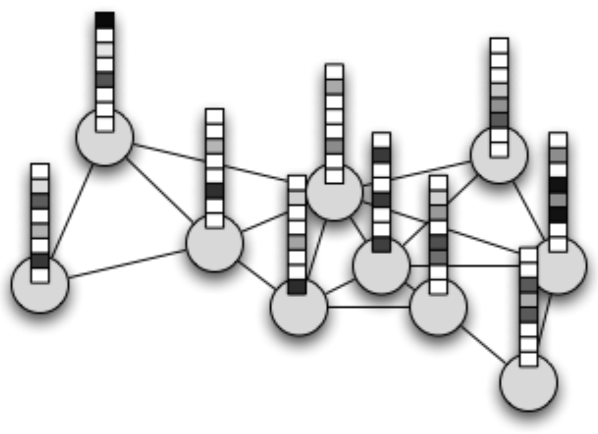
Learning a Distance Metric from a Network

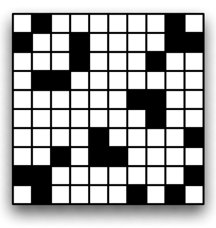
Blake Shaw — Foursquare
Bert Huang — University of Maryland
Tony Jebara — Columbia University

Introduction

Real-world network data often consists of **both** node features and connectivity. Can we learn a metric that relates features to links?



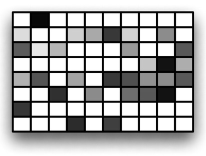
Adjacency Matrix



$\mathbf{A} \in \mathbb{B}^{n\times n}$

+

Node Features

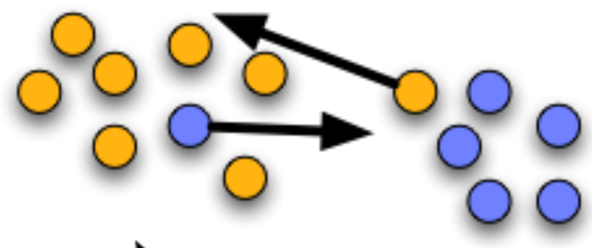


$\mathbf{X} \in \mathbb{R}^{d\times n}$

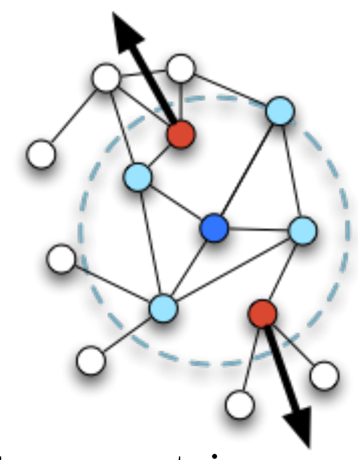
Examples:
Wikipedia: node features are word-counts and links are hyperlinks
Facebook: node features are profiles and links are friendships
Foursquare: node features are places people visit and links are friendships

- While homophily is expected in natural networks, nodes do not simply connect based on similarity of their features alone
- Modeling independent links is insufficient, so one must account for the inherent topology of the network
- We propose learning a distance metric from large social networks that captures relationships between node features and the structure of the network: Structure Preserving Metric Learning (SPML)

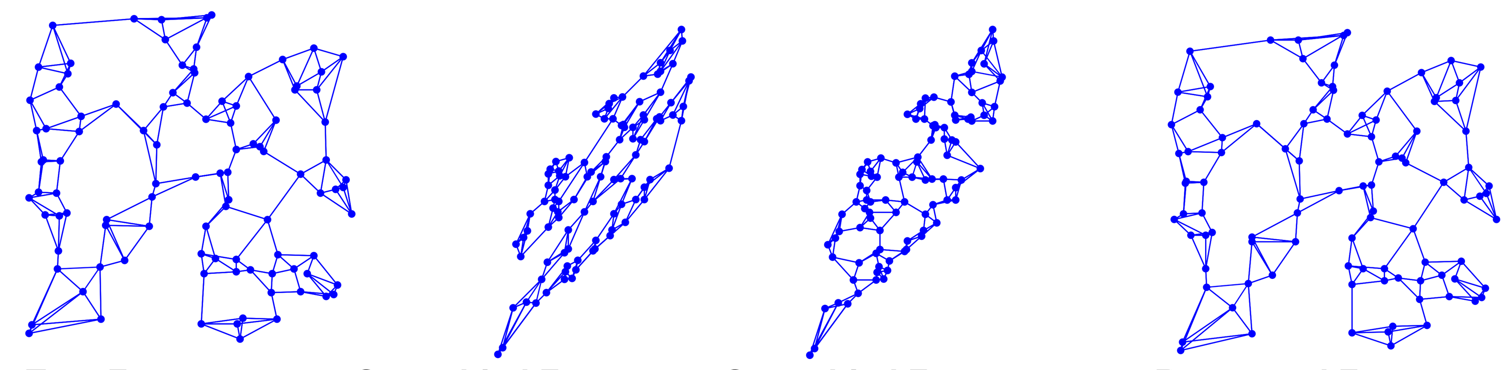
Background



- Current metric learning algorithms are used for supervised tasks like classification [Chechnik et al. '10, Weinberger et al. '10]
- These methods push away "class impostors"
- SPML pushes away "graph impostors"



