On the Optimal Assignment of Conference Papers to Reviewers

Camillo J. Taylor
University of Pennsylvania, cjtaylor@cis.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-08-30

On the Optimal Assignment of Conference Papers to Reviewers

Camillo J. Taylor
Department of Computer and Information Science
University of Pennsylvania
cjtaylor@cis.upenn.edu

1 Introduction

This monograph will describe an algorithm for optimally assigning conference papers to reviewers. The process was developed and used for CVPR 2006 (the IEEE Conference on Computer Vision and Pattern Recognition) where it successfully matched over a 1100 papers to over 500 reviewers based on preferences indicated by a set of area chairs.

The procedure was designed to address some of the issues associated with the more traditional manual approaches to paper assignment. Previously, area chairs were responsible for individually and collectively solving a fairly subtle constrained optimization problem. Specifically, they were tasked with coming up with an assignment of papers to reviewers such that each paper received a full set of reviews from qualified reviewers without unduly burdening any individual reviewer with too many papers. As the number of papers and reviewers grows, it becomes more difficult to ensure that papers are adequately reviewed and that popular reviewers are used optimally. As an example of the kind of issues that can arise, area chairs based in earlier time zones had the opportunity to assign their papers to popular reviewers early in the process before chairs in later time zones had access to the reviewer pool.

In the proposed system the area chairs are each assigned a set of papers and are responsible for creating soft assignments between each paper and a set of potential reviewers. That is, for each paper the area chair indicates a small set of preferred reviewers and assigns a positive number to each potential review indicating the affinity between the paper and the reviewer; higher values indicate a greater affinity. Once all of the area chairs have assigned their weights the assignment procedure comes up with an 'optimal' assignment taking into account the inputs from all of the area chairs simultaneously.

In this model the area chairs focus on the task of coming up with an appropriate set of candidate reviewers without paying undue attention to the constraints associated with reviewer load. The assignment procedure solves the global constrained optimization problem automatically based on the preferences indicated by the area chairs.

2 Technical Approach

Our assignment problem is clearly a type of bipartite matching problem as depicted in figure 1. In this figure the nodes on the left correspond to papers and the nodes on the right to reviewers. The edges between these two sets of nodes correspond to affinities indicated by the area chairs. Each of these edges is associated with a positive number which represents the strength of the affinity - larger numbers correspond to a greater preference. Let n_p denote the number of papers, n_r denote the number of reviewers and n_e denote the number of edges in the affinity graph. We can encode the entire graph in a sparse affinity matrix $A \in \mathbb{R}^{n_p \times n_r}$ where A_{ij} denotes the affinity between paper i and reviewer j.

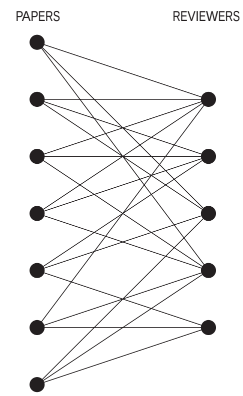

Figure 1. In this figure the nodes on the left correspond to papers and the nodes on the right to reviewers. The edges between the nodes denote soft assignments provided by the conference organizers. The weights associated with each of these edges indicate the strength of the affinity between the paper and the reviewer.

In a standard bipartite matching problem our goal is to find a one-to-one matching between nodes in the graph which optimizes the overall affinity. The difference in this situation is that each paper should ultimately be mapped to multiple reviewers and each reviewer should have be assigned no more than a specified number of papers. Our problem is to deal with this multiplicity of matches.

Our objective then is to arrive at an assignment of papers to reviewers which maximizes the overall affinity subject to the constraints that each paper should receive no more than c_p reviews and each reviewer should be assigned no more than c_r papers. This assignment can be encoded in a second matrix $B \in \mathbb{R}^{n_p \times n_r}$. If paper i is to be assigned to reviewer j the corresponding matrix entry B_{ij} should be 1 otherwise it should be 0. The aforementioned constraints on the assignment can then be expressed as constraints on the row and column sums of this assignment matrix B. Specifically the constraint that each paper should be assigned no more than c_p reviewers corresponds to a requirement that the row sums of the B should all be less than c_p. Similarly the constraint that no reviewer should be assigned more than c_r papers can be expressed by stating that none of the column sums of B should be greater than c_r.

