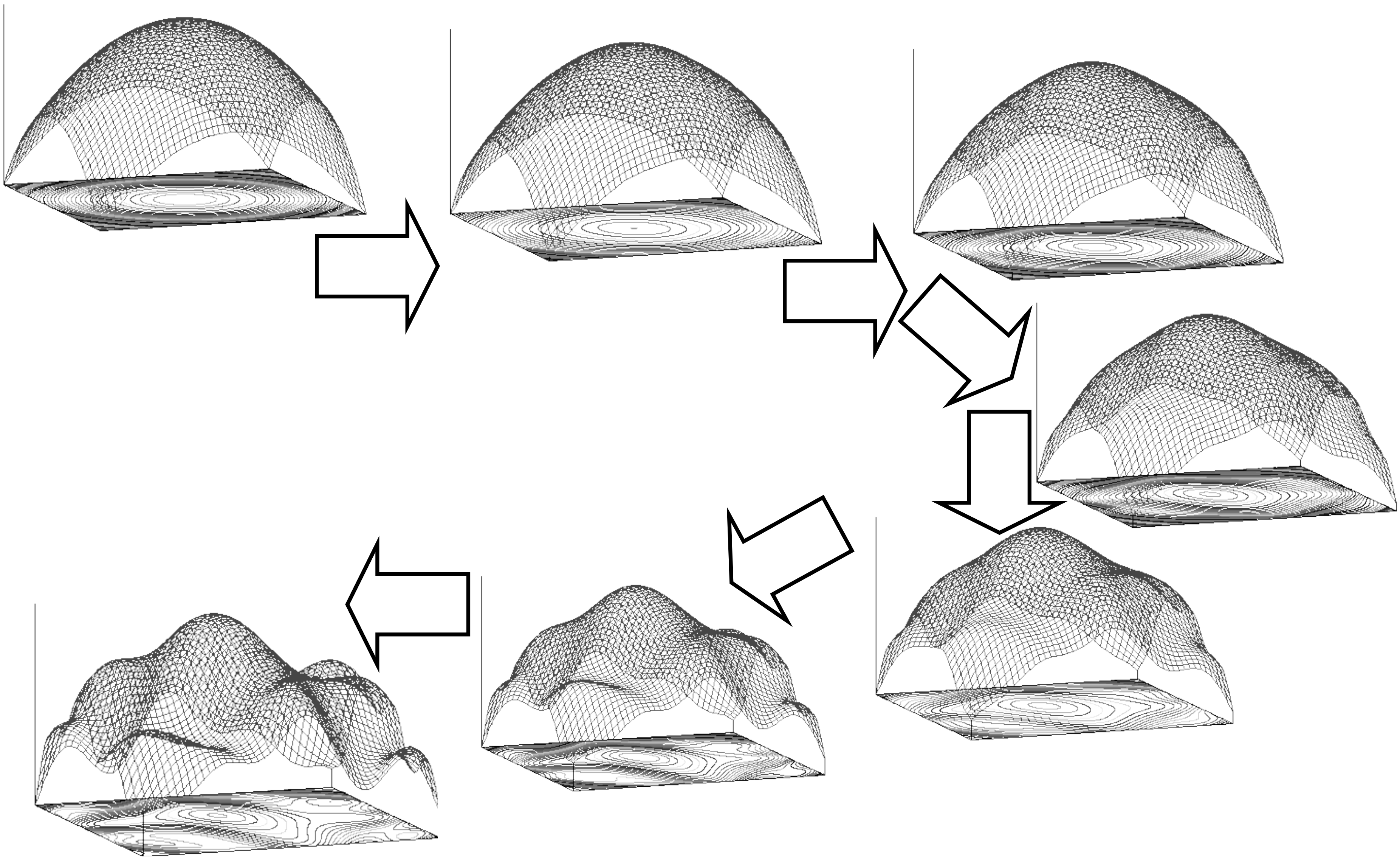
Phase transitions are the effect of gradually increasing β.

DA Clustering

- Start out with two clusters: c and its twin, c.t, and set β to be close to zero.
- Iteratively re-estimate the cluster centroids, gradually increasing β.
  - Whenever a cluster c and its twin c.t become sufficiently distinct (in terms of distance from each other), **split** c.t into a new cluster c', and give c and c' new twins (slight perturbations).

Note: can extract a hierarchical clustering from this! How?

The Objective Function View



Comparison

K-Means	Brown et al., 1992	Pereira et al., 1993
• Hard classes • Distance feature (similarity model) • Fixed # classes K • Winner-take-all EM (optimize "extreme" likelihood)	• Hard classes • Bigram features (bigram class model) • # classes: V → K • Greedy search based on MI (optimize likelihood)	• Soft classes • Distributional similarity feature • # classes: 1 → K • DA/EM search (optimize likelihood)

All three can be seen as trying to optimize likelihood.

Schütze (1993)

- Map words into high-dimensional $\mathbb{R}$ vector of coocurrence counts (-2, -1, +1, +2).
- Singular value decomposition to reduce dimensionality
- Didn't work well for ambiguous words; used a neural network to do classification *in context*.
- See paper for more details.

Next Time

- EM-based unsupervised learning with models of discrete structures (sequences and trees).