- Following intuition from Structure Preserving Embedding [Shaw and Jebara '09], SPML finds a metric that is structure preserving
- Given adjacency matrix $\mathbf{A}$ and node features $\mathbf{X}$, a distance metric parametrized by $\mathbf{M} \in \mathbb{R}^{d\times d}$ is **structure preserving** with respect to a connectivity algorithm $\mathcal{G}$ if $\mathcal{G}(\mathbf{X},\mathbf{M}) = \mathbf{A}$



True Features / True Connectivity — Scrambled Features / True Connectivity — Scrambled Features / Implied Connectivity — Recovered Features / Implied Connectivity

Structure Preserving Metric Learning

- Regularizing $\mathbf{M}$ by penalizing its Frobenius norm, SPML objective is a semidefinite program (SDP), which is too expensive for large networks
- Instead we rewrite the objective function over a large set of triplet constraints and optimize via stochastic gradient descent

$$f(\mathbf{M}) = \frac{\lambda}{2}||\mathbf{M}||_F^2 + \frac{1}{|S|}\sum_{(i,j,k)\in S} \max(D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k) + 1, 0),$$

where $S = \{(i,j,k) \mid A_{ij} = 1, A_{ik} = 0\}$

- Using the distance transformation

$$D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{M}\mathbf{x}_i + \mathbf{x}_j^\top \mathbf{M}\mathbf{x}_j - \mathbf{x}_i^\top \mathbf{M}\mathbf{x}_j - \mathbf{x}_j^\top \mathbf{M}\mathbf{x}_i,$$

constraints can be written using a sparse matrix $\mathbf{C}^{(i,j,k)}$, where

$$C_{jj}^{(i,j,k)}, C_{ik}^{(i,j,k)}, C_{ki}^{(i,j,k)} = 1, \quad \text{and} \quad C_{ij}^{(i,j,k)}, C_{ji}^{(i,j,k)}, C_{kk}^{(i,j,k)} = -1$$

- The subgradient of f at $\mathbf{M}$ is then

$$\nabla f = \lambda\mathbf{M} + \frac{1}{|S|}\sum_{(i,j,k)\in S_+} \mathbf{X}\mathbf{C}^{(i,j,k)}\mathbf{X}^\top,$$

where $S_+ = \{(i,j,k) | D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k) + 1 > 0\}$

- We perform stochastic subgradient descent by randomly sampling a mini-batch of triplets at each iteration
- Theorem: This method does not scale with the size of the network, only the desired approximation error!!!

Algorithm 3 Structure preserving metric learning with nearest neighbor constraints and optimization with projected stochastic subgradient descent

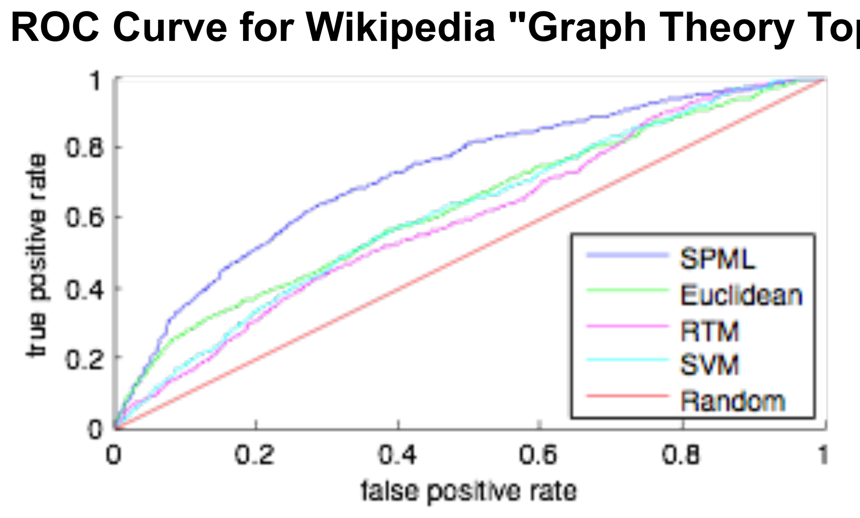
```
Input: A ∈ B^{n×n}, X ∈ R^{d×n}, and parameters λ, T, B
M_1 ← I_d
for t from 1 to T − 1 do
    η_t ← 1/(λt)
    C ← 0_{n,n}
    for b from 1 to B do
        (i, j, k) ← Sample random triplet from S = {(i, j, k) | A_ij = 1, A_ik = 0}
        if D_{M_t}(x_i, x_j) − D_{M_t}(x_i, x_k) + 1 > 0 then
            C_jj ← C_jj + 1, C_ik ← C_ik + 1, C_ki ← C_ki + 1
            C_ij ← C_ij − 1, C_ji ← C_ji − 1, C_kk ← C_kk − 1
        end if
    end for
    ∇_t ← XCX^⊤ + λM_t
    M_{t+1} ← M_t − η_t ∇_t
    Optional: M_{t+1} ← [M_{t+1}]^+ {Project onto the PSD cone}
end for
return M_T
```

