Learning a Distance Metric from a Network: Appendix

Blake Shaw[*]
Computer Science Dept.
Columbia University
blake@cs.columbia.edu

Bert Huang[*]
Computer Science Dept.
Columbia University
bert@cs.columbia.edu

Tony Jebara
Computer Science Dept.
Columbia University
jebara@cs.columbia.edu

A Degree Distributional Metric Learning

While SPML using k-nearest neighbors learns a structure preserving metric, one of its limitations is in predicting full graphs in an out-of-sample setting. On the training data, the degree of each node is known, so the connectivity algorithm connects the exact number of neighbors as necessary to reconstruct the input graph. On a new set of nodes, however, the target degree is unknown. One method to address this is to learn a non-stationary *degree preference function* over node features that relates the features of a node to its target degree.

As one possible variant to *structure preserving metric learning* (SPML), *degree distributional metric learning* (DDML) simultaneously learns a metric while also learning a parameterized, non-stationary degree preference function used to compute the connectivity of nodes. This extension can be understood as SPML with an adaptive connectivity algorithm, rather than the default k-nearest neighbors.

The connectivity algorithm uses a degree preference function g, which takes a node's feature vector $\mathbf{x}$ and a target degree k, and is parameterized by matrix $\mathbf{S} \in \mathbb{R}^{d \times n}$. The score is then computed via

$$g(k|\mathbf{x};\mathbf{S}) = \sum_{k'=1}^{k} \mathbf{x}^\top \mathbf{s}_{k'}.$$

The score of a graph A is then the sum of all edge distances and the degree preference functions for each node

$$F(\mathbf{A}|\mathbf{X};\mathbf{M},\mathbf{S}) = \sum_{ij} A_{ij} D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - \sum_i g\left(\sum_j A_{ij}|\mathbf{x}_i;\mathbf{S}\right).$$

The objective for DDML is otherwise analogous to that of SPML:

$$f(\mathbf{M}) = \frac{\lambda}{2}||\mathbf{M}||^2 - \sum_{\tilde{\mathbf{A}} \in \mathbb{B}^{n \times n}} \max(F(\mathbf{A}|\mathbf{X};\mathbf{M},\mathbf{S}) - F(\tilde{\mathbf{A}}|\mathbf{X};\mathbf{M},\mathbf{S}) + \Delta(\mathbf{A},\tilde{\mathbf{A}}), 0),$$

where Δ denotes Hamming distance. This objective is solvable via the cutting-plane style optimization by iteratively finding the worst-violating $\tilde{\mathbf{A}}$ and adding it to a constraint set. For concave degree preference functions, the worst-violated constraint can be found by converting the problem to a maximum weight b-matching on an augmented graph [1], thus an additional concavity constraint on g is added to the optimization.

A similar approach to the stochastic SPML algorithm is also possible to perform DDML much faster, and, by parameterizing the degree preference function only up to a fixed maximum degree, also eliminates the dependence of the running time on the size of the graph. As in stochastic SPML, a DDML objective can be written in terms of triplets of nodes i, neighbor j, disconnected node

[*]Blake Shaw is currently at Foursquare, and Bert Huang is currently at the University of Maryland.

triplets k. Let $\mathbf{A}^{(i,j,k)}$ denote the false graph produced by toggling the edge between nodes i and j and the edge between nodes i and k. The DDML objective using the triplet-style constraints is

$$f^{\deg}(\mathbf{M}) = \frac{\lambda}{2}||\mathbf{M}||^2 - \frac{1}{|S|}\sum_{(i,j,k)\in S}\max(F(\mathbf{A}|\mathbf{X};\mathbf{M},\mathbf{S}) - F(\mathbf{A}^{(i,j,k)}|\mathbf{X};\mathbf{M},\mathbf{S}) + 1, 0).$$

The difference in scores decomposes into four scalar values, since the only differences changing $\mathbf{A}$ to $\mathbf{A}^{(i,j,k)}$ are that $\mathbf{A}^{(i,j,k)}$ is missing edge (i,j), gains edge (i,k), the degree of node j decreases by one and the degree of node k increases by one. Thus, the difference can be computed by evaluating the distance from node i to node j, the distance from node i to node k, the change in degree preference score from the degree of node j to its degree minus one, and the change in degree preference from the degree of node k from its degree plus one. Let the degrees of all nodes be stored in array c, such that the degree of node j is $c[j]$. The difference is then computable as

$$F(\mathbf{A}|\mathbf{X};\mathbf{M},\mathbf{S}) - F(\mathbf{A}^{(i,j,k)}|\mathbf{X};\mathbf{M},\mathbf{S}) = D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k) + \mathbf{x}_j^\top \mathbf{s}_{(c[j]-1)} - \mathbf{x}_k^\top \mathbf{s}_{(c[k]+1)}.$$

This formulation eliminates the need for the expensive separation oracle and allows stochastic optimization. The gradient update for the metric parameter $\mathbf{M}$ is the same as in SPML. The gradient with respect to $\mathbf{s}_{(c[j]-1)}$ is $\mathbf{x}_j$ and the gradient with respect to $\mathbf{s}_{(c[k]+1)}$ is $(-\mathbf{x}_k)$.

