CLUSTERING

Quiz information

- The second midterm quiz is on Thursday (11/21)
- In-class (75 minutes!)
- Allowed one two-sided (8.5x11) cheat sheet
- Solutions for optional problems to HW5 posted today

Quiz information

- Covered
  - Markov models, filtering, smoothing
  - Supervised learning, decision trees
  - Perceptrons, neural networks
  - Support vector machines, naïve Bayes
  - Ensembles
  - Clustering (today's lecture only)
- Not Covered
  - Prediction, Most likely explanation, Viterbi Algorithm
  - Particle filtering
  - Pruning decision trees
  - Won't ask you to derive Delta algorithm, Backprop., SVMs
  - Expectation Maximization
  - No calculator needed

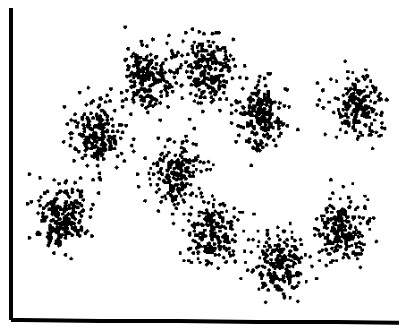
Today

- Reading
  - Introduction to Information Retrieval (IR) Ch. 16, 17
- Goals
  - Flat clustering algorithms
    - K-means
  - Hierarchical clustering algorithms
    - Agglomerative clustering
    - Divisive clustering (it's divisive!)
  - Evaluating clusters

Unsupervised Learning

- Learning without labels
  - Often comes down to clustering
  - Can be used as a surrogate for supervised learning

$$D = \begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \\ \\ \mathbf{x}_N \end{array} \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1M} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2M} \\ & & & & \\ & & & & \\ x_{N1} & x_{N2} & x_{N3} & \cdots & x_{NM} \end{bmatrix}$$



Clustering

- Grouping data into (hopefully useful) sets

- Why do clustering?
  - Labeling is costly
  - Data pre-processing
    - Text Classification (e.g., search engines, Google Sets)
  - Hypothesis Generation/Data Understanding
    - Clusters might suggest natural groups.
  - Visualization

Terminology

- An m-clustering of D is a partition of D into sets (clusters) C_1, C_2,…,C_m such that
  - The clusters are non-empty
  - The union of the clusters is D
  - The intersection of the clusters is empty
- The centroid of a cluster is the mean of all the elements in the cluster

How many possible clusterings?

- Stirling number of the second kind
  - n – size of dataset
  - m – number of clusters

$$S(n,m) = \frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-i}\binom{m}{i}i^n$$

$$S(15,3) = 2{,}375{,}101$$
$$S(20,4) = 45{,}232{,}115{,}901$$
$$S(100,5) \approx 10^{68}$$

How many possible clusterings?

- We can't try all possible clusterings.
- Clustering algorithms look at a small fraction of all partitions of the data.
- The exact partitions tried depend on the kind of clustering used.

$$S(n,m) = \frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-i}\binom{m}{i}i^{n}$$

Types of clustering algorithms

- Flat versus Hierarchical
  - Flat algorithms return an unstructured set of clusters
  - Hierarchical algorithms return a hierarchy of clusters
- Sequential (online) versus Batch
  - Sequential algorithms are typically fast
- Hard versus soft
  - Hard algorithms make a hard assignment of elements to clusters
  - Soft algorithms compute a distribution over clusters for each element

K-means Clustering

- The most well-known (widely-used) clustering algorithm
- Minimizes the sum of the squared distances of each vector from its centroid:

$$||\vec{x} - \mu_i||_2 = \left(\sum_{m=1}^{M} (x_m - \mu_{im})^2 \right)^{1/2}$$