Note a subtle but important feature of this formulation - we are not requiring that every paper be assigned c_p reviewers - indeed depending on the structure of the affinity graph this may be impossible. Our goal is to come up with the assignment that maximizes the overall affinity. Since the affinities are all positive numbers this optimization process will generally prefer to make as many assignments as possible while taking into account the constraints on the assignment matrix and the relative affinities of the edges.

Our optimization problem can be expressed as follows:

$$\begin{array}{ll} \text{maximize} & \text{trace}(A^T B) = \sum_i \sum_j A_{ij} B_{ij} \\ \text{subject to} & B_{ij} \in [0,1] \; \forall ij \\ & \sum_j B_{ij} \leq c_p \; \forall i \\ & \sum_i B_{ij} \leq c_r \; \forall j \end{array} \quad (1)$$

At first blush, it would appear that restricting the entries in B to binary values, 1 and 0, would leave us with an integer programming problem. This would be unfortunate since integer programming problems are notoriously difficult to solve. Happily, this is not the case for the problem at hand. It turns out that we will be able to reformulate our optimization problem as a linear program where the structure of the constraints naturally restrict the solution to integer values without any special effort on our part.

To do this we will make use of the node edge adjacency matrix, $N \in \mathbb{R}^{(n_p+n_r)\times n_e}$. Every row in this matrix corresponds to a node in the affinity graph and every column to an edge. Each column contains exactly two non-zero entries corresponding to the two nodes joined by that edge. That is $N_{ij} = 1$ if edge j impinges upon node i and is 0 otherwise. Since we are concerned with a bipartite graph where the nodes are divided into two disjoint sets corresponding to papers and reviewers, it is convenient to divide the node edge matrix into two components $N_p \in \mathbb{R}^{n_p \times n_e}$ and $N_r \in \mathbb{R}^{n_r \times n_e}$ to reflect this partition. The matrix N can then be expressed as follows.

$$N = \begin{pmatrix} N_p \\ N_r \end{pmatrix} \quad (2)$$

The node edge matrix has the property that if $\mathbf{x} \in \mathbb{R}^{\mathbf{n_e}}$ denotes a vector of edge weights then $N\mathbf{x}$ returns a vector in $\mathbb{R}^{n_p+n_r}$ containing the sum of the weights of the edges incident on each node in the graph. We will let $\mathbf{a} \in \mathbb{R}^{\mathbf{n_e}}$ denote a vector indicating the weights of the edges in the affinity matrix. Similarly we will let $\mathbf{b} \in \mathbb{R}^{\mathbf{n_e}}$ denote the weights of the edges in the final assignment matrix B.

With these definitions in place the optimization problem can be rewritten as follows.

$$\begin{array}{ll} \text{maximize} & \mathbf{a^T b} \\ \text{subject to} & N_p \mathbf{b} \leq \mathbf{c_p} \\ & N_r \mathbf{b} \leq \mathbf{c_r} \\ & \mathbf{b} \leq \mathbf{1} \\ & \mathbf{b} \geq \mathbf{0} \end{array} \quad (3)$$

In this reformulation the matrix inner product $trace(A^T B)$ is replaced by $\mathbf{a^T b}$, the row and column constraints on B are replaced by $N_p \mathbf{b} \leq \mathbf{c_p}$ and $N_r \mathbf{b} \leq \mathbf{c_r}$ respectively and the range constraints on the entries of B are encoded by $\mathbf{b} \geq \mathbf{0}$ and $\mathbf{b} \leq \mathbf{1}$. [1] We can combine all of the linear inequality constraints into a single expression $K\mathbf{b} \leq \mathbf{d}$ as shown below.

