

**CS599: Structure and Dynamics of Networked Information (Spring 2005)**  
**02/14/2005: Correlation Clustering, Metric Labeling**  
**Scribes: Ramakrishna Gummadi**

During the last lecture, we started the analysis of an LP-rounding based approximation algorithm for minimizing disagreements in correlation clustering. Recall that in correlation clustering [1], we are given a graph each of whose edges is labeled either '+' or '-'. The goal is to partition nodes into clusters so as to minimize the number of '+' edges across clusters plus the number of '-' edges within clusters. (In the maximization version, we want to maximize the number of '+' edges within clusters plus the number of '-' edges across — at optimality, these are the same, but the two objectives differ in how well they can be approximated.)

Although the problem can be defined for arbitrary (weighted) graphs, we considered the LP for the case of a complete graph in which all edge weights are 1. The LP was the following, given in [2].

$$\begin{aligned} \text{Minimize} \quad & \sum_{+(ij)} x_{ij} + \sum_{-(ij)} (1 - x_{ij}) \\ \text{such that} \quad & x_{ik} \leq x_{ij} + x_{jk} \quad \text{for all } i, j, k \\ & x_{ij} \in [0, 1] \quad \text{for all } i, j. \end{aligned}$$

The algorithm repeatedly removed clusters of nodes and their neighbors — see the notes from the previous lecture. We had already covered the case when the cluster that is formed is a singleton cluster. Also, we managed to show that the total cost incurred by the approximation algorithm for '-' edges that were included inside clusters is at most four times the LP cost of the corresponding edges. Here, we verify the bound for '+' edges.

A '+' edge  $(i, j)$  only contributes to the cost of our solution if one endpoint, say,  $i$ , is included in the cluster that is formed, while the other,  $j$ , is not. So we are dealing with the case that  $x_{ui} \leq \frac{1}{2}$  and  $x_{uj} \geq \frac{1}{2}$ . If in fact,  $x_{uj} \geq \frac{3}{4}$ , then  $x_{ij} \geq \frac{1}{4}$  by the triangle inequality, so the cost paid by our algorithm is within a factor of 4 of the LP-cost for any such edge. Hence, we now focus on the case of nodes  $j$  with  $x_{uj} \in (\frac{1}{2}, \frac{3}{4})$ . We compare the number of '+' edges cut by our algorithm to the total LP cost of all edges ('+' and '-') between  $j$  and nodes in the cluster  $C$ .

By the triangle inequality (captured in Lemma 4 in the previous lecture), writing  $p_j$  and  $n_j$  for the number of '+' resp. '-' edges between  $j$  and nodes from  $C$ , we obtain that

$$\begin{aligned} \text{LP}_j &= \sum_{i \in C, +(ij)} x_{ij} + \sum_{i \in C, -(ij)} (1 - x_{ij}) \\ &\geq \sum_{i \in C, +(ij)} (x_{uj} - x_{ui}) + \sum_{i \in C, -(ij)} (1 - x_{ui} - x_{uj}) \\ &= p_j x_{uj} + n_j (1 - x_{uj}) - \sum_{i \in C} x_{ui} \end{aligned}$$

Because the algorithm didn't form a singleton cluster, the average distance of nodes in  $C$  from  $u$  is at most  $\frac{1}{4}$ , so  $\text{LP}_j$  is bounded below by  $p_j x_{uj} + n_j (1 - x_{uj}) - \frac{p_j + n_j}{4}$ . But  $\frac{3}{4} \geq x_{uj} \geq \frac{1}{2}$ , so

$$\text{LP}_j \geq \frac{p_j}{2} + \frac{n_j}{4} - \frac{p_j + n_j}{4} = \frac{p_j}{4}$$

As the algorithm cuts  $p_j$  '+' edges, the total cost of edges cut by the algorithm is at most four times the LP-cost. By summing this over all nodes  $j$ , and all clusters formed, we obtain that the algorithm is a 4-approximation.

We mentioned above that, while the optimal solution for the minimization and maximization version is the same, the approximation guarantees differ. For the minimization version on complete graphs, the algorithm from [2] we just analyzed gives a 4-approximation. On the other hand, [2] also shows that the problem is APX-hard, i.e., there is some constant such that the minimization version on complete graphs cannot be approximated to within better than that constant unless P=NP. On arbitrary graphs, the best known approximation is  $O(\log n)$ ; however, it is open whether the problem can be approximated to within a constant.

For the maximization version, there is a PTAS on complete graphs [1]. For graphs that are not complete, the problem is APX-hard. However, in this case, it is known how to approximate it to within a constant factor. The factor of 0.7664 from [2] was improved to an 0.7666 approximation via semi-definite programming by Swamy [3].

Even though there is a constant-factor approximation for the minimization version on complete graphs, together with an APX-hardness result, we may wonder what is the best possible constant. We will show that the LP used above has an integrality gap of 2, so no rounding approach solely based on that LP can yield an algorithm with a better guarantee. The example is the “wheel” graph, in which all nodes of a  $k$ -cycle are connected to one additional center node with a ‘+’ edge (while all edges of the cycle are labeled ‘-’). Then, the integral optimal solution puts all of the cycle nodes in different clusters, paying a total of  $k - 1$ , while the fractional optimum assigns  $x_{ij} = \frac{1}{2}$  to all edges between the center and the cycle nodes, paying  $\frac{k}{2}$ . The ratio approaches 2 as  $k \rightarrow \infty$ .

The authors show that no rounding algorithm based on the same type of “region growing” will lead to an approximation guarantee of better than 3, and conjecture that in fact, no similar approach will give a better approximation than the factor of 4 obtained.