Euclidean distance

$$\text{RSS} = \sum_{k=1}^{K} \sum_{\vec{x} \in C_k} ||\vec{x} - \mu_k||_2^2$$

Residual sum of squares

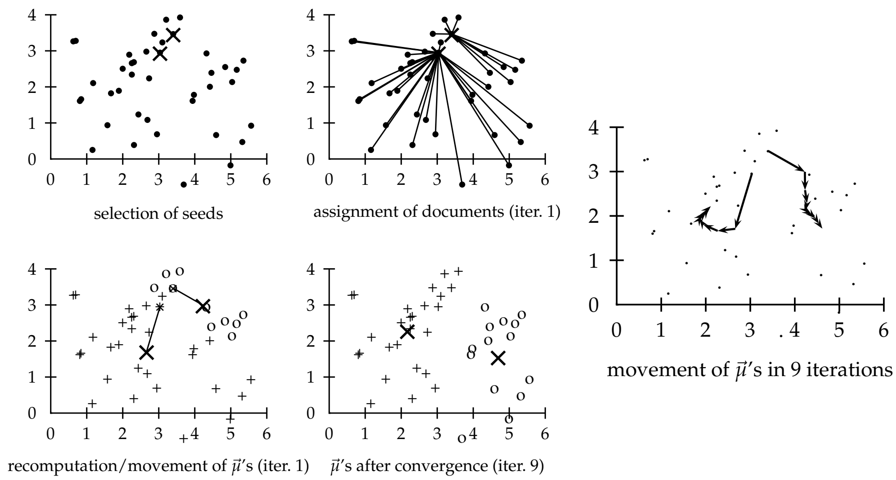
- User must set K
- Assumes instances x represented as normalized vectors in a real-valued space

K-means Clustering

$\text{K-MEANS}(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow \text{SELECTRANDOMSEEDS}(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
for $k \leftarrow 1$ **to** K
do $\vec{\mu}_k \leftarrow \vec{s}_k$
while stopping criterion has not been met
do for $k \leftarrow 1$ **to** K
    do $\omega_k \leftarrow \{\}$
    for $n \leftarrow 1$ **to** N
    do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
        $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
    for $k \leftarrow 1$ **to** K
    do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

K-means Clustering



K-means Clustering

- Starting seeds (centroids)
  - Randomly select K initial data points to be the centroids
  - Run multiple K-means each with different random seeds
  - Use results from a different clustering algorithm (run on random subset of data)
- Stopping criteria
  - See IR for a number of possible stopping criteria. For example:
  - Fixed number of iterations
  - Residual sum of squares (RSS) falls below threshold
- Choosing K
  - Rules-of-thumb or experience
  - Try multiple values for K and plot RSS. Look for the elbow.
  - Add regularization term

Flat versus Hierarchical

- K-means
  - Returns unstructured set of clusters
  - Requires user to determine K
  - Non-deterministic
  - Linear run time O(IKNM)
- Hierarchical (e.g. Agglomerative clustering)
  - Returns a hierarchy of clusters
  - No need to (initially) determine K
  - Deterministic
  - Quadratic run time

Hierarchical Clustering

- Cluster data points based on distance metric or similarity measure
- A norm is a function that assigns a positive real-valued number (representing length or size) with every vector in a vector space
- The p-norm (L_p) of a vector y in R^M is given by

$$||\vec{y}||_p = \left(\sum_{m=1}^{M} |y_m|^p \right)^{1/p}$$

$$\vec{y} = \vec{x}_i - \vec{x}_j$$

- A metric is a function that assigns a positive real-valued number (representing distance) to every pair of vectors in some vector space
- Minkowski distance is given by

$$d_p(\vec{x}_i, \vec{x}_j) = \left(\sum_{m=1}^{M} |x_{im} - x_{jm}|^p \right)^{1/p}$$