To retain coherence between the different degree functions, we add a requirement that the resulting degree preference function for each node is concave. One way to enforce concavity is by stochastically sampling a node i per iteration, and projecting $\mathbf{S}$ such that entries in $\mathbf{x}_i^\top \mathbf{S}$ are in decreasing order.

The pseudocode for stochastic DDML is in Algorithm 4.

Algorithm 4 Stochastic degree distributional metric learning

Input: $\mathbf{A} \in \mathbb{B}^{n\times n}$, $\mathbf{X} \in \mathbb{R}^{d\times n}$, and parameters λ, T, B

$\mathbf{M}_1 \leftarrow \mathbf{I}_d$, $\mathbf{S}_1 \leftarrow \mathbf{0}_{d,n}$
Compute degree array c s.t. $c[i] = \sum_j A_{ij}, \forall i$
for t from 1 to $T-1$ **do**
  $\eta_t \leftarrow \frac{1}{\lambda t}$
  $\mathbf{C} \leftarrow \mathbf{0}_{n,n}$
  $\mathbf{S}' \leftarrow \lambda \mathbf{S}$
  for b from 1 to B **do**
    $(i,j,k) \leftarrow$ Sample random triplet from $S = \{(i,j,k) \mid A_{ij} = 1, A_{ik} = 0\}$
    if $F(\mathbf{A}|\mathbf{X};\mathbf{M}_t,\mathbf{S}_t) - F(\mathbf{A}^{(i,j,k)}|\mathbf{X};\mathbf{M}_t,\mathbf{S}_t) + 1 > 0$ **then**
      $\mathbf{C}_{jj} \leftarrow \mathbf{C}_{jj} + 1$, $\mathbf{C}_{ik} \leftarrow \mathbf{C}_{ik} + 1$, $\mathbf{C}_{ki} \leftarrow \mathbf{C}_{ki} + 1$
      $\mathbf{C}_{ij} \leftarrow \mathbf{C}_{ij} - 1$, $\mathbf{C}_{ji} \leftarrow \mathbf{C}_{ji} - 1$, $\mathbf{C}_{kk} \leftarrow \mathbf{C}_{kk} - 1$
      $\mathbf{s}'_{c[j]} \leftarrow \mathbf{s}'_{c[j]} + \mathbf{x}_j$
      $\mathbf{s}'_{c[k]} \leftarrow \mathbf{s}'_{c[k]} - \mathbf{x}_k$
    end if
  end for
  $\nabla_t \leftarrow \mathbf{X}\mathbf{C}\mathbf{X}^\top + \lambda \mathbf{M}_t$
  $\mathbf{M}_{t+1} \leftarrow \mathbf{M}_t - \eta_t \nabla_t$
  $\mathbf{S}_{t+1} \leftarrow \mathbf{S}_t - \eta_t \mathbf{S}'$
  $i \leftarrow$ Sample random index
  Project $\mathbf{S}$ so $\mathbf{x}_i^\top \mathbf{S}$ is monotonically nonincreasing
  Optional: $\mathbf{M}_{t+1} \leftarrow [\mathbf{M}_{t+1}]^+$ {Project onto the PSD cone}
end for
return $\mathbf{M}_T$

Experiments Using DDML on the same Wikipedia experiments from the main paper, we score comparable AUC to SPML. On "graph theory", "philosophy concepts", and "search engines", DDML scores AUCs of 0.691, 0.746, and 0.725. While these scores are quite close to those of SPML, the DDML variant provides a tradeoff between running time and model richness. In the case of the Wikipedia category "philosophy concepts", DDML even provides a performance improvement, which may indicate a clear signal in degree preference learnable from the word counts.

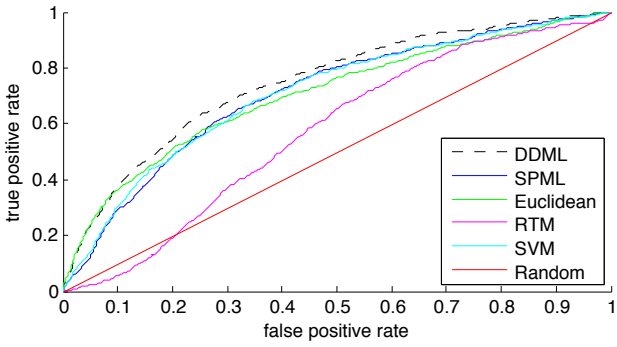


Figure 4: ROC curve for various algorithms on the "philosophy concepts" category.