$$\begin{array}{ll} \text{maximize} & \mathbf{a^T b} \\ \text{subject to} & K\mathbf{b} \leq \mathbf{d} \\ \text{where} & K = \begin{pmatrix} N_p \\ N_r \\ I \\ -I \end{pmatrix}, \mathbf{d} = \begin{pmatrix} c_p \\ c_r \\ 1 \\ 0 \end{pmatrix} \end{array} \quad (4)$$

At this point we note that the integer matrix K is *totally unimodular*. A square integer matrix, F, is termed *unimodular* if its determinant $\det(F) = \pm 1$. An integer matrix G is termed *totally unimodular* if every square, nonsingular submatrix of G is unimodular [2].

In order to show that the matrix K is totally unimodular we begin by observing that the node edge matrix, N, is totally unimodular by construction. We then recall that if a matrix G is totally unimodular then matrices derived from G by appending I or $-I$ are also unimodular. That is if G is totally unimodular then so are the matrices: $(G\, I)$, $(G\, -I)$ and G^T. Proofs of these properties are quite straightforward and can be found in [2].

Since the constraint matrix K is totally unimodular and the entries in the vector $\mathbf{d}$ are all integral we can conclude that the vertices of the convex polytope defined by the linear equation $K\mathbf{b} \leq \mathbf{d}$ have integral coordinates. [2] Since the optimal values of a linear program correspond to the vertices of the associated convex set we can conclude that at least one of the optimal solutions to our problem will have binary entries. In other words, we can simply formulate and solve the linear program given in Equation 4 and be assured that the resulting assignment matrix B will have binary entries and will satisfy all relevant constraints.

The resulting linear program can be solved quite readily using standard interior point methods [1]. For CVPR 06 assignment problems involving over 1100 papers and over 500 reviewers were routinely solved in a matter of seconds using the linprog routine in MATLAB.

2.1 Assigning Affinities

The procedure described above can be used to find an optimal assignment between papers and reviewers given an affinity matrix $A \in \mathbb{R}^{n_p \times n_r}$. There are, however, quite a number of reasonable approaches to assigning affinities between

[1] Since the edge weights in the affinity matrix are all positive we don't actually need to explicitly enforce the constraint $\mathbf{b} \geq \mathbf{0}$

papers and reviewers. Let us consider the most straightforward approach wherein each assignment that the area chair makes between a paper and a reviewer is given equal weight. That is all of the non-zero entries in the matrix A have the same value. In this case the optimization routine will simply find the solution with the greatest number of assignments. Since all of the assignments have equal weight the assignment procedure is motivated to make as many assignments as possible subject to the load constraints on the reviewers and on the papers. Clearly, the maximum number of assignments that can be made is bounded above by $\min(n_p c_p, n_r c_r)$.

Alternatively, we can use the affinities to suggest that some assignments should be preferred over others. An area chair may indicate that the first choice reviewer assigned to a paper will have an affinity score of 5, the second best reviewer a score of 4 and so on down to 1. In this scenario the assignment procedure gets 5 times the reward for assigning the most preferred reviewer as it would get for assigning the least favored reviewer. However this optimal assignment may contain fewer assignments since the assignment procedure may be biased to assign preferred reviewers to papers at the expense of leaving other papers partially assigned. One can balance the ratio between the largest affinity score and the smallest affinity value to smoothly transition between finding the largest number of assignments vs assigning the most preffered reviewers. Different choices of affinity will result in slightly different assignments.

One can monitor the total number of assignments that are made by the procedure by computing the row sums of the final assignment matrix B. Ideally, each paper will end up with c_p assignments but a few papers may be incompletely assigned if there are not sufficiently many edges in the graph or if the affinity function is skewed in such a way that an incomplete assignment produces a higher total affinity.

Similarly it is useful to monitor the total affinity of the assignment produced by the optimization procedure and compare it to the theoretical maximum score which corresponds to every paper being assigned its n_p favorite reviewers. This number provides a good indication of how effectively the assignment procedure is satisfying the preferences indicated by the area chairs.