Hierarchical Clustering

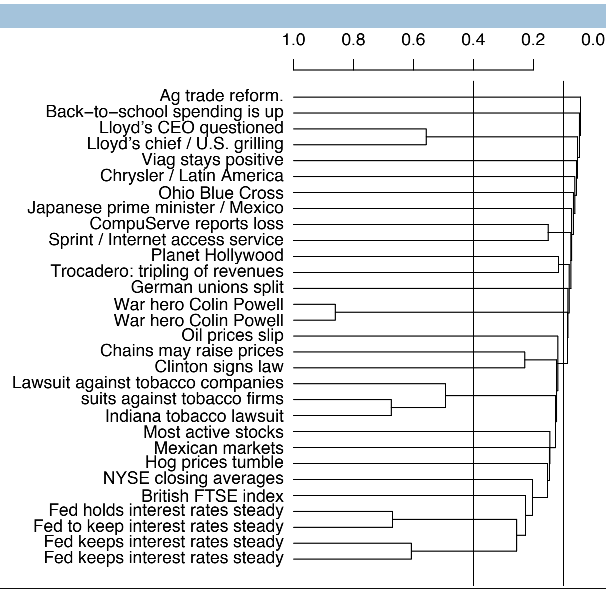
- Minkowski distance is given by $d_p(\vec{x}_i, \vec{x}_j) = \left(\sum_{m=1}^{M} |x_{im} - x_{jm}|^p \right)^{1/p}$
- For p = 1, Manhattan distance $d_1(\vec{x}_i, \vec{x}_j) = \sum_{m=1}^{M} |x_{im} - x_{jm}|$
- For p = 2, Euclidean distance $d_2(\vec{x}_i, \vec{x}_j) = \left(\sum_{m=1}^{M} |x_{im} - x_{jm}|^2 \right)^{1/2}$
- Cosine similarity also common measure (Note inverse of distance)

$$\cos(\vec{x}_i, \vec{x}_j) = \frac{\vec{x}_i^{\mathsf{T}} \vec{x}_j}{||\vec{x}_i||_2 \, ||\vec{x}_j||_2} = \frac{\sum_{m=1}^{M} x_{im} \cdot x_{jm}}{||\vec{x}_i||_2 \, ||\vec{x}_j||_2}$$

Hierarchical Clustering

- Agglomerative clustering
  - Start with *N* clusters each with one data point
  - Merge similar clusters to form larger clusters until there is only a single cluster left
- Divisive Clustering
  - Start with a single cluster containing all data points
  - Divide large clusters into smaller clusters until each cluster contains a single data point

Agglomerative Clustering



Agglomerative Clustering

SimpleHAC($d_1, \ldots, d_N$)
for $n \leftarrow 1$ **to** N
do for $i \leftarrow 1$ **to** N
  do $C[n][i] \leftarrow \text{Sim}(d_n, d_i)$
  $I[n] \leftarrow 1$ *(keeps track of active clusters)*
$A \leftarrow []$ *(assembles clustering as a sequence of merges)*
for $k \leftarrow 1$ **to** $N-1$
do $\langle i, m \rangle \leftarrow \arg\max_{\{\langle i,m \rangle : i \neq m \wedge I[i]=1 \wedge I[m]=1\}} C[i][m]$
  A.Append($\langle i, m \rangle$) *(store merge)*
  for $j \leftarrow 1$ **to** N
  do $C[i][j] \leftarrow \text{Sim}(i, m, j)$
    $C[j][i] \leftarrow \text{Sim}(i, m, j)$
  $I[m] \leftarrow 0$ *(deactivate cluster)*
return A

Compute similarity matrix

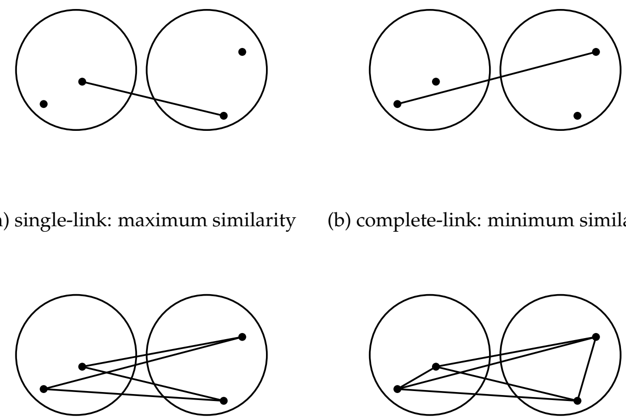
Find two clusters to merge

The i^{th} row and column of C are now the distances for the new cluster

Similarity between cluster c_j and cluster $c_i \cup c_m$

► **Figure 17.2** A simple, but inefficient HAC algorithm.

