Clustering with Constraints: Incorporating Prior Knowledge into Clustering

Adapted from a Tutorial of Sugato Basu and Ian Davidson (SDM 2005)

Clustering

- The given data consists of input vectors without any corresponding target values
- The goal is to discover groups of similar examples within the data
A Motivating Example

- Given a set of instances $S$
- Find the “best” set partition
  $S = \{S_1 \cup S_2 \cup ... S_k\}$
- Multitude of algorithms that define “best” differently
  - K-Means
  - Mixture Models
  - Hierarchical clustering
- Aim is to find the **underlying** structure/patterns/groups in the data.

### Clustering Example
(Number of Clusters=2)

![Clustering Example](image-url)
Horizontal Clusters

Vertical Clusters
K-Means Clustering

• Standard iterative partitional clustering algorithm

• Finds \( k \) representative centroids in the dataset
  – Locally minimizes the sum of distance (e.g., squared Euclidean distance) between the data points and their corresponding cluster centroids

\[
\sum_{s_i \in S} D(s_i, C_{li})
\]

K-Means Algorithm

1. Randomly assign each instance to a cluster
2. Calculate the centroids for each cluster
3. For each instance
   • Calculate the distance to each cluster center
   • Assign the instance to the closest cluster
4. Goto 2 until distortion is small
Clustering with Constraints

K Means Example (k=2)

Initialize Means

Assign Points to Clusters
Clustering with Constraints

K Means Example

Re-estimate Means

Re-assign Points to Clusters
K Means Example

Re-estimate Means

K Means Example

Re-assign Points to Clusters
K Means Example
Re-estimate Means and Converge

Convergence
A Few Issues With K-Means

• Sensitivity to initial centroids
  – The algorithm is typically restarted many times from random starting centroids
  – Intelligently setting initial centroids [Bradley & Fayyad 2000]

• Convergence time of algorithm can be slow
  – Use KD-Trees to accelerate algorithms [Pelleg and Moore 1999]

• Which distance function should I use?
  – L1, L2, Mahalanobis etc.

• Constraints can help address these problems and more …

Automatic Lane Finding from GPS traces

[Wagstaff et al. ’01]

- Lane-level navigation (e.g., advance notification for taking exits)
- Lane-keeping suggestions (e.g., lane departure warning)

• Constraints inferred from trace-contiguity (ML) & max-separation (CL)
Mining GPS Traces (Schroedl et’ al)

- Instances are represented by the x, y location on the road. We also know when a car changes lane, but not what lane to.
- True clusters are very elongated and horizontally aligned with the lane central lines
- Regular k-means performs poorly on this problem instead finding spherical clusters.

Unconstrained K-Means Can Provide Not Useful Clusters

Only significant clusters shown
Semi-supervised Learning

- Unlabeled data may be easily available, while labeled ones may be expensive to obtain because they require human effort.
- Semi-supervised learning is a recent learning paradigm: it exploits unlabeled examples, in addition to labeled ones, to improve the generalization ability of the resulting classifier.

With lots of unlabeled data the decision boundary becomes apparent.
Basic Instance Level Constraints

- Historically, instance level constraints motivated by the availability of labeled data
  - i.e., Much unlabeled data and a little labeled data available generally as constraints, e.g., in web page clustering
- This knowledge can be encapsulated using instance level constraints [Wagstaff et al. ’01]
  - Must-Link Constraints
    - A pair of points \( s_i \) and \( s_j \) \((i \neq j)\) must be assigned to the same cluster.
  - Cannot-Link Constraints
    - A pair of points \( s_i \) and \( s_j \) \((i \neq j)\) can not be assigned to the same cluster.

