

- 1) data preparation and attribute selection, 2) similarity measure selection, 3) algorithm and parameter selection, 4) cluster analysis, 5) validation.

Feature Selection

Selected features are stored in a vector for use with a (dis)similarity measure.

Brand	Type	Usable Min (in)	Usable Max (in)	Min Width	Weight (oz)	Color	Price	Own
BD	Camalot 0.1	0.36	0.46	1.25	2.2	red	59 N	59 N
BD	Camalot 0.2	0.43	0.55	1.3	2.4	yellow	59 N	59 N
CCH	Allen 2	1.3	1.7	1.33	4.4	purple	49 Y	49 Y
Metolius	TCU 1	0.52	0.65	1.22	2.3	blue	49 Y	49 Y
Metolius	TCU 2	0.63	0.78	1.21	2.5	yellow	49 N	49 N

Feature vectors can hold continuous or binary values:
 $x = (\text{Min, Max, Width, Weight, Price})$
 $b = (\text{PrimaryColor, Own, Below50, Above50})$

Normalization

Large scale features may be scaled to reduce biases.

Dimensionality Reduction

High dimensional data can be reduced using PCA, MDS, Fastmap or other algorithms that project data into lower dimensional spaces.

(dis)Similarity*

Distance Metrics

Minkowski metrics correspond to the common sense notions of distance on a grid. These metrics work well with compact or isolated clusters but can give excessive weight to "large-scaled" features. Scaling or normalization prior to clustering can help alleviate this.

Mahalanobis distance removes the effect of linear correlation between features by including the covariance matrix Σ in the distance calculation. x and y must be from the same distribution. If $\Sigma = \text{Identity}$, then this is simply the Euclidean distance. If Σ is diagonal, then this is referred to as the *normalized Euclidean distance*.

The cosine distance computes the cosine of the angle between two feature vectors and is used frequently in text mining where vectors are very large but sparse.

Manhattan, $d=4$

Euclidean, $d=3.16$

Infinity, $d=3$

Cosine, $d=.93$

$$D_{\text{Manhattan}} = \sum_{i=1}^n |x_i - y_i| \quad (1\text{-norm})$$

$$D_{\text{Euclidean}} = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{\frac{1}{2}} \quad (2\text{-norm})$$

$$D_{\text{Minkowski}_p} = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (p\text{-norm})$$

$$D_{\text{Infinity}} = \lim_{p \rightarrow \infty} \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (\infty\text{-norm})$$

$$D_{\text{Mahalanobis}}(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}$$

$$D_{\text{Cosine}}(A, B) = \frac{A \cdot B}{|A||B|}$$

Correlation Coefficients

The Pearson correlation coefficient compares profiles that share common sampling points, such as gene expression patterns or time series. The Jackknife coefficient takes the best of multiple correlations between two variables, excluding one feature at a time to remove the effect of abnormal spikes in single samples.

The Spearman rank-order coefficient ranks each feature in the vectors and uses the ranks r_x instead of the values for comparison.

Profiles compared with Pearson (top) and Jackknife with the 6th feature removed (bottom). The matrices show the profile correlations.

a	b	c
1.0	0.3	0.7
b	1.0	0.5
c	0.5	1.0

$$\text{Pearson}_{x,y} = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sigma_x \sigma_y} \quad (\text{aka } \rho_{x,y})$$

$$\text{Jackknife}_{x,y} = \min(\rho_{xy}^1, \rho_{xy}^2, \dots, \rho_{xy}^n)$$

$$\text{Spearman}_{x,y} = 1 - \frac{6 \sum_{i=1}^n (r_{xi} - r_{yi})^2}{|r_x||r_y|^2 - 1}$$

Binary Distances

The Rand and Jaccard coefficients measure the similarity between binary vectors. The Rand coefficient considers the absence of truth important whereas the Jaccard ignores it. For binary vectors x and y , a is the number of features that are true in x and not y , b the number in y but not x , c the number of features that are true in both, and d the number that are false in both.

$$D_{\text{Rand}} = \frac{c + d}{a + b + c + d}$$

$$D_{\text{Jaccard}} = \frac{c}{a + b + c}$$

$x: 101011101$
 $y: 110011010$
 cbadccaba

$D_{\text{Rand}} = .444, D_{\text{Jaccard}} = .375$

Information Theoretic Measures

Mutual information gives the amount of information, in 'bits', shared between two random variables, X and Y . If X and Y are independent, $\text{MutInf}(X, Y) = 0$

$$\text{MutInf}(X, Y) = \sum_{x,y} p(x,y) \cdot \log_2 \frac{p(x,y)}{f(x)g(y)}$$

*similarities are larger if objects are closer, dissimilarities are smaller (e.g. distance)

Hierarchical Clustering

Hierarchical clustering algorithms iteratively build clusters by joining (agglomerative) or dividing (divisive) the clusters from the previous iteration. The resulting tree has nodes created at each cutoff point that can be used to generate different clusterings (figure (d)).