Agglomerative Clustering



(a) single-link: maximum similarity (b) complete-link: minimum similarity

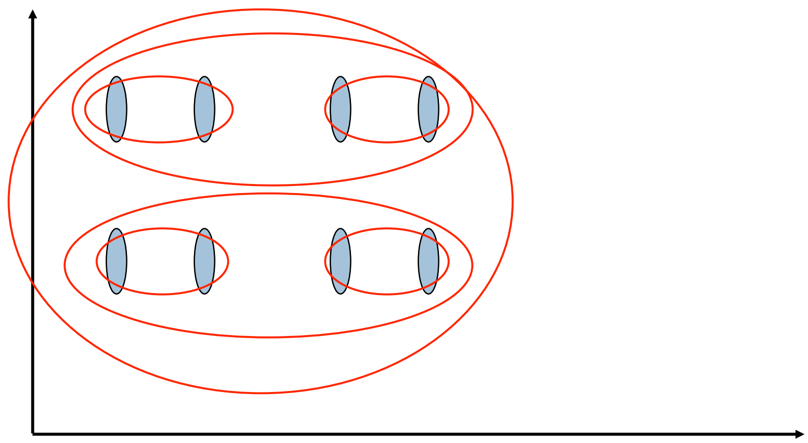
(c) centroid: average inter-similarity (d) group-average: average of all similarities

Cluster similarity: Single-link

- Single link
  - Similarity of c_j and c_i U c_m is the similarity of their *most similar* members
  - Can result in unwanted "long" clusters due to chaining

$$sim((c_i \cup c_m), c_j) = \max(sim(c_i, c_j), sim(c_m, c_j))$$

Cluster similarity: Single-link

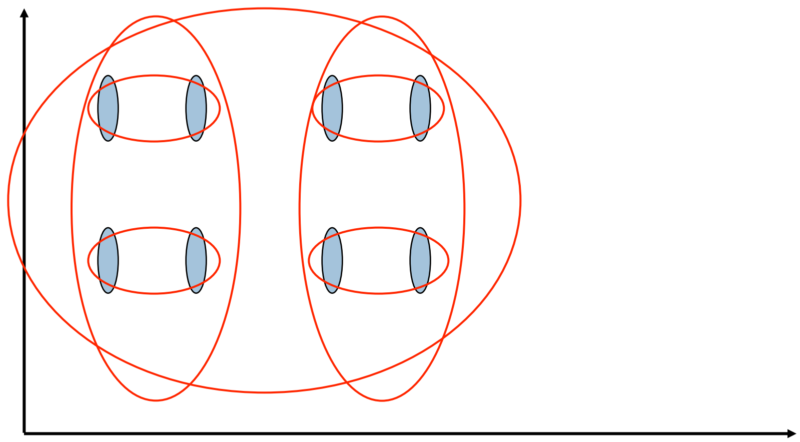


Cluster similarity: Complete-link

- Complete link
  - Similarity of c_j and $c_i \cup c_m$ is the similarity of their least similar members
  - Makes “tighter” spherical clusters that are typically preferable.
  - Sensitive to outliers

$$sim((c_i \cup c_m), c_j) = \min(sim(c_i, c_j), sim(c_m, c_j))$$

Cluster similarity: Complete-link



Cluster similarity: Group-average

- Group-average (average-link)
  - Uses all vectors in clusters c_j and c_i U c_m to compute similarity
  - Average similarity between all pairs of vectors from c_j and c_i U c_m (including pairs from same cluster)
  - Efficient computing of the group-average can be done if using cosine similarity

$$\mathrm{sim}(c_i, c_j) = \frac{1}{\Big(|c_i| + |c_j|\Big)\Big(|c_i| + |c_j| - 1\Big)} \sum_{\substack{\vec{x}_n, \vec{x}_m \in c_i \cup c_j \\ \vec{x}_n \neq \vec{x}_m}} d(\vec{x}_n, \vec{x}_m)$$