Properties of Instance Level Constraints

- **Transitivity of Must-link Constraints**
  - \( ML(a,b) \) and \( ML(b,c) \rightarrow ML(a,c) \)
  - Let \( X \) and \( Y \) be sets of ML constraints
  - \( ML(X) \) and \( ML(Y) \), \( a \in X, a \in Y \), \( \rightarrow ML(X \cup Y) \)

- **The Entailment of Cannot link Constraints**
  - \( ML(a,b), ML(c,d) \) and \( CL(a,c) \rightarrow CL(a,d), CL(b,c), CL(b,d) \)
  - Let \( CC_1 \) ... \( CC_p \) be the groups of must-linked instances (i.e., The connected components)
  - \( CL(a \in CC_n, b \in CC_p) \rightarrow CL(x,y), \forall x \in CC_n, \forall y \in CC_p \)
Uses of Constraints: The Big Picture

- Clustering with constraints:
  Partition unlabeled data into groups called clusters
  + use constraints to aid and bias clustering

- Goal:
  Examples in same cluster similar, separate clusters different + constraints are maximally respected

Enforcing Constraints

- Clustering objective modified to enforce constraints
  – Strict enforcement: find “best” feasible clustering respecting all constraints
  – Partial enforcement: find “best” clustering maximally respecting constraints

- Uses standard distance functions for clustering

[Demiriz et al.’99, Wagstaff et al.’01, Segal et al.’03, Davidson et al.’05, Lange et al.’05]
Example: Enforcing Constraints

Clustering respecting all constraints

© Basu and Davidson 2005
Clustering with Constraints
Learning Distance Function

- Constraints used to learn clustering distance function
  - $ML(a, b) \rightarrow a$ and $b$ and surrounding points should be “close”
  - $CL(a, b) \rightarrow a$ and $b$ and surrounding points should be “far apart”

- Standard clustering algorithm applied with learned distance function

[Klein et al.’02, Cohn et al.’03, Xing et al.’03, Bar Hillel et al.’03, Bilenko et al.’03, Kamvar et al.’03, Hertz et al.’04, De Bie et al.’04]

Example: Learning Distance Function
Example: Learning Distance Function
Space Transformed by Learned Function

Example: Learning Distance Function
Clustering with Trained Function
Why Learn Distance functions?

Nearest Neighbor

Image retrieval

Given a query image return the K-nearest neighbors of the image from the database.

Euclidean distance on Color Coherence Vectors returns both images as similar to query image

Enforce Constraints + Learn Distance

• Integrated framework [Basu et al.’04]
  – Respect constraints during cluster assignment
  – Modify distance function during parameter re-estimation

• Advantage of integration
  – Distance function can change the space to decrease constraint violations made by cluster assignment
  – Uses both constraints and unlabeled data for learning distance function
Real-world examples

Gene Clustering Using Micro-array Data

- Gene expressions
- Red => low expression w.r.t baseline
- Green => high expression w.r.t baseline
- Gene clusters
- Constraints from gene interaction information in DIP

Experiments
Content Management: Document Clustering

Documents

Clustered Documents

Directory structure constraints

Automatic Lane Finding from GPS traces

[Wagstaff et al. '01]

Lane-level navigation (e.g., advance notification for taking exits)

Lane-keeping suggestions (e.g., lane departure warning)

• Constraints inferred from trace-contiguity (ML) & max-separation (CL)
Benefits of Constraints

- Find clusters where standard distance functions could not
- Find solutions with given properties
- Improve convergence time of algorithms

Learning Distance Functions
The Effects of Constraints on Clustering Solutions

- Constraints divide the set of all plausible solutions into two sets: feasible and infeasible: \( S = S_F \cup S_I \)
- Constraints effectively reduce the search space to \( S_F \)
- \( S_F \) all have a common property
- So it’s not unexpected that we find solutions with a desired property and find them quickly.

