Clustering 2

Nov 6 2009
Flat vs hierarchical clustering

- Hierarchical clustering generates nested clusterings
- Flat clustering generates a single partition of the data without resorting to the hierarchical procedure
- Representative example for flat clustering
  - K-means clustering
Mean-Squared Error Objective

• Given a set of examples, each represented as a real-valued vector
• Given a partition of the data into $k$ clusters, we can compute the centroid (i.e., mean or center of mass) of each cluster

$$\mu_c = \frac{1}{|c|} \sum_{x \in c} x$$

• For a tight and well formed cluster, its points should be close to its centroid. We measure this using mean squared error (MSE) is:

$$J_e = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - \mu_i||^2$$

• This gives us a clustering objective:

$$C^* = \arg\min_{\mathbf{C}=\{c_1, \ldots, c_k\}} \sum_{i=1}^{k} \sum_{x \in c_i} ||x - \mu_i||^2$$

• Note that this is a combinatorial optimization problem
  – Difficult to find the exact solution
  – Kmeans solves this problem using an iterative approach
Kmeans: Basic idea

- We assume that the number of desired clusters, $k$, is given
- Randomly choose $k$ examples as seeds, one for each cluster.
- Form initial clusters based on these seeds.
- Iterate by repeatedly reallocating instances to different clusters to improve the overall clustering.
- Stop when clustering converges or after a fixed number of iterations.
K-means algorithm (MacQueen 1967)

**Input:** D={x₁, x₂, ..., xₙ} and desired number of clusters k

**Output:** c₁, ..., cₖ such that D = c₁ U c₂ ... U cₖ

Let d be the distance function between examples

1. **Initialization** – select k random samples from D as centers {μ₁, ..., μₖ}

2. Do

3. For each example xᵢ do

4. assign xᵢ to cluster cⱼ such that d(μⱼ, xᵢ) is minimized

5. for each cluster cⱼ update its cluster center

6. \[
   μⱼ = \frac{1}{|cⱼ|} \sum_{x \in cⱼ} x
   \]

7. **Until convergence**
K-Means Example (K=2)
K-Means Example (K=2)

Pick seeds
K-Means Example (K=2)

Pick seeds
Reassign clusters
K-Means Example (K=2)

Pick seeds
Reassign clusters
Compute centroids
K-Means Example (K=2)

Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
K-Means Example (K=2)

Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
Compute centroids
K-Means Example (K=2)

Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
Compute centroids
Reassign clusters
K-Means Example (K=2)

Pick seeds
Compute centroids
Reassign clusters
Compute centroids
Reassign clusters
Converged!
Monotonicity of K-means

- **Monotonicity Property:** Each iteration of K-means strictly decreases the MSE until convergence.

- The following lemma is key to the proof:
  - **Lemma:** Given a finite set $C$ of data points, the value of $\mu$ that minimizes the MSE:

$$
J = \sum_{x \in C} ||x - \mu||^2
$$

is:

$$
\mu = \frac{1}{|C|} \sum_{x \in C} x
$$
Proof of monotonicity

• Given a current set of clusters with their means, the MSE is given by:

\[ J_e = \sum_{i=1}^{\kappa} \sum_{x \in c_i} ||x - \mu_i||^2 \]

• Consider the reassignment step:
  – Since each point is only reassigned if it is closer to some other cluster than its current cluster, so we know the reassignment step will only decrease MSE

• Consider the re-center step:
  – From our lemma we know that \( \mu_i' \) minimizes the distortion of \( c_i' \) which implies that the resulting MSE again is decreased.

• Combine the above two, we know Kmeans always decreases the MSE
Kmeans properties

• Kmeans always converges in a finite number of steps
  – Typically converges very fast (in fewer iterations than the number of points)

• Time complexity:
  – Assume computing distance between two instances is $O(d)$ where $d$ is the dimensionality of the vectors.
  – Reassigning clusters: $O(kn)$ distance computations, or $O(knd)$.
  – Computing centroids: Each instance vector gets added once to some centroid: $O(nd)$.
  – Assume these two steps are each done once for $I$ iterations: $O(Iknd)$.
  – Linear in all relevant factors, assuming a fixed number of iterations, more efficient than $O(n^2)$ HAC.
More Comments

• Highly sensitive to the initial seeds

• This is because MSE has many local minimal solutions, i.e., solutions that can not be improved by local reassignments of any particular points
Solutions

• Run multiple trials and choose the one with the best MSE
  – This is typically done in practice

• Heuristics: try to choose initial centers to be far apart
  – Using furthest first traversal
    • Start with a random initial center, set the second center to be furthest from the first center, the third center to be furthest from the first two centers, and so on

• One can also initialize with results of other clustering method, then apply kmeans
Even more comments

• K-Means is exhaustive:
  – Cluster every data point, no notion of outlier

• Outliers may cause problems, why?
  • Outliers will strongly impact the cluster centers
  • Alternative: K-medoids – instead of computing the mean of each cluster, we find the medoid for each cluster, i.e., the data point that is on average closest to other objects in the cluster
  • This will result in a computationally more expensive procedure, but more robust to outliers
Deciding $k$ – a model selection problem

- What if we don’t know how many clusters there are in the data?
- Can we use MSE to decide $k$ by choosing $k$ that gives the smallest MSE?
  - We will always favor larger $k$ values
- Any quick solutions?
  - Find the knee
- We will see some other model selection techniques later