Active Learning with Clustering

Zalán Bodó, Zsolt Minier, Lehel Csató

Faculty of Mathematics and Computer Science, Babeș–Bolyai University, Cluj-Napoca, Romania
Outline

Active learning

The proposed algorithm

Spectral clustering

   KASP
   Clustering with a hyperplane

Spectral graph transducer

Learning with support vector machines

The method used in the AL challenge

Experiments
Active learning

- reducing the number of examples for learning methods by selecting the most informative points

- AL scenarios:
  - pool-based: labeled/unlabeled data set ($l << u$); selection by evaluating all the unlabeled points
  - query synthesis: de-novo query generation
  - stream-based selective sampling: successive and independent evaluation of the examples

- AL query strategies [Set09]:
  - uncertainty sampling
  - query by committee
  - expected model change
  - expected error reduction
  - variance reduction
  - density-weighted methods
The proposed algorithm

- based on spectral clustering [SM00] and spectral graph transducer (SGT) [Joa03]
- using the fact, that normalized spectral clustering is clustering with a hyperplane [RR04]
- uses the output of spectral clustering and graph-based transductive learning algorithms to determine the informativeness of a point
Spectral clustering

- two popular spectral clustering methods: ratio cut and normalized cut

\[
\text{rcut}(A_1, A_2) = \sum_{i=1}^{2} \frac{\text{cut}(A_i, \overline{A_i})}{|A_i|}
\]
\[
\text{ncut}(A_1, A_2) = \sum_{i=1}^{2} \frac{\text{cut}(A_i, \overline{A_i})}{\text{vol}(A_i)}
\]

- relaxed version of normalized cuts:

\[
\begin{align*}
\min_{\mathbf{y}} \quad & \mathbf{y}'\mathbf{L}\mathbf{y} \\
\text{s.t.} \quad & \mathbf{y}'\mathbf{D}\mathbf{y} = 1 \text{ and } \mathbf{y}'\mathbf{D}\mathbf{1} = 0
\end{align*}
\]

- solution: \( \mathbf{y}^* = \mathbf{D}^{-1/2} \mathbf{v}_2 \), where
  - \( \mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \)
  - \( \mathbf{v}_2 = \) eigenvector of the second smallest eigenvalue of \( \mathbf{L} \)
- crisp clusters: threshold \( \mathbf{y}^* \)
**Active Learning with Clustering**

Zalán Bodó, Zsolt Minier, Lehel Csató

**Active learning**

The proposed algorithm

Spectral clustering

KASP

Clustering with a hyperplane

Spectral graph transducer

Learning with support vector machines

The method used in the AL challenge

**Experiments**

K-means-based approximate spectral clustering (KASP)

- spectral clustering has time and space complexity of $O(N^3)$ and $O(N^2)$, respectively
- difficult to apply on large data sets
- approximation: reduce the number of points considered for clustering without losing too many of the characteristic features of the original data set
- KASP [YHJ09]:

---

**Algorithm 1**

1. Perform $k$-means clustering on the whole data set.
2. Consider the output of $k$ centers as the representative points.
3. Run a spectral clustering algorithm on the representative points.
4. Assign each original data point to the cluster that was assigned to the center of the clusters computed with the $k$-means algorithm.
Clustering with a hyperplane

- spectral clustering is clustering with a hyperplane \( \{ x \mid w'x = 0 \} \) [RR04] maximizing the gap:
  \[
  \| w' \Phi D^{-1/2} \|_2^2
  \]

- the cluster indicator function is:
  \[
  f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)
  \]

where
- \( \alpha = D^{-1/2} v_2 \)
- \( k(x, y) = \phi(x)'\phi(y) \) is the kernel function
Spectral graph transducer

- constrained spectral graph clustering with explicit label constraints
- proposed in [Joa03]
- we modified it to use normalized cuts and obtained a similar optimization problem to the one described in [Joa03]

\[
\begin{align*}
\min_z & \quad z' \left( L + c D^{-1/2} C D^{-1/2} \right) z \\
& -2cz' D^{-1/2} C \gamma \\
\text{s.t.} & \quad z' z = 1 \\
& \quad z' D^{1/2} 1 = 0
\end{align*}
\]

where

- \( z = D^{1/2} y \), \( y \) is the cluster indicator
- \( L \) is the symmetric normalized graph Laplacian
- \( \gamma \) contains the label constraints, \( \gamma_i = \pm 1 \) for labeled and 0 for unlabeled points
- \( C \) diagonal matrix with nonzero positive weights for the labeled points
since SGT can be considered as spectral clustering + a quadratic constraint ⇒ it also clusters with a hyperplane