Effects of Constraints on Convergence Time

![Graphs showing the effects of constraints on convergence time](image-url)
• Algorithms for constrained clustering
  • Enforcing constraints
  • Hierarchical
  • Learning distances
  • Initializing and pre-processing
  • Graph-based

Enforcing Constraints

• Constraints are strong background information that should be satisfied.
• Two options
  – Satisfy all constraints if possible
  – Satisfy as many constraints as possible
COP-k-Means – Nearest-”Feasible”-Centroid Idea

Input: $S_U$: unlabeled data, $S_L$: labeled data, $c$: the number of clusters to find, $g$: number of constraints to generate
Output: A set partition of $S = S_U \cup S_L$ into $k$ clusters so that all the constraints in $C = ML \cup CL$ are satisfied.

1. $ML = \emptyset$, $CL = \emptyset$
2. loop $g$ times do
   (a) Randomly choose two distinct points $x$ and $y$ from $S_U$
   (b) if(\text{Label}(x) = \text{Label}(y)) $ML = ML \cup \{x, y\}$ else $CL = CL \cup \{x, y\}$
3. Compute the transitive closure from ML to obtain the connected components $CC_1, \ldots, CC_r$
4. For each $i$, $1 \leq i \leq r$, replace data points in $CC_i$ with the average of the points in $CC_i$
5. Randomly generate cluster centers $C_1, \ldots, C_k$
6. loop until convergence do
   (a) for $i = 1$ to $|S|$ do
      (a.1) Assign $x_i$ to closest feasible cluster
   (b) Recalculate $C_1, \ldots, C_k$

Example: COP-K-Means - 1

© Basu and Davidson 2005
Clustering with Constraints
Example: COP-K-Means – 2
ML points Averaged

Example: COP-K-Means – 3
Nearest-Feasible-Assignment
Trying To Minimize VQE and Satisfy As Many Constraints As Possible

• Can’t rely on expecting that I can satisfy all constraints at each iteration.
• Change aim of K-Means from:
  – Find a solution satisfying all the constraints and minimizing VQE
  TO
  – Find a solution satisfying most of the constraints (penalized if a constraint is violated) and minimizing VQE
• Two tricks
  – Need to express penalty term in same units as VQE/distortion
  – Need to re-derive K-Means (as a gradient descent algorithm).

• Algorithms for constrained clustering
  • Enforcing constraints
  • Hierarchical
    • Learning distances
  • Initializing and pre-processing
  • Graph-based
Distance Learning as Convex Optimization [Xing et al. '02]

- Learns a parameterized Mahalanobis distance

\[
\min_A \sum_{(s_i, s_j) \in ML} \|s_i - s_j\|^2_A = \min_A \sum_{(s_i, s_j) \in ML} (s_i - s_j)^T A (s_i - s_j)
\]

s.t. \( \sum_{(s_i, s_j) \in CL} \|s_i - s_j\|^2_A \geq 1 \), \( A \) is positive-definite

Learning Mahalanobis distance

- Mahalanobis distance = Euclidean distance parameterized by matrix \( A \):

\[
\| x - y \|^2_A = (x - y)^T A (x - y)
\]

e.g. Let 2 points be \( x^T = (2,3) \), \( y^T = (4,5) \)

\[
D(x,y) \propto (2-4, 3-5)(2-4, 3-5)^T
\]

\[
= (2-4, 3-5)(l_{1,1}(2-4), l_{2,2}(3-5))^T
\]

\[
\propto 1 \cdot (2-4)^2 + 1 \cdot (3-5)^2
\]

\[
D_A(x,y) \propto (2-4, 3-5)A(2-4, 3-5)^T
\]

\[
\propto A_{1,1}(2-4)^2 + A_{2,2}(3-5)^2
\]

Typically, \( A \) is the covariance matrix, but we can also learn it given constraints
Example: Learning Distance Function

Space Transformed by Learned Function
Example: Learning Distance Function

- Must-link: \( ML(a, b), a = (1,1), b = (1,2) \)
- Cannot-link: \( CL(e, f), e = (1,2), f = (2,1) \)

\[
A = \begin{bmatrix}
1 & 0 \\
0 & \varepsilon
\end{bmatrix}
\]