## 1 Metric Labeling

When looking at correlation clustering, we started from the motivation that nodes that have ‘+’ edges between them are more likely to belong to the same cluster. In a sense, we could then consider the clusters we formed as communities, sharing perhaps some similarity in topic. If we have an a priori estimate of the topics that nodes (web pages) are about, we can use a similar approach to correct and refine our estimates. For instance, if a page about coffee points to a lot of pages about programming, then perhaps we misinterpreted the meaning of “Java” in one case, and should revise our initial estimate. Hence, we will try to optimize two conflicting goals: agreements with an a priori estimate, and agreements between nodes with edges between them.

Essentially the same problem arises frequently in the context of vision [4]. The goal there is to label pixels into classes (classification problem) and assign labels representing these classes as foreground or background objects (or different colors). Again, the competing constraints are that we don’t want many disagreements with the a priori labels, but also not between adjacent (physically close) pixels.

More formally, we can state the problem as follows. Given a graph  $G$  on a vertex set  $V$  and edge set  $E$ , where each edge  $e = (u, v)$  has weight  $w_e$  representing the cost or strength of the relationship between vertices  $u$  and  $v$ , we want to assign labels  $a \in L$  such that we minimize the assignment cost:

$$\sum_v c(v, f(v)) + \sum_{e=(u,v)} w_e \cdot d(f(u), f(v)).$$

Here,  $f : V \rightarrow L$  is the label mapping that the algorithm has to choose.  $c(v, f(v))$  is the cost of choosing label  $f(v)$  for node  $v$ , which will be a result of deviating from the a priori label.  $d(a, a')$  is a metric representing the distance between two labels  $a, a'$ : it is the cost that is incurred by having labels  $a$  and  $a'$  on adjacent nodes. Here, we will be concerned with the simple Potts model, which does not distinguish *how* different labels are, just whether or not they are different. That is,

$$d(a, a') = \begin{cases} 0 & : a = a' \\ 1 & : a \neq a' \end{cases}$$

This problem is also called the *uniform labeling* problem. It is possible to come up with a 2-approximation algorithm to this problem based purely on local search and graph min-cuts [4]. Here, we will obtain a 2-approximation using an LP-rounding technique, based on the work in [5]. To phrase this problem as an (integer) linear program, we define variables  $x_{v,a}$ , which will be equal to 1 if node  $v$  is assigned label  $a$ , and equal to 0 otherwise.

Using the  $x_{v,a}$  values, we would like to define a variable  $y_e$  that is equal to 1 if the two endpoints of  $e$  have different labels, and equal to 0 otherwise. To do this, we notice that if the endpoints  $u$  and  $v$  of  $e$  have the same label, then  $x_{v,a} - x_{u,a} = 0$  for all  $a$ . Otherwise, it is 0 for all but two of the  $a$ , and equal to 1 and -1 for the other two. Thus, for  $e = (u, v)$ ,

$$y_e = \frac{1}{2} \sum_a |x_{v,a} - x_{u,a}|.$$

However, this is still no linear constraint, as  $|x_{v,a} - x_{u,a}|$  is not a linear function. We will see in a moment how to deal with this problem. For now, notice that our goal is to minimize

$$\sum_{v,a} x_{v,a} c(v, a) + \sum_e w_e \cdot y_e. \tag{1}$$

The only constraint is that each node should obtain exactly one label, so  $\sum_a x_{va} = 1$  for all  $v$ , and each  $x_{v,a} \in \{0, 1\}$ .

To return to the issue of how to express  $y_e$  via linear constraints, we first write  $y_e = \frac{1}{2} \sum_a y_{e,a}$ , where  $y_{e,a} = |x_{v,a} - x_{u,a}|$ . While we cannot directly express this as a linear constraint, we can require that  $y_{e,a} \geq |x_{v,a} - x_{u,a}|$ , by writing  $y_{e,a} \geq x_{u,a} - x_{v,a}$  and  $y_{e,a} \geq x_{v,a} - x_{u,a}$  for all  $e = (u, v)$ . But since the objective is to minimize the objective function (1), the optimum solution will never choose  $y_{e,a}$  larger than necessary, so we will indeed have  $y_{e,a} = |x_{v,a} - x_{u,a}|$ . In summary, we have derived the following IP for the metric labeling problem:

$$\begin{array}{ll} \text{Minimize} & \sum_{v,a} x_{v,a} c(v, a) + \sum_e w_e \cdot y_e \\ \text{Subject to} & \sum_a x_{va} = 1 \quad \text{for all } v, a \\ & y_e = \frac{1}{2} \sum_a y_{e,a} \quad \text{for all } e \\ & y_{e,a} \geq x_{v,a} - x_{u,a} \quad \text{for all } e = (u, v), a \\ & y_{e,a} \geq x_{u,a} - x_{v,a} \quad \text{for all } e = (u, v), a \\ & x_{v,a} \in \{0, 1\} \quad \text{for all } v, a. \end{array}$$

Next lecture, we will relax this IP to an LP by changing the last constraint to  $x_{v,a} \in [0, 1]$ , and then investigate how to round the fractional solution.

## References

- [1] N. Bansal, A. Blum, S. Chawla. Correlation Clustering In *Machine Learning*, 56(1-3):89-113, 2004.
- [2] M. Charikar, V. Guruswami, and A. Wirth. Clustering with Qualitative Information. In *Proc. 44th IEEE FOCS (2003)*.
- [3] C. Swamy. Correlation Clustering: Maximizing Agreements via Semidefinite Programming. In *Proceedings of SODA 2004*.
- [4] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.
- [5] J. Kleinberg, and E. Tardos. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. In *Proc. 40th IEEE Symposium on Foundations of Computer Science*, 1999.