Figure: Separating hyperplanes – thick lines – for the two moons data set containing two labeled examples: (a) Normalized spectral clustering; (b) Normalized spectral graph transducer.
Learning with support vector machines

- binary hyperplane-based classification with maximum margin in different feature spaces [BGV92]

\[
\min_{w,b,\xi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} \xi_i \\
\text{s.t.} \quad y_i (w'x + b) \geq 1 - \xi_i, \xi_i \geq 0 \\
i = 1, \ldots, \ell
\]

- common kernels:

\[
k_{\text{poly}}(x,z) = (ax'z + b)^c
\]

\[
k_{\text{rbf}}(x,z) = \exp \left( -\frac{1}{2\sigma^2} \|x - z\|^2 \right)
\]

- decision function:

\[
f(x) = \sum_{i=1}^{\ell} \alpha_i^* y_i k(x_i, x) + b^*
\]
The method used in the AL challenge

Algorithm 2

1. Perform $k$-means-based approximate spectral clustering on the whole data set.
2. **if** $(\ell < \theta)$ **then**
   3. Perform $k$-means on the unlabeled and test data.
   4. Form the new data set from the labeled points and the centers of the clusters.
   5. Perform SGT on the new data set.
3. **else** $(\ell \geq \theta)$
   6. Perform $k$-means on the unlabeled and test data.
   7. Perform $k$-means on the labeled data separately in each of the two classes.
   8. Form the new data set from the centers of the obtained clusters.
10. **end if**
11. Perform bagging with linear SVMs.
Experiments

- the methods used in the experiments:
  - ALG1 – the algorithm described on the previous slide, i.e. the method which starts with normalized spectral clustering when only the label of one point is known, then uses a normalized spectral graph transducer, and finally bagging with linear SVMs
  - ALG2 – the simplified version of the previous algorithm which initially uses normalized spectral clustering and then requests all the training labels and uses bagging with linear SVMs
  - SVM – the simple and yet effective algorithm using linear SVMs

- we used the following data sets:

<table>
<thead>
<tr>
<th>Data set</th>
<th>Domain</th>
<th>Features</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALEX</td>
<td>Toy data set</td>
<td>11</td>
<td>5000</td>
</tr>
<tr>
<td>IBN_SINA</td>
<td>Handwriting recognition</td>
<td>92</td>
<td>10361</td>
</tr>
<tr>
<td>NOVA</td>
<td>Text categorization</td>
<td>16969</td>
<td>9733</td>
</tr>
<tr>
<td>A</td>
<td>Handwriting recognition</td>
<td>92</td>
<td>17535</td>
</tr>
<tr>
<td>D</td>
<td>Text categorization</td>
<td>12000</td>
<td>10000</td>
</tr>
<tr>
<td>F</td>
<td>Ecology</td>
<td>12</td>
<td>67628</td>
</tr>
</tbody>
</table>
Active Learning with Clustering

Zalán Bodó, Zsolt Minier, Lehel Csató

Active learning

The proposed algorithm

Spectral clustering

KASP

Clustering with a hyperplane

Spectral graph transducer

Learning with support vector machines

The method used in the AL challenge

Experiments

Figure: ROC curves for development datasets: ALEX, IBN_SINA and NOVA.

Table: Exact results (AUC/ALC) obtained for the development data sets with algorithms ALG1, ALG2 and SVM.
Active Learning with Clustering

Zalán Bodó, Zsolt Minier, Lehel Csató

Active learning

The proposed algorithm

Spectral clustering

KASP

Clustering with a hyperplane

Spectral graph transducer

Learning with support vector machines

The method used in the AL challenge

Experiments

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>D</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALG2</td>
<td>84.96/4.19</td>
<td>96.41/86.10</td>
<td>96.27/38.07</td>
</tr>
</tbody>
</table>

Table: Results obtained in the challenge (AUC/ALC) using the simplified algorithm (ALG2)

- $W$ is computed using the Gaussian similarity; $\sigma$ was set as the mean norm of the feature vectors in the data set
- NOVA and D: principal component analysis using $k = 50$
- NOVA and D: $tfidf$ transformation [BYRN99]
- implemented in MATLAB using the sample code provided for the challenge
- used LIBSVM [CL01] and fast $k$-means [Elk03]
References


