Learning with Subgraph Estimation and Degree Priors

Bert Huang and Tony Jebara
Computer Science Department
Columbia University, New York, NY 10027
{bert, jebara}@cs.columbia.edu

We present a general tool for finding exact *maximum a posteroiri* (MAP) estimates of graph structure given local edge likelihood and log-concave degree priors. Before this result, only approximate inference was possible for this general class. Many classical subgraph processes, including k-nearest neighbors, b-matching, and ϵ-balls, can be generalized by this framework. Given a data set X, we assume the probability of the resulting adjacency matrix A between the data points factorizes as in the following MAP objective function:

$$\max_A \Pr(X|A)\Pr(A) = \prod_{ij} \Pr(x_i, x_j|A_{ij}) \prod_k \Pr(|\vec{A}_k|),$$

where the first term $\Pr(x_i, x_j|A_{ij})$ represents Bernoulli likelihoods of whether an edge exists between endpoints x_i and x_j, and the second term $\Pr(|\vec{A}_k|)$ is the likelihood of the k'th node's degree. Our algorithm exactly and efficiently solves any problem formulated this way when the degree probability functions are log-concave.

The algorithm finds the MAP solution by augmenting the original graph with a set of auxiliary nodes, which reward or penalize according to the number of edges selected from the original graph. The soft-degree MAP solution then becomes the b-matching solution on the augmented graph. Maximum weight b-matching can be solved using belief propagation with distributable message-passing [1, 2], which has been shown to converge in $\mathcal{O}(N^2)$ time under certain assumptions [5]. Figure 1 illustrates this conversion from soft-degree priors to b-matching.

The degree priors of this framework can be set in various ways to emulate classical subgraph algorithms. For example, k-nearest neighbors limits the indegree of each node to some integer k. Other examples include requiring all nodes to have degree b, which parallels the perfect b-matching algorithm, and placing a linear penalty on the degree of each node, which causes the solution to be a thresholding of the edge weights (a.k.a. ϵ-balls).

Maximum weight b-matching is applied to various classical applications such as advertisement allocation in search engines [4], as well as machine learning applications such as semi-supervised learning [3], and embedding of data and graphs [6, 7]. These existing applications that use degree-based graph sparsification methods may benefit from extending their simple, hard-constrained degree priors to soft priors, which we can now exactly optimize. Machine learning applications may further benefit from exploring their statistical interpretations. Typically, these algorithms are applied during the process of machine learning for heuristic reasons, based on informal intuitions; they are often used to preprocess input data or postprocess predictions, or they are themselves integral parts of the learning algorithms. For example, the reasoning for the singleton edge potentials and the degree priors may be unclear, and a proper statistical treatment may illuminate

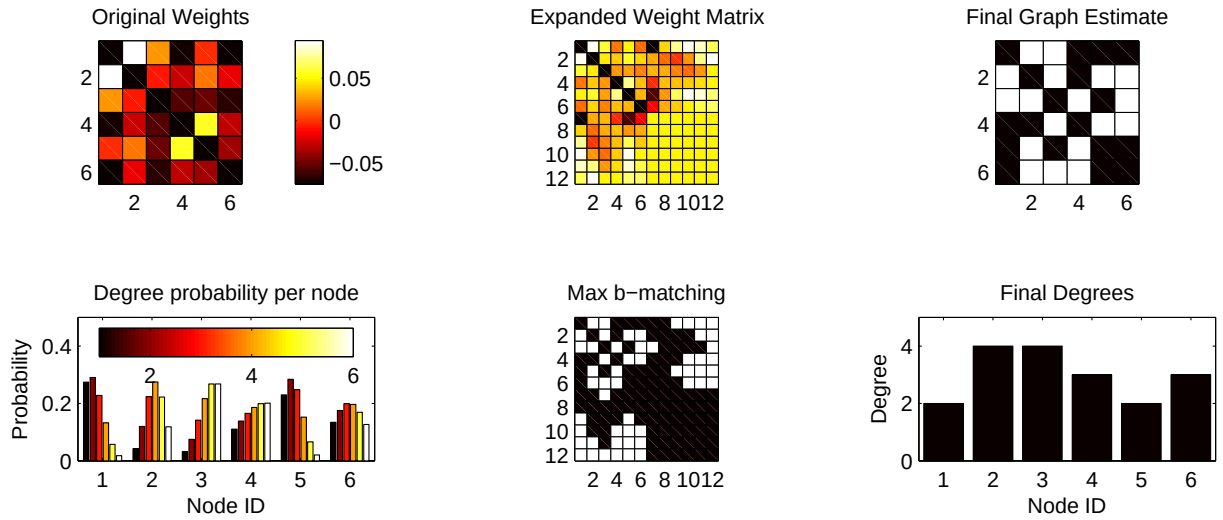

Figure 1: Example of mapping a degree dependent problem to a hard-constrained b-matching. Left: Original weight matrix and row/column degree distributions. The degree distributions are depicted with a bar plot for each node, with the colored bars representing the likelihood of that node having its respective degree. Upper Middle: Weight matrix of expanded graph, whose solution is now constrained to have exactly 6 neighbors per node. Lower Middle: Resulting b-matching, whose upper left quadrant is the final output. Right: MAP solution and final node degrees.

more principled approaches. Future work also includes exploring how to include the true target variable in the estimation process. For example, when performing k-nearest neighbor classification, the labels are ignored during the graph estimation process, which is entirely unsupervised. Only once the k-NN graph is fixed are the maximum likelihood labels computed.

References

[1] M. Bayati, D. Shah, and M. Sharma. Maximum weight matching via max-product belief propagation. In *Proc. of the IEEE International Symposium on Information Theory*, 2005.

[2] B. Huang and T. Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, volume 2 of JMLR: W&CP, March 2007.

[3] T. Jebara, J. Wang, and S.F. Chang. Graph construction and b-matching for semi-supervised learning. In *International Conference on Machine Learning*, 2009.

[4] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5), 2007.

[5] J. Salez and D. Shah. Optimality of belief propagation for random assignment problem. In Claire Mathieu, editor, *SODA*, pages 187–196. SIAM, 2009.

[6] B. Shaw and T. Jebara. Minimum volume embedding. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, volume 2 of JMLR: W&CP, March 2007.

[7] B. Shaw and T. Jebara. Structure preserving embedding. In *International Conference on Machine Learning*, 2009.