QAP, LAP, and Bayes Plugins

John M. Conroy

May 3, 2011

1 LAP as an Approximation to the QAP

In this note we discuss the linear assignment problem (LAP) method as applied to a class of the quadratic assignment problems (QAPs) that arise in classification of 2 by 2 stochastic un-labeled block model. In particular, we show that the LAP is an optimal solution of a relaxed version of the QAP. More generally, the LAP approach solves an assignment problem that corresponds to a linearization of the quadratic function in for symmetric instances of the QAP.

Recall, the QAP is defined by the following, where we seek to find a permutation matrix $\hat{Q}$ such that

$$\hat{Q}_{\text{QAP}} = \operatorname*{argmin}_{Q \in \mathcal{Q}} \|QAQ^{\mathsf{T}} - B\|_F^2 \tag{1}$$

The classification problem studied is a class of homogeneous "kidney-egg" graphs with n nodes. We are given graphs generated from one of two classes of stochastic block models. Our model problem has the following parameters: $n = 10$, so $\eta^y \in (0,1)^{n \times n}$. We further assume a stochastic block model, with two blocks, for both classes. For class 0, $\eta_{ij}^0 = q_0 = 0.25$ for $i, j \in \{1, 2, 3\}$, and for class 1, $\eta_{ij}^1 = q_1 = 0.75$ for $i, j \in \{1, 2, 3\}$. In both classes, $\eta_{ij} = p = 0.5$ for $i, j \in \{4, \ldots 10\}$. For training data we are given a sample matrix for each class (B_0 and B_1); the parameters for the block model are unknown to classifier. In each trail of the simulation we are given a matrix A and then use an approximate algorithm to find a approximate solutions to the 2 QAPs with $B = B_i, i = 0, 1$.

The LAP approximation for this class problem is given by the following equation:

$$\hat{Q}_{\text{LAP}} = \operatorname*{argmin}_{Q \in \mathcal{Q}} \|AQ^{\mathsf{T}} - B_i\|_F^2, \ i = 0, 1. \tag{2}$$

As the LAP approximation applies just column operations to the A, it has power to distinguish between the two classes due to the fact that both classes have non-uniform column distributions. The entries in the columns of are distributed as a mixture of the binomial distribution for the kidney and the egg in both classes. (This panel of the the matrices we might call a "scambled-egg," as the egg "signal" is mixed into portion of the "kidney" edges.) The ability of the LAP to distinguish between class 0 and 1 relies on the fact that the column distributions of the scambled-egg portions differ from the kidney columns in both classes are non-zero with probability 0.5. In our model problem for class 0, the entries in columns 1,2, and 3 are non-zero with probability $\frac{17}{40} = \frac{3(0.25)+7(0.5)}{10}$ while the remaining columns' entries are non-zero with probability $\frac{1}{2}$. Similarly, for class 1, entries in columns 1,2, and 3 are non-zero with probabiltiy $\frac{23}{40} = \frac{3(0.75)+7(0.5)}{10}$. As long as the stochastic block models for at least one of the classes has a "scambled-egg" subset of columns that differ from the remaining subset of columns, the LAP approximation of this class of QAPs will have power to distinguish between the two classes.

Secondly, the LAP approximation to the QAP solves the same linear assignment algorithm used in the first iteration of the Frank Wolfe algorithm (FW) when started at the identity, when A and B are symmetric. In particular, recall that function and its gradient employed by FW are given by

$$f(P) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(A \circ PBP^t\right)_{ij}$$
$$\nabla_P f(P) = APB^t + A^t PB.$$

The Frank-Wolfe method (FW) is an iterative method to approximately solve a quadratic programming problem. FW takes a starting point $X^{(0)} \in \mathcal{D}$, the set of doubly stochastic matrices, and then proceeds by solving the LAP derived from a first order Taylor series expansion about $X^{(0)}$ as given in Step 1:

$$W^{(k)} = \text{argmin}\, W^{(k)}\{\sum_{i,j=1}^{m} \left(\nabla_X f(X^{(k)}) \circ W^{(k)}\right)_{ij} \mid W^{(k)} \in \mathcal{Q}\}$$

The solution of this LAP defines the direction to search for the next iterate. An exact line search is used to find the next iterate which is also a doubly stochastic matrix. The procedure is iterated until a desired accuracy is obtained or convergence. Our variation of FW outputs the nearest permutation matrix to the double stochastic matrix on the final iteration.

We note that the LAP approximation to the QAP solves equation 2 as follows:

$$\hat{Q}_{LAP} = \underset{Q}{\text{argmin}}\{\sum_{i,j=1}^{m} \left(AB^{\mathsf{T}} \circ Q\right)_{ij} \mid Q \in \mathcal{Q}\}$$

In the case where A and B are symmetric we note that

$$\nabla_X f(I) = 2AB^{\mathsf{T}},$$

so in this case the LAP solves the assignment problem that would be tackled by FW on the first iteration given the starting point of the identity matrix.

We note that in our experiments with FW the starting point in the classification experiments was $\frac{1}{n}J$, the "flat" doubly stochastic matrix.

2 Bayes Plug-In

We illustrate how to formulate the QAP to serve as a Bayes plugin. In particular, if the parameters of the classes $\eta^0_{ij} > 0$ and $\eta^1_{ij} > 0$ are known for both classes, a QAP for each class can be formulated whose solution finds a permutation that maximizes the log probability of the observed edges in the graph, conditional on a class label. Thus, given an binary matrix A representing the adjacency matrix, we consider the two QAP problems for $y = 0$ and $y = 1$.

$$\hat{Q^y}_{QAP} = \underset{Q}{\text{argmin}}\{\sum_{i,j=1}^{m} \left(QH^yQ^{\mathsf{T}} \circ A\right)_{ij} \mid Q \in \mathcal{Q}\}$$

where

$$H^y_{ij} = \log(\eta^y_{ij}).$$

The solutions to these QAPs return the $\log(P(A|y))$ for $y = 0, 1$. Given priors $P(y)$ and Bayes rule we can readily compute the posterior probabliities of the class labels 0 and 1 and thereby form a Bayes plugin classifier.

3 C-elegans Results

To further test the approximate solution of the QAP we turn to the c-elegans data. Three representations of the data expressing the neurons interrationships are available. While in each case, there are 279 neurons in the model their connectivity is represented differently as a weighted graph (sparse matrix). The below table shows some of the characteristics. We note in particular, that the Agordered representation has the fewest edges and unique edge weights.

Graph	Number of Edges	Number of Unique Edge Weights
Agordered	1031	11
Ainittordered	2194	29
Ac	2194	37

100 random permututations were used to generate 300 test problems, 100 for each representation. A test problem consisted of two matrices, the original and a permuted verion of the original. Finding the unknown permuation is an instance of a quadratic assignment problem. In each test problem the Frank-Wolfe method was applied starting

from the “flat” double-stochastic matrix. For Agtordered a maximum of 20 iteration was done and for the remaining formulations a maximum of 5 iterations were done. The number of iterations was chosen so that the average run time was approximately the same. For Agordered the FW method failed to converge in less than 20 iterations in 73 of the 100 problems. In contrast, convergence was achieved in less than 5 iterations for both of the richer models. The below table gives a summary of the results. As seen in each case for the richer models the assignment of the neurons was correctly found by the FW method.

Graph	Mean Time	Mean Iterations	Mean Correct
Agtordered	82.95	19.16	162
Ainittordered	79.90	3.00	279
Ac	79.98	3.57	279