Total number of pairs

All pairs of distinct vectors from c_i U c_j

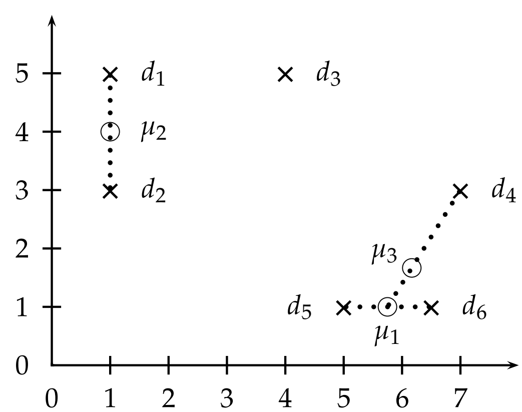
Cluster similarity: Centroid

- Centroid clustering
  - Similarity of cluster c_j and cluster c_i U c_m is the similarity of their centroids

$$
\begin{aligned}
\text{SIM-CENT}(\omega_i, \omega_j) &= \vec{\mu}(\omega_i) \cdot \vec{\mu}(\omega_j) \\
&= \left(\frac{1}{N_i} \sum_{d_m \in \omega_i} \vec{d}_m\right) \cdot \left(\frac{1}{N_j} \sum_{d_n \in \omega_j} \vec{d}_n\right) \\
&= \frac{1}{N_i N_j} \sum_{d_m \in \omega_i} \sum_{d_n \in \omega_j} \vec{d}_m \cdot \vec{d}_n
\end{aligned}
$$

  - Equivalent to the average similarity of all pairs of documents from different clusters
  - Similarity between clusters can increase as we merge clusters (known as inversions)
    - Horizontal merge lines can be lower than the previous merge line

Cluster similarity: Centroid



Computational Complexity

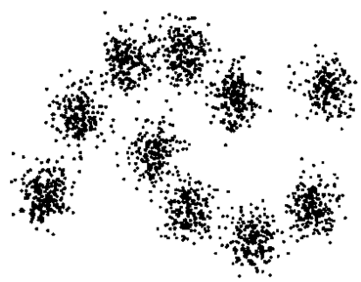
- In the first iteration, all HAC methods need to compute similarity of all pairs of N initial instances, which is $O(N^2)$.
- In each of the subsequent N-2 merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall $O(N^2)$ performance, computing similarity to each other cluster must be done in constant time.
  - Often $O(N^3)$ if done in a naïve way
  - or $O(N^2 \log N)$ if done in a more clever way

Divisive Clustering

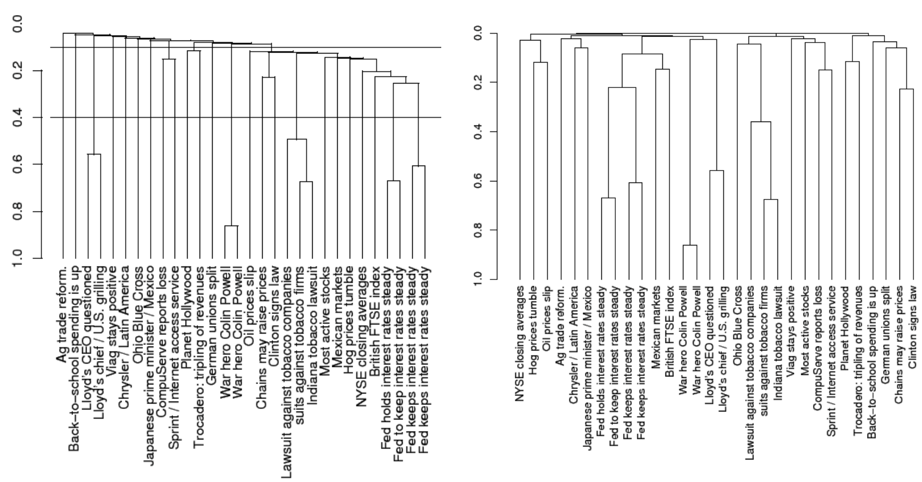
- Top-down clustering
- Divisive clustering algorithm uses a flat clustering algorithm as a subroutine
  - Start with all data points in one cluster
  - Split using a flat clustering algorithm
  - Apply recursively until each data point is in its own cluster
- Can be more efficient than agglomerative
- Benefits from complete information about the entire data set

