
Exponential Family Graph Matching and Ranking

James Petterson, Tibério S. Caetano, Julian J. McAuley and Jin Yu
 NICTA, Australian National University
 Canberra, Australia

Appendix A

For completeness we include a description of the sampling algorithm presented in [1]. The algorithm is an *accept-reject* algorithm. The core idea of such an algorithm is very simple: assume we need to sample from a distribution p in a given domain \mathcal{M} , but that such a task is intractable. Instead, we sample from a distribution q in a *superset* \mathcal{N} of the original domain (in which sampling is easier), whose restriction to the original domain coincides with the original distribution: $q|_{\mathcal{M}} = p$. We then only ‘accept’ those samples that effectively fall within the original domain \mathcal{M} . Clearly, the efficiency of such a procedure will be dictated by (i) how efficient it is to sample from q in \mathcal{N} and (ii) how much mass of q is in \mathcal{M} . Roughly speaking, the algorithm presented in [1] manages to sample perfect matches of bipartite graphs such that both conditions (i) and (ii) are favorable.

The reasoning goes as follows: the problem consists of generating variates $y \in \mathcal{Y}$ (y is a match) with the property that $p(y) = w(y)/Z$, where $w(y)$ is the non-negative score of match y and $Z = \sum_y w(y)$ is the partition function, which in our case is a permanent as discussed in Section 4.1. We first partition the space \mathcal{Y} into $\mathcal{Y}_1, \dots, \mathcal{Y}_I$, where $\mathcal{Y}_i = \{y : y(1) = i\}$. Each part has its own partition function $Z_i = \sum_{y \in \mathcal{Y}_i} w(y)$. Next, a suitable upper bound $U(\mathcal{Y}_i) \geq Z_i$ on the partition function is constructed such that the following two properties hold:¹

$$(P1) \quad \sum_{i=1}^M U(\mathcal{Y}_i) \leq U(\mathcal{Y}).$$

$$(P2) \quad \text{If } |\mathcal{Y}_i| = 1, \text{ then } U(\mathcal{Y}_i) = Z_i = w(y).$$

That is, (i) the upper bound is super-additive in the elements of the partition and (ii) if \mathcal{Y}_i has a single match, the upper bound *equals* the partition function, which in this case is just the score of that match.

Now the algorithm: consider the random variable \mathcal{J} where $p(\mathcal{J} = i) = U(\mathcal{Y}_i)/U(\mathcal{Y})$. By (P1), $\sum_{i=1}^M p(i) \leq 1$, so assume $p(\mathcal{J} = 0) = 1 - \sum_{i=1}^M p(i)$. Now, draw a variate from this distribution, and if $\mathcal{J} = i = 0$, reject and restart, otherwise recursively sample in \mathcal{Y}_i .² This algorithm either stops and restarts or it reaches $\mathcal{Y}_{\text{final}}$ which consists of a match, i.e., $|\mathcal{Y}_{\text{final}}| = 1$. This match is then a legitimate sample from $p(y)$. The reason this is the case is because of (P2), as shown below. Assuming the algorithm finishes after k samples, the probability of the match is the telescopic product

$$\frac{U(\mathcal{Y}_{\mathcal{J}(1)})}{U(\mathcal{Y})} \frac{U(\mathcal{Y}_{\mathcal{J}(2)})}{U(\mathcal{Y}_{\mathcal{J}(1)})} \cdots \frac{U(\mathcal{Y}_{\mathcal{J}(k)})}{U(\mathcal{Y}_{\mathcal{J}(k-1)})} \stackrel{(P2)}{=} \frac{w(y)}{U(\mathcal{Y})}, \quad (1)$$

and since the probability of acceptance is $Z/U(\mathcal{Y})$, we have

$$p(y) = \frac{w(y)/U(\mathcal{Y})}{Z/U(\mathcal{Y})} = \frac{w(y)}{Z}, \quad (2)$$

¹See [1] for details.

²Due to the self-reducibility of permutations, when we fix $y(1) = i$, what remains is also a set of permutations. We then sample $y(2), y(3) \dots y(M)$.

which is indeed the distribution from which we want to sample. For pseudocode and a rigorous presentation of the algorithm, see [1].

References

- [1] Huber, M., & Law, J. (2008). Fast approximation of the permanent for very dense problems. *SODA*.