Experiments

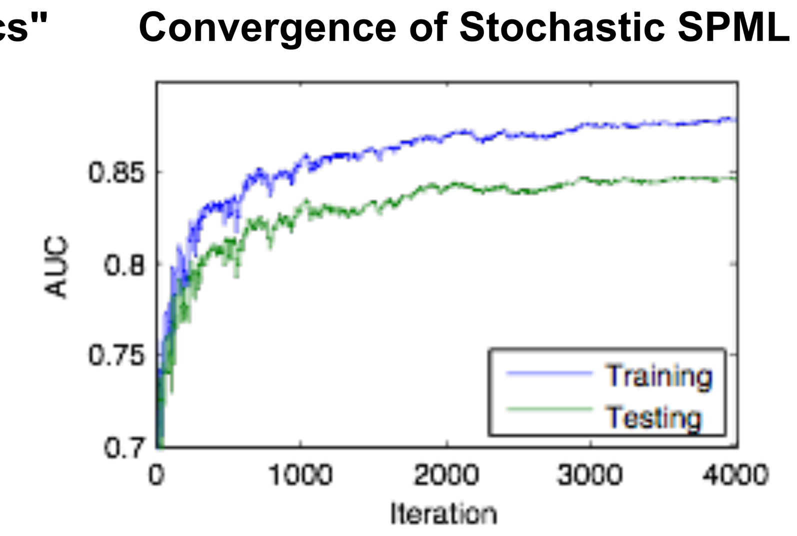
- Learn a metric using 80% of nodes as training and evaluate prediction of links for the held-out 20%, scoring AUC of ranking
- Data sources: Wikipedia data, Facebook data, Foursquare data
- Compare with existing methods: Euclidean distance, Relational Topic Models (RTM), and Support Vector Machines (SVM)

		n	m	d	Euclidean	RTM	SVM	SPML
Wikipedia	Graph Theory	223	917	6695	0.624	0.591	0.610	**0.722**
	Philosophy Concepts	303	921	6695	0.705	0.571	**0.708**	0.707
	Search Engines	269	332	6695	0.662	0.487	0.611	**0.742**
	Philosophy Crawl	100,000	4,489,166	7702	0.547	–	–	**0.601**
FB	Harvard	1937	48,980	193	0.764	0.562	0.839	**0.854**
	MIT	2128	95,322	173	0.702	0.494	0.784	**0.801**
	Stanford	3014	147,516	270	0.718	0.532	0.784	**0.808**
	Columbia	3050	118,838	251	0.717	0.519	0.796	**0.818**
4SQ	Foursquare	83	4322	24082	0.760	0.501	0.710	**0.829**

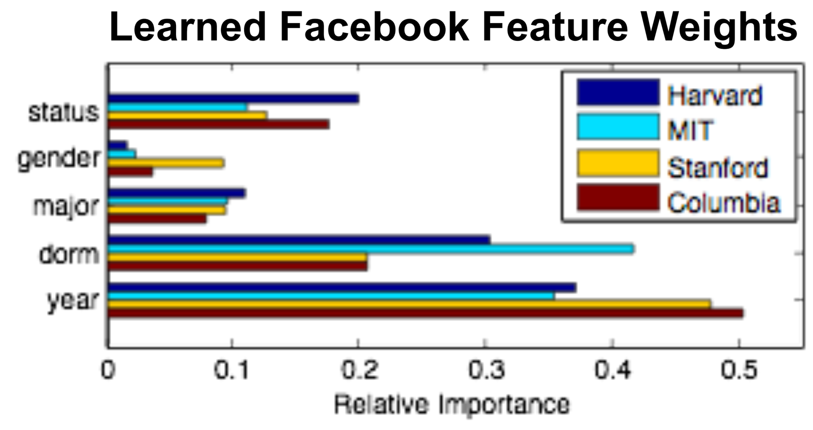
ROC Curve for Wikipedia "Graph Theory Topics"



Convergence of Stochastic SPML



Learned Facebook Feature Weights



Extensions

- To alleviate the cost of optimizing over the full $\mathbf{M}$ matrix in high dimensional problems, we can limit the optimization to allow nonzero entries only along the diagonal of $\mathbf{M}$
- Alternatively, we can optimize a fixed-sized, low-rank factorization of $\mathbf{M}$ by rewriting $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$ and optimizing $\mathbf{L}$
- We can simultaneously learn feature-dependent *degree preference functions* which adds dependency between the node feature and its structural degree (see NIPS '11 workshop talk)

Learning a Degree-Augmented Distance Metric from a Network
Bert Huang, Blake Shaw, Tony Jebara
at NIPS Workshop -- Beyond Mahalanobis: Supervised Large-Scale Learning of Similarity