Constrained K-means Clustering with Background Knowledge

paper by
Kiri Wagstaff, Claire Cardie, Seth Rogers and Stefan Schroedl

presented by
Siddharth Patwardhan

An Overview of the Talk
• Introduction to clustering and the common trends.
• The basic k-means algorithm.
• Applying constraints based on background knowledge.
• How do we evaluate?
• Experiment using artificial constraints.
• A GPS lane finding experiment.
• Related work.
• Conclusions.

Clustering!
• An unsupervised method for data analysis.
• Grouping of data with some notion of “similarity”.
• Uses just the data to determine which of the data points look alike.
• New instances of data are assigned to the closest cluster.

Background Knowledge
• Traditional clustering algorithms don’t use any background knowledge about the data in the clusters.
• If domain knowledge told us…
  – Two data points are part of the same class.
  – Or two data points are in different classes.
• … could we improve the clusters formed?

The K-means Algorithm
• Automatically partitions data points into \( k \) groups.
• Starts with \( k \) initial “cluster centers”, iteratively assigns points to clusters and updates the “cluster centers”.
• Converges when there is no further change in assignment of points to clusters.

Using Background Knowledge
• Two types of constraints based on the domain knowledge.
• Must-link constraints specify that two instances must be in the same cluster.
• Cannot-link constraints specify that two instances cannot be in the same cluster.
The Constrained K-means Algorithm
(1) Initialize the $k$ cluster centers.
(2) For each data point $d_i$, assign $d_i$ to its closest cluster such that none of the constraints are violated. If no such cluster exists, clustering fails.
(3) Update the cluster centers.
(4) Iterate (2) and (3) till convergence.

Testing Constraint Violation
• The distance of a data point $d$ to each of the cluster-centers is computed.
• Constraint violation for $d$ is tested for each cluster in ascending order of distance of $d$ from the cluster-center.

Testing Constraint Violation
• For a cluster $C$, from the data-points that have been assigned to clusters, if the data points that “must-link” to $d$ are not in $C$, then the must-link constraint for $d$ is violated.
• For a cluster $C$, if any of the data points that “cannot-link” to $d$ are in $C$, then the cannot-link constraint for $d$ is violated.

Evaluating Clusters
• $\text{Rand Index}$ used to measure the agreement between partitions.
• In this case the partitions are
  – that formed by the clustering.
  – that specified by the data point labels.
• Accuracy measured for the entire data set and for a “held-out” test set (subset of the non-constrained data point) using a 10-fold cross validation.

More Evaluation
• The constraints can be viewed as a partition of the data points and thus can be evaluated using the $\text{Rand Index}$.
• The accuracy of the partition of just the constraints determine how good the constraints by themselves are at forming the clusters.
• This analysis consequently determines how well the domain knowledge by itself clusters the data.

You Are Here!
• Introduction to clustering and the common trends.
• The basic k-means algorithm.
• Applying constraints based on background knowledge.
• How do we evaluate?
  • Experiment using artificial constraints.
• A GPS lane finding experiment.
• Related work.
• Conclusions.
An Experiment using Artificial Constraints

• Used 6 well-known test sets from the UCI repository to test the performance of the constrained k-means algorithm.
• Tested basic k-means on each data set to provide a baseline for comparison.
• Tested the constrained k-means on each data set, varying the number of constraints.

Generating the Constraints

• Randomly select a pair of data points.
• If the two points have the same label, create a must-link constraint between them.
• If the two points have a different label, create a cannot-link constraint between them.
• Repeat the above process $n$ times to generate $n$ constraints.

The soybean Data Set

• 47 instances, 35 attributes, 4 classes.
• Unconstrained k-means achieves an accuracy of 87 %.
• Accuracy of the constraints alone: 48 %.
• Accuracy of constrained k-means on entire data set increases with the number of constraints up to 99 %, with 100 random constraints.
• With the held-out data set it increases at almost the same rate to 98 %.

The mushroom Data Set

• 50 instances, 21 attributes, 2 classes.
• Unconstrained k-means achieves an accuracy of 69 %.
• Accuracy of the constraints alone: 73 %.
• Accuracy of constrained k-means on entire data set increases with the number of constraints up to 96 %, with 100 random constraints.
• With the held-out data set it increases at almost the same rate to 83 %.
Making a Point

• On the other data sets too (part-of-speech, tic-tac-toe, iris, wine) the overall accuracy rose sharply into the 90s.
• Held-out accuracy increased only marginally.
• Improvement in clustering accuracy depends on the data set in question.
• Improvements can be observed on unconstrained instances, if constraints are generalizable to the full data set.

You Are Here!

• Introduction to clustering and the common trends.
• The basic k-means algorithm.
• Applying constraints based on background knowledge.
• How do we evaluate?
• Experiment using artificial constraints.
  • A GPS lane finding experiment.
• Related work.
• Conclusions.

GPS Lane Finding Experiment

• Representing the problem as a clustering problem.
• Extracting constraints from background knowledge.
• Applying k-means with and without using constraints.
• Comparison of the results.

Lane Finding

• Clustering data points gathered from GPS systems.
• Clusters indicate lanes – densely traveled spaces.
• Can be used to alert drivers drifting from their lane.

Data Description

• Each data point represented by two features:
  – Distance along the road segment.
  – Perpendicular offset from the road centerline.

Road Segment

Data Description

• For evaluation purposes, each data point is also classified by the lane in which it lies.
• Data collected once per second from several drivers with GPS receivers.
• Drivers specified which lane they were in to help label each data point.
**The Background Knowledge**

- Domain-specific heuristics – *trace contiguity* and *maximum separation*.
- Trace contiguity: In the absence of lane changes, all points from a vehicle should be in the same cluster.
- Maximum separation: If two points are at least 4m apart vertically, they cannot be in the same lane.