B Low-rank structure preserving metric learning

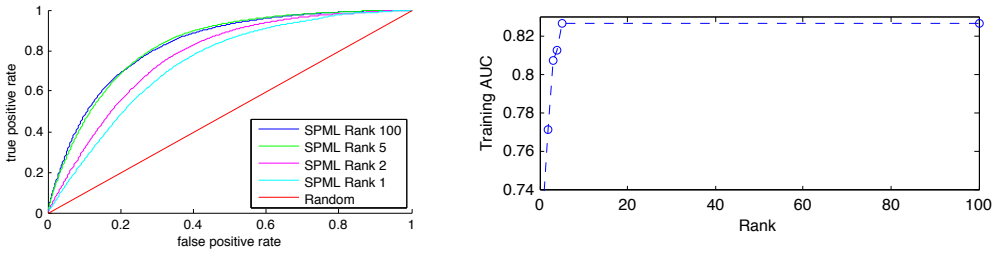
In this section, we present the low-rank variant of SPML first introduced in Section 2.4. The low-rank variant computes all distances using a factorization $\mathbf{L} \in \mathbb{R}^{r \times d}$ of $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$, eliminating the need to compute a $d \times d$ matrix. Existing metric learning algorithms use similar low-rank factorizations [2]. Low-rank SPML has an additional parameter r, which limits the rank of $\mathbf{M}$ by explicitly determining the size of $\mathbf{L}$. The optional projection onto the PSD cone is no longer necessary because $\mathbf{L}^\top \mathbf{L}$ always forms a valid metric by construction. This optimization not convex, but initial experimental results seem to show that the stochastic optimization avoids local minima in practice. Algorithm 5 details the steps of low-rank SPML.

Algorithm 5 Low-rank structure preserving metric learning with nearest neighbor constraints and optimization with projected stochastic subgradient descent

Input: $\mathbf{A} \in \mathbb{B}^{n \times n}$, $\mathbf{X} \in \mathbb{R}^{d \times n}$, and parameters λ, T, B, r

$\mathbf{L}_1 \leftarrow \text{rand}(r, d)$ {Initialize $\mathbf{L}$}
for t from 1 to $T - 1$ **do**
  $\eta_t \leftarrow \frac{1}{\lambda t}$
  $\mathbf{C} \leftarrow \mathbf{0}_{n,n}$
  for b from 1 to B **do**
    $(i, j, k) \leftarrow$ Sample random triplet from $S = \{(i, j, k) \mid A_{ij} = 1, A_{ik} = 0\}$
    if $||\mathbf{L}_t \mathbf{x}_i - \mathbf{L}_t \mathbf{x}_j||^2 - ||\mathbf{L}_t \mathbf{x}_i - \mathbf{L}_t \mathbf{x}_k||^2 + 1 > 0$ **then**
      $\mathbf{C}_{jj} \leftarrow \mathbf{C}_{jj} + 1, \mathbf{C}_{ik} \leftarrow \mathbf{C}_{ik} + 1, \mathbf{C}_{ki} \leftarrow \mathbf{C}_{ki} + 1$
      $\mathbf{C}_{ij} \leftarrow \mathbf{C}_{ij} - 1, \mathbf{C}_{ji} \leftarrow \mathbf{C}_{ji} - 1, \mathbf{C}_{kk} \leftarrow \mathbf{C}_{kk} - 1$
    end if
  end for
  $\nabla_t \leftarrow 2\mathbf{X}\mathbf{C}\mathbf{X}^\top \mathbf{L}_t^\top + \lambda \mathbf{L}_t$
  $\mathbf{L}_{t+1} \leftarrow \mathbf{L}_t - \eta_t \nabla_t$
end for
return $\mathbf{L}_T$

We run low-rank SPML on the Harvard Facebook data, fixing $\lambda = 1\text{e} - 5$ and varying the rank parameter r. The ROC curves and AUC scores using training data for different ranks are in Figure 5. With greater rank, SPML has more flexibility to construct a metric that fits the training data, but lower rank provides a tradeoff between efficiency and reconstruction quality. It is clear from this dataset that a rank of $r = 5$ is sufficient to represent the structure preserving metric, while reducing the number of parameters from d^2 =37,249 to $d \times r = 965$. Training fewer parameters requires less time, and allows low-rank SPML to handle large-scale networks with many nodes and high-dimensional features.



(a) Training ROC curve for different ranks

(b) Training AUC as a function of rank

Figure 5: Performance of low-rank SPML on training data varying the rank parameter, run on a single Facebook school. The results imply that a significantly smaller rank than the true feature dimensionality is sufficient to fit the training data.

References

[1] B. Huang and T. Jebara. Exact graph structure estimation with degree priors. In *ICMLA*, pages 111–118, 2009.

[2] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.