Which clustering is correct?

- Different techniques cluster the same data set differently.
- Who is right? Is there a "right" clustering?



Which clustering is correct?

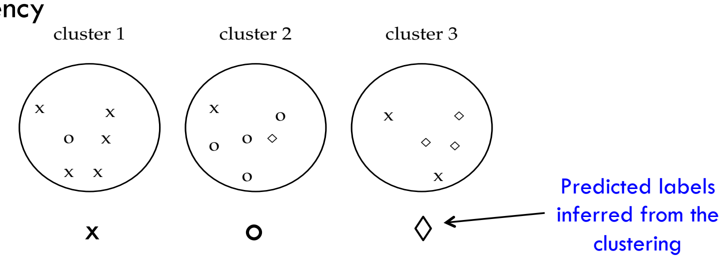


Which clustering is correct?

- Internal criteria
  - A good clustering has high intra-cluster similarity and low inter-cluster similarity
- External criteria
  - Use an external task (e.g. search, document classification) to validate the clustering
  - Requires labeled data

External Criteria

- Purity
  - Set aside labels from labeled data
  - Cluster data
  - Predicted label for each cluster is label with highest frequency



  - Compute accuracy: $\frac{5+4+3}{17} = 0.71$

External Criteria

- Normalized Mutual Information
  - Mutual Information is an information theoretic quantity similar to entropy and information gain

$$\mathrm{I}(X,Y) = \sum_y \sum_x p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = \mathrm{H}(X) - \mathrm{H}(X|Y)$$

  - How much information does the clustering contain about the class labels?

External Criteria

- Normalized Mutual Information
  - Define random variables for the clustering and for the class label:

Prob. data point is in cluster k and has label j

Prob. data point has label j

$$\begin{aligned} I(\Omega;\mathbb{C}) &= \sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} \\ &= \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \frac{N|\omega_k \cap c_j|}{|\omega_k||c_j|} \end{aligned}$$

R.V. for the cluster

Prob. data point is in cluster k

R.V. for the class label

External Criteria

- Normalized Mutual Information
  - Given by the equation:

$$\text{NMI}(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2}$$

  - Why are we normalizing by the entropy?

Rand Index

- Two data points should be in the same cluster if and only if they have the same label
- Define contingency table:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- Once we have a contingency table, we can compute the Rand Index which is just the accuracy

$$\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$$

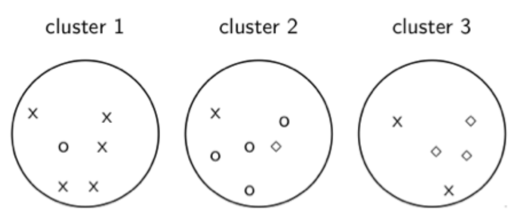
38

Rand Index Example

- There are $\binom{17}{2} = 136$ pairs of data points

	same cluster	diff. cluster
same class	20	24
diff class	20	72

RI = (20+72)/136 = 0.68



39

F-measure

- Given the contingency table, we can compute the precision, recall, and F-measure

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

- The parameter β controls the weighting between precision and recall

Clustering Evaluation

	purity	NMI	RI	F_5
lower bound	0.0	0.0	0.0	0.0
maximum	1.0	1.0	1.0	1.0
value for example	0.71	0.36	0.68	0.46

All four measures range from 0 (really bad clustering) to 1 (perfect clustering).