Example: Hierarchical Agglomerative Clustering

- 1) Assign each object its own cluster
- 2) Select a distance cutoff d (a-d, red lines)
- 3) For each pair of clusters (x,y) where $\text{dist}(x,y) < d$, merge (x,y) into a new cluster
- 4) Repeat 2-3 until all objects are in one cluster

The three main linkage algorithms compute inter-cluster distances at each iteration:

Single link: the shortest distance between objects

Complete link: the largest distance between objects

Average link: various average distance algorithms

The common average link methods are UPGMA - Unweighted Pair-Groups Method Average - and UPGMC - Unweighted Pair-Groups Method Centroid (pictured above). Weighted versions also exist. An alternative merging algorithm, *Ward's method*, merges clusters that produce the least variance.

Complexity: $O(n^2)$, nearest-neighbor (NN) chains can improve inter-cluster distance computations

Pros: Best overall clustering technique, flexible cluster structure, linkage algorithms can be tuned to data

Cons: Expensive, not all linkage algorithms detect same types of clusters

Uses: Small to medium size data, single-link can find chained and concentric clusters

Examples: CURE, CHAMELEON, ROCK, PDDP (divisive), COBWEB (conceptual)

Density-based Clustering

Density-based algorithms build clusters based on a rigorous definitions of dense areas and connectivity built around a neighborhood parameter, ϵ , and a minimum density measure, MinPts :

- p is in the ϵ -neighborhood of q if $\text{dist}(p,q) < \epsilon$
- A **core-object** (b , red point) has at least MinPts in its ϵ -neighborhood
- p is **directly density-reachable** from q if q is a core object and p is in q 's ϵ -neighborhood (b , blue points)
- p is **density-reachable** from q if there is a chain of directly density reachable objects between them (c , arrows)
- p is **density-connected** to q if they are both density-reachable from an object o . (d , arrows)

Example: OPTICS

- 1) Choose p at random or from the queue, output p and its core/reachability distances
- 2) Find p 's **core-distance**, the smallest distance $e' \leq \epsilon$ such that p is a core-object
- 3) Compute the reachability-distance (red lines) for each other point in p 's ϵ -neighborhood
- 4) Visit the points in order of their reachability-distance, repeating steps 1,2,3 as necessary

OPTICS generates an ordering of the points that can be easily visualized and used to extract partitions and hierarchical clusters.

Complexity: $O(n \log n)$

Pros: Fast, generates hierarchal clusters and partitions, finds odd shaped and concentric clusters

Cons: New (early 90s), few real-world examples, interpretation not well understood

Uses: All data sizes, spatial data, visual data exploration with reachability plots

Examples: (G)DBSCAN, DBCLASD, DENCLUE, DHC

Partitional Clustering

Partitional, or relocation, clustering algorithms assign each object to a partition. Squared-error algorithms start with a fixed number k of partitions and attempt to minimize an objective function that assigns objects to clusters. Graph-based clustering algorithms build graphs using combinations of the objects, features, or both as the nodes/edges and partition the graph using graph-theoretic algorithms.

Example: K-Means

- 1) Initialize k centroids with random points, sampled points, or a more complex algorithm (e.g., SVD) (a)
- 2) For each object, find the closest centroid and assign it to that cluster (b-c, red lines)
- 3) Move each centroid to the center of the points assigned to it (c-d)
- 4) If the centroids are stable (e.g., didn't move), end, otherwise, repeat steps 2-3

Steps 2 and 3 are an instance of the *Estimation Maximization* (EM) algorithm. Step 2 estimates the cluster, step 3 maximizes it based on the latest estimate.

Standard k-means is a *crisp* clustering algorithm, assigning each point to one cluster. A *soft* variation, *fuzzy k-means* (aka *fuzzy c-means*), assigns every point to every cluster, weighting the membership and relocation effects by a proportional contribution from each point.

Graph algorithms are usually a based on a *minimum spanning tree* of the graph. Edges above a threshold (red edges) are removed, leaving clusters behind.

Complexity: K-Means: $O(kn)$ per iteration; Graph: $O(\text{edges})$ or $O(\text{vertices})$ for planar graphs

Pros: Easily understood, approx. methods limit iterations, k-means is most used method

Cons: K-Means: Fixed k , assumes normal distribution, won't find concentric clusters; Graph: expensive for highly connected graphs

Uses: All data sizes, best with well separated clusters

Examples: K-medoids, K-harmonic means, PAM, CLARA, CLARANS; Graph: HMETIS, BAG

Other Methods

Probabilistic Clustering

Goal: Fit models to data, cluster based on models

Pros: Easy to interpret results, 'on-line' (can be stopped/resumed), complex models possible

Cons: Expensive, can get stuck in local minima, requires careful setup

Uses: All data, large databases

Examples: EM, AUTOCLASS, SNOB, MCLUST

Grid Clustering

Goal: Partition the *attribute space* into segments

Pros: Generally not dependent on data ordering, works with heterogeneous features,

Cons: Space partitions don't allow for irregular clusters

Uses: High-dimensional data, industrial databases

Examples: CLIQUE, MAFIA (dim. reduction); BANG, GRIDCLUST (heirarchical); STING (statistical); WaveClust (wavelet); FC (fractal)