2.2 Adding Noise

One situation that can occur if integral values are used for the affinity scores is that multiple assignments may have precisely the same total affinity. Geometrically this corresponds to a situation where the vector defined by the edge weights, $\mathbf{a}$, is perpendicular to a facet of the convex polytope defined by the linear constraints, $K\mathbf{b} \leq \mathbf{d}$. Although the optimal solutions are still integral the barrier method may converge to a point in the center of the polygonal facet yielding non-integral entries in the final assignment vector.

One simple way to overcome this is by slightly perturbing the affinity vector with a small amount of additive noise. That is, we can replace the vector of edge weights, $\mathbf{a}$, with the vector $\mathbf{a} + \rho\delta$ where δ is a vector in $\mathbb{R}^{n_e}$ with random entries between 0 and 1 and ρ is a scalar. This corresponds to adding a small random number to the affinity score associated with each of the vertices of the constraint polytope. That is, if $\mathbf{p} \in \mathbb{R}^{\mathbf{n_e}}$ denotes the coordinates of a vertex of the convex ploytope then its affinity score before the perturbation would be $\mathbf{p^T a}$ and its score after would be $\mathbf{p^T}(\mathbf{a} + \rho\delta) = \mathbf{p^T a} + \rho(\mathbf{p^T}\delta)$

The maximum value of the perturbation added to each vertices score should be less than one so that the noise does not materially affect the relative ordering of the assignments. This implies that $\rho(\mathbf{p^T}\delta) < \mathbf{1}$. Since the maximum value of $\mathbf{p^T}\delta$ is bounded by $c_p n_p$ - the maximum possible number of 1s in the assignment vector $\mathbf{p}$ - we can conclude that the scalar ρ should be bounded by the following inequality $\rho(c_p n_p) < 1$

2.3 Odds and Ends

In this framework one can manually assign a paper to a reviewer by setting the corresponding entry in the assignment matrix B to a constant and removing that variable from the optimization procedure. Similarly one can manually specify that certain papers should *not* be assigned to certain reviewers by setting the corresponding entries in the affinity matrix A to zero. This effectively removes the edge between the paper and the reviewer and ensures that the assignment will not be made.

It is also possible to indicate that different reviewers should have different load levels. In this situation c_r should be interpreted not as a scalar which specifies a single load constraint for all reviewers but rather as a vector which specifies an individual load limit for each reviewer. [2] Similarly, we could interpret c_p as a vector which indicates the number of reviewers that are to be assigned to each paper individually.

This flexibility can be convenient in situations where some of the assignments are being made manually and others automatically. Here the load limit vectors would indicate how many assignments remain to be made for each paper or reviewer after the manual assignments have been applied.

One needs to keep in mind that the assignment procedure may not fully assign each and every paper. That is the assignment may result in some papers having fewer than c_p reviewers. This problem can be solved by adding more edges to the affinity graph. More edges provide the system with greater flexibility in finding assignments and will ultimately allow the system to find a complete assignment. For example, for CVPR 2006 a three of five scheme was employed where area chairs were asked to make five soft assignments so that the assignment procedure would be able to automatically choose three reviewers.

For a given affinity matrix one can experiment with the affinity assignment policy and the load limits of the reviewers in an attempt to arrive at a suitable global assignment. Of course, depending on the number of incomplete assignments one might find it simplest to simply manually assign the remaining papers.

3 Conclusion

This paper describes an approach to solving a variant of the bipartite matching problem which occurs in the context

[2] One might not want to publicize this feature for fear of being innundated with requests for smaller reviewing loads.

of finding an optimal assignment between conference papers and reviewers. The affinities between papers and reviewers are encoded in an affinity graph, or equivalently, a sparse affinity matrix A and the goal of the assignment procedure is to find an assignment between papers and reviewers that maximizes the total affinity subject to constraints on the number of reviewers that can be assigned to a paper and the load that can be assigned to any individual reviewer. The problem is formulated as a linear program which can be readily solved using standard interior point methods.

4 References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Prentice Hall, 1982.