**Applying the Knowledge**

- Using trace contiguity – Data points generated from the same vehicle that didn’t change lanes in a particular segment constrained to the same cluster.
- Using maximum separation – Data points separated by a distance greater than 4m vertically are constrained to be in different clusters.

**Cluster-Center Representation**

Rather than a point:

![Cluster-Center Representation](image)

Represented as a line:

![Cluster-Center Representation](image)

**Selecting the value of \( k \)**

- Used a second measure to compute the best value of \( k \).
- Randomly select a value of \( k \) from 1 to 5 and apply the clustering algorithm.
- Minimize:

\[
\sum_{i=1}^{n} \text{dist}(d_i, \cdot - \text{clusterCenter}) \times k^2
\]

**Selecting the value of \( k \)**

- Selected the best \( k \) across 30 trials each.
- The basic k-means never chose the correct value for \( k \).
- The COP-KMEANS selected the correct value for \( k \) for all but one road segment.

**The Data**

- 20 data sets, i.e. 20 road segments.
- Different number of lanes in each segment – i.e. different \( k \) for each data set.
- Number of data points for each segment ranging from 115 to 1160.
- Significantly larger than previous experiment.
Results

• Average accuracy of unconstrained k-means: 58.0%.
• Un-constrained k-means attained a maximum accuracy of 75%.
• Selected the wrong value for \( k \) for all sets.

• Average accuracy of constrained k-means: 98.6%.
• Constrained k-means attained 100% accuracy for 17 out of the 20 data sets.
• Selected the wrong value for \( k \) for one set.

Results

• Experiment specifying the correct value of \( k \) to the unconstrained k-means on data set #6 showed that it still performs poorly.
• Seeks compact spherical clusters.
• Clusters formed span multiple lanes.

Conclusions

• General method to incorporate background knowledge in clustering by using instance level constraints.
• Successfully applied to a real world problem.
• Scalable to large data sets.

More Conclusions

• Might be argued that k-means is fundamentally a poor choice of algorithm for the task.
• The constraints by themselves do not achieve good clustering.
• Combination of the constraints and a poor clustering algorithm can boost its performance.

Order-sensitive Clustering

• A downside of the method is that the clustering is sensitive to the order of assignment of points to the clusters.
• A poor decision at the start can result in can result in poor clusters or “no possible clusters” later.
• Ideally, backtracking could be incorporated in the latter case.

Related Work – Some Other Techniques

• Some agglomerative clustering algorithms use contiguity constraints.
• These cover the entire data set and cannot handle partial constraints.
• No accommodation for constraints to separate data items.
Related Work

- K-means can evolve empty clusters.
- This will result in fewer than $k$ clusters.
- By imposing a minimum size on each cluster, this can be avoided.
- This is a cluster level constraint.
- Like instance level constraints, cluster level constraints can be used to incorporate domain knowledge.

Putting to the Test

- Using constrained k-means in text clustering – clustering contexts with the same sense of a target word.
- Using background knowledge from sources like dictionaries and statistical information from large corpora to generate constraints.
- Using background knowledge to select the initial $k$ points.
Constrained K-means clustering with background knowledge

By: Kiri Wagstaff, Claire Cardie
Seth Rogers, Stefan Schroedl

Presented by: Siddharth Patwardhan
Comments by: Sachin Sharma

Soft Constraints (preferences)…

- Previously defined two *Hard Constraints*
  - must link
  - cannot link
- Augmenting strength factor to each relation
  ⇒ soft constraints or preferences
  e.g. \(<d_i, d_k, s>\)
  \[ 0 \leq |s| \leq 1 \]

Soft constraints, cont…

- Subsumes both *soft* and *hard* constraints
  - \(s\), +ve values: group together
    -ve values: don’t group together
  
  \(<d_i, d_k, 1>\) == must link hard constraint
  \(<d_i, d_k, -1>\) == cannot link hard constraint
  \(<d_i, d_k, 0>\) == don’t care hard constraint

Soft constraint closure

```plaintext
for all i, j, k : given produce
\[ d_i =_{m} d_j \] and \[ d_j =_{m} d_k \] \[ d_i =_{m} d_j \] and \[ d_j \neq_{c} d_k \]
\[ d_i \neq_{c} d_j \] and \[ d_j =_{m} d_k \] \[ d_i \neq_{c} d_j \] and \[ d_j \neq_{c} d_k \]
```

If both constraints negative .... conclusion ?...?
Constrained K-means Clustering with background Knowledge

Presented by Siddhartha Patwardhan

Comments
Sweta Sinha

Overview

• Integration of Background knowledge in constrained K-means clustering
• Background knowledge incorporated in the form of instance level constraints
• Variant of k-means algorithm
• Significant improvements in accuracy

Evaluation Method

• Dataset used for evaluation has label for each instance
• Rand index used to calculate measure of agreement between cluster obtained and the actual classification

Let us take dataset with 6 instances \{a,b,c,d,e,f\}
Clusters by clustering algorithm \{(a,b,c), (d,e,f)\}
Actual classification \{(a,b), (c,d,e), (f)\}

<table>
<thead>
<tr>
<th>Point pair</th>
<th>ab</th>
<th>bc</th>
<th>cd</th>
<th>de</th>
<th>ef</th>
<th>ac</th>
<th>bd</th>
<th>ce</th>
<th>df</th>
</tr>
</thead>
<tbody>
<tr>
<td>together</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>separate</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>mixed</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total no of pairs = n*(n-1)/2 = 15
Similarity = (2+7)/15 = 0.6