Artificial Intelligence

Goal: Apply AI techniques to clustering

Pros: Adaptive, parallel, closely related to other algorithms (e.g., SOM \approx k-means)

Cons: Local optimas, long running time, results difficult to interpret

Uses: Small to medium sized data

Examples: Self-organizing maps (SOM), Genetic Algorithms, Tabu search

SOMs are useful for visualizing natural clusters

SVD/PCA

Goal: Use the eigenvectors or prinicipal components as cluster representatives

Pros: Standard multivariate analysis techniques

Cons: Expensive, cluster interpretations not always meaningful

Uses: Small, high-dimensionl data, dimensionality reduction, cluster seeding

External Measures

External measures compare the clusters with known class labels. Purity and completeness measure how many items in the cluster are from the same class and how many clusters a class was divided into. These values can be used with the *F-measure* get assess the overall clustering quality.

The Rand and Jaccard coefficients can be used for cluster validation by comparing the cluster and class assignments between all pairs of points. Using the equations from the Similarity panel, c and d count the cases where clusters and class assignments agree and a and b count the cases where they disagree.

Internal Measures

Internal measures assess the clusters against their own structural properties. Compactness, connectedness, and separation use statistical measures such as intra-cluster variance and distance between centroids to evaluate clusters. These can be combined to form composite measures such as the *Silhouette Width* and the *Dunn Index*.

Prediction Strength

Prediction strength methods remove an element and check the robustness of the clusters by 1) re-clustering the data and comparing it against the original clusters, or 2) building a classifier from the remaining items and see if it properly classifies (predicts) the missing item.

Simulation

If enough statistical properties are available to generate synthetic data sets, simulations can provide a systematic way to study the effectiveness of different clustering algorithms prior to clustering the actual data.

Human Experts and Custom Measures

The most common form of validation is to have an expert inspect the clusters and see if they make sense. Another common approach is to 'invent' a measure based on assumptions about the algorithm and data. These are best used only for exploratory analysis.

Validation

Two alternative clusterings of the same data using K-means. How well does each clustering hold up to different validation measures? What does this tell us about the clustering process in general?

If $P(t, C_k)$ is the purity of cluster C_k wrt. class t , $R(t, C_k)$ the completeness, N_i the size of t and N the data size, then the F-measures for a class and clustering are:

$$F(t, C_k) = \frac{2P(t, C_k)R(t, C_k)}{P(t, C_k) + R(t, C_k)}, \quad F(C) = \sum_{t \in T} \frac{N_t}{N} \cdot \max_{C_k \in C} F(t, C_k)$$

If a_i is the average distance between i and the items in the same cluster and b_i is the average distance to the items in the closest cluster, then the Silhouette value for i and the Silhouette width for the clustering are:

$$S_i = \frac{b_i - a_i}{\max(b_i, a_i)}, \quad S_C = \frac{\sum_i S_i}{n}$$

If $\text{diam}(C_m)$ is the max intra-cluster distance in C_m and $\text{dist}(C_k, C_l)$ is the min distance between pairs of items in C_k and C_l , the Dunn Index is:

$$D_C = \min_{C_k \in C} \left(\min_{C_l \in C} \frac{\text{dist}(C_k, C_l)}{\max_{C_m \in C} \text{diam}(C_m)} \right)$$

The Validation section is based on Handl, et. al. 2005

Applications

Paper	Data	Pre-processing	(dis)Similarity	Algorithm	Validation	Results
Eisen 1998	Microarray	Normalize, Filter Noise, Subset	$\frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sigma_x \sigma_y}$	Agglomerative	Human, "Scramble" Strength	Redundant genes cluster, Genes with similar function cluster
Ralf-Herwig 1999	cDNA Fingerprint	Subset (sim only)	$\sum_{x,y} p(x,y) \cdot \log_2 \frac{p(x,y)}{f(x)g(y)}$	Graph-based	Custom, Simulation	Short (< 500 bp) seqs cluster, Parallel software
Gasch 2002	Microarray	Filter by $\sigma(\text{expr})$, PCA (seed)	$\frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sigma_x \sigma_y}$	Fuzzy k-means	Human, Truth	Genes assigned to mult. clusters show subtle coregulation patterns, Found correlations with experimental conditions
Holliday 2004	2D Structure	PCA (dim. red.)	$\left(\sum_{i=1}^n x_i - y_i ^2 \right)^{\frac{1}{2}}$	Graph-based	Predictive Strength, Truth	Grouped structurally similar molecules together, Clusters are hard to interpret
Alter 2000	Microarray image from Eisen 1998, SVD images from Alter 2000, caffeine image from wikipedia.org	SVD of Expression Matrix, Filter eigengenes and eigenarrays, Sort by correlation to principal eigenarrays			Human, Truth	Detected cell cycles from eigengenes, Detected cell states from eigenarrays

The complete list of references is available at: <http://www.osl.iu.edu/~chemuell/new/oral-quals.php>
 Unless otherwise noted, all text and graphics ©2005 Christopher Mueller