- \( D(a, b) = \varepsilon \)
- \( D(e, f) = 1 + \varepsilon \)

The Diagonal \( A \) Case

\[
g(A) = g(A_{11}, \ldots, A_{nn}) = \sum_{(x_i, x_j) \in S} ||x_i - x_j||^2_A - \log \left( \sum_{(x_i, x_j) \in D} ||x_i - x_j||^2_A \right)
\]

Use Newton Raphson Technique
• Algorithms for constrained clustering
  • Enforcing constraints
  • Hierarchical
  • Learning distances
    • Initializing and pre-processing
  • Graph-based

Finding Informative Constraints
given a quota of Queries

• Active learning for constraint acquisition [Basu et al.’04]:
  – In interactive setting, constraints obtained by queries to a user
  – Need to get informative constraints to get better clustering

• Two-phase active learning algorithm:
  – Explore: Use farthest-first traversal [Hochbaum et al.’85] to explore
    the data and find $K$ pairwise-disjoint neighborhoods (cluster skeleton)
    rapidly
  – Consolidate: Consolidate basic cluster skeleton by getting more points
    from each cluster, within max $(K-1)$ queries for any point
Algorithm: Explore

- Pick a point $s$ at random, add it to neighborhood $N_j$, $\lambda = 1$
- While queries are allowed and ($\lambda < k$)
  - Pick point $s$ farthest from existing $\lambda$ neighborhoods
  - If by querying $s$ is cannot-linked to all existing neighborhoods, then set $\lambda = \lambda + 1$, start new neighborhood $N_j$ with $s$
  - Else, add $s$ to neighborhood with which it is must-linked

Active Constraint Acquisition for Clustering
Explore Phase
Active Constraint Acquisition for Clustering
Explore Phase

Active Constraint Acquisition for Clustering
Explore Phase
Active Constraint Acquisition for Clustering
Explore Phase

Active Constraint Acquisition for Clustering
Explore Phase
Algorithm: Consolidate

- Estimate centroids of each of the $\lambda$ neighborhoods
- While queries are allowed
  - Randomly pick a point $s$ not in the existing neighborhoods
  - Query $s$ with each neighborhood (in sorted order of decreasing distance from $s$ to centroids) until must-link is found
  - Add $s$ to that neighborhood to which it is must-linked

Active Constraint Acquisition for Clustering
Consolidate Phase

![Active Constraint Acquisition Diagram](image-url)
Active Constraint Acquisition for Clustering
Consolidate Phase

Height

Weight
Active Constraint Acquisition for Clustering
Consolidate Phase

© Basu and Davidson 2005
Clustering with Constraints
- Algorithms for constrained clustering
  - Enforcing constraints
  - Hierarchical
  - Learning distances
  - Initializing and pre-processing
    - Graph-based

Graph-based Clustering

- Data input as graph:
  - real valued edges between pairs of points denotes similarity
Constrained Graph-based Clustering

- Clustering criterion:
  minimize normalized cut

- Possible solution:
  Spectral Clustering
  [Kamvar et al. '03]

- Constrained graph clustering:
  minimize cut in input graph while maximally respecting constraints in auxiliary constraint graph

Kernel-based Clustering

- 2-circles data not linearly separable
- transform to high-D using kernel
  \[ \langle s_1, s_2 \rangle = e^{-|s_1 - s_2|^2} \]

- Cluster data using kernel K-Means
Constrained Kernel-based Clustering

- Use the data and the specified constraints to create appropriate kernel

Today we talked about …

- Introduction
- Uses of constraints
- Real-world examples
- Benefits of constraints
- Algorithms for constrained clustering
  - Enforcing constraints
  - Hierarchical
  - Learning distances
  - Initializing and pre-processing
  - Graph-based
References - 1


References - 2

[16] David Gondet and Thomas Hofmann Non-Redundant Data Clustering, 4th IEEE International Conference on Data Mining (ICDM), 2004: Best Paper Award.
References - 3


References - 4