Learning a Degree-Augmented Distance Metric from a Network

Bert Huang, U of Maryland
Blake Shaw, Foursquare
Tony Jebara, Columbia U

Beyond Mahalanobis: Supervised Large-Scale Learning of Similarity
NIPS Workshop, Sierra Nevada, Dec. 16, 2011
Motivation: Similarity in Networks

- Homophily occurs in natural networks: neighbors are similar
- Learning must account for structural nature of networks
Outline

• Problem formulation
• Structure Preserving Metric Learning
• Degree Distributional Metric Learning
• Analysis
• Experiments
Problem Formulation

• Given: node feature matrix $X$ and adjacency matrix $A$

• Learn the inherent distance metric related to the homophily of the network
Structure Preserving Metric Learning

- Connectivity algorithms: k-nn, b-matching, MST, $\epsilon$-neighborhoods
- Distances are *structure-preserving* if the connectivity algo outputs the true connectivity [SJ09]
- Parameterize distances

$$D_M(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$$
Structured Prediction Motivation

- Constraints: true $A$ must score higher than any feasible adjacency matrix $\tilde{A}$
- Frobenius regularization $\rightarrow$ SVM
- Cutting plane doesn’t scale: requires iterating SDP and separation oracle
- Relaxation:
  - (Optional) drop PSD constraint
  - only consider small changes to $A$
Stochastic SPML (k-nn)

• Consider only changes along node-neighbor-impostor triplets
  \[ T = \{ (i, j, k) | A_{ij} = 1, A_{ik} = 0 \} \]

• Difference between scores is only along triplet edges
  \[
  \frac{\lambda}{2} \| M \|_F^2 + \frac{1}{|T|} \sum_{(ijk) \in T} h(D_M(x_i, x_j) - D_M(x_i, x_k) + 1)
  \]

• Randomly sample triplets and follow stochastic subgradients

• (Periodically project to PSD)
Out-of-Sample Extension

• Connectivity algorithm is fixed: structural parameters must be known
  • e.g., $k =$ degree of training nodes
• What degree should new nodes have?
  • Feature-dependent degree preference functions
Degree Distributional Metric Learning

- Simultaneously learn feature-dependent \textit{degree preference functions} such that the connectivity algorithm maximizes

\[ F(A|X; M, S) = - \sum_{ij} A_{ij} D_M(x_i, x_j) + \sum_i g(c[i]|x_i; S) \]

- Linear deg. pref. score \[ g(k|x; S) = \sum_{k' = 1}^{k} x^\top s_{k'} \]

- Regularizing w/ \( \|S\|_F^2 \), DDML can also be a structural SVM
Stochastic DDML

- Triplet-based loss function:
  \[
  \min_{M,S} \frac{\lambda}{2} (\|M\|^2 + \|S\|^2) + \frac{1}{|T|} \sum_{ijk \in T} h(F(A|X; M, S) - F(A^{(ijk)}|X; M, S) + 1)
  \]

- Score difference cancels except for four quantities:
  \[
  F(A|X; M, S) - F(A^{(ijk)}|X; M, S) =
  D_M(x_i, x_k) - D_M(x_i, x_j) + x_k^T s(c[k]+1) - x_j^T s(c[j]-1)
  \]

- (Project toward concave degree prefs)
Learner Running Time

- Limit the maximum degree so the number of parameters for degree preference function is constant

- Subgradient computation:
  \( O(d^2) \) for SPML and \( O(d^2 + c_{\text{max}}) \) for DDML

- Learner reduces to PEGASOS algorithm (Shalev-Shwartz et al. ’07) on one-class SVM:

\[
O \left( \frac{1}{\epsilon \lambda} \right) \quad \text{time for } \epsilon\text{-convergence}
\]
For concave degree preferences, the connectivity algorithm reduces to an \( O(N^3) \) combinatorial algorithm (Huang & Jebara, ’09).

Or rank edges by degree-augmented distance:

\[ D_M(b, c) \]
\[ g(b, 2) - g(b, 1) \]
\[ g(c, 3) - g(c, 2) \]
Experiments

- Wikipedia: word counts and hyperlinks
- Facebook: status, gender, major, dorm, year, and friendship links [Traub, et al., '11]
- Randomly hold out 20% of nodes and incident edges for testing
Experiment Results

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>$m$</th>
<th>$d$</th>
<th>Euclid.</th>
<th>RTM</th>
<th>SVM</th>
<th>SPML</th>
<th>DDML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Theory</td>
<td>223</td>
<td>917</td>
<td>6695</td>
<td>0.624</td>
<td>0.591</td>
<td>0.610</td>
<td>0.722</td>
<td>0.691*</td>
</tr>
<tr>
<td>Philosophy Concepts</td>
<td>303</td>
<td>921</td>
<td>6695</td>
<td>0.705</td>
<td>0.571</td>
<td>0.708</td>
<td>0.707</td>
<td>0.746*</td>
</tr>
<tr>
<td>Search Engines</td>
<td>269</td>
<td>332</td>
<td>6695</td>
<td>0.662</td>
<td>0.487</td>
<td>0.611</td>
<td>0.742</td>
<td>0.725*</td>
</tr>
<tr>
<td>Philosophy Crawl</td>
<td>100k</td>
<td>4m</td>
<td>7702</td>
<td>0.547</td>
<td>–</td>
<td>–</td>
<td>0.601</td>
<td>0.562</td>
</tr>
<tr>
<td>Harvard</td>
<td>1937</td>
<td>48k</td>
<td>193</td>
<td>0.764</td>
<td>0.562</td>
<td>0.839</td>
<td>0.854</td>
<td>0.848</td>
</tr>
<tr>
<td>MIT</td>
<td>2128</td>
<td>95k</td>
<td>173</td>
<td>0.702</td>
<td>0.494</td>
<td>0.784</td>
<td>0.801</td>
<td>0.797</td>
</tr>
<tr>
<td>Stanford</td>
<td>3014</td>
<td>147k</td>
<td>270</td>
<td>0.718</td>
<td>0.532</td>
<td>0.784</td>
<td>0.808</td>
<td>0.810</td>
</tr>
<tr>
<td>Columbia</td>
<td>3050</td>
<td>118k</td>
<td>251</td>
<td>0.717</td>
<td>0.519</td>
<td>0.796</td>
<td>0.818</td>
<td>0.821</td>
</tr>
</tbody>
</table>
Summary + Open Problems + Thanks!

- SPML: metrics consistent with structural behavior of networks
- DDML: explicit degree preference functions that are feature-dependent
- Linear constraints and Frobenius regularization: constant time convergence
- Other applications of structure-preserving metric
- More natural regularizer?
- Stopping criterion
- Large-scale prediction

Thanks!