Submitted to Advances in Neural Information Processing Systems (NIPS) 8

A convergence proof for the softassign quadratic assignment algorithm[1]

Anand Rangarajan[2]
Department of Diagnostic Radiology
Yale University
e-mail: anand@noodle.med.yale.edu

Steven Gold
Department of Computer Science
Yale University
e-mail: gold-steven@cs.yale.edu

Eric Mjolsness
Dept. of Computer Science and Engineering
University of California San Diego
e-mail: emj@cs.ucsd.edu

Abstract

The softassign quadratic assignment algorithm has recently emerged as an effective strategy for a variety of optimization problems in pattern recognition and combinatorial optimization. While the effectiveness of the algorithm was demonstrated in thousands of simulations, there was no known proof of convergence. Here, we provide a proof of convergence for the most general form of the algorithm.

1 Introduction

Recently, a new neural optimization algorithm has emerged for solving quadratic assignment like problems [5, 3]. Quadratic assignment problems (QAP) are characterized by quadratic objectives with the variables obeying permutation matrix constraints. Problems that roughly fall into this class are TSP, graph partitioning (GP) and graph matching. The new algorithm is based on the softassign procedure which guarantees the satisfaction of the doubly stochastic matrix constraints (resulting from a "neural" style relaxation of the permutation matrix constraints). While the effectiveness of the softassign procedure has been demonstrated via thousands of simulations, no proof of convergence was ever shown.

Here, we show a proof of convergence for the softassign quadratic assignment algorithm. The proof is based on algebraic transformations of the original objective and on the non-negativity of the Kullback-

[1]Keywords: Algorithms and Architectures, Preference: Oral

[2]Author to whom all correspondence should be addressed. Address: Department of Diagnostic Radiology, Yale University School of Medicine, 333 Cedar Street, New Haven, CT 06520-8042.

Leibler measure. A central requirement of the proof is that the softassign procedure always returns a doubly stochastic matrix. After providing a general criterion for convergence, we separately analyze the cases of TSP and graph matching.

2 Convergence proof

The deterministic annealing quadratic assignment objective function is written as [5, 6]:

$$E_{\mathrm{qap}}(M,\mu,\nu) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai} M_{bj} + \sum_a \mu_a (\sum_i M_{ai} - 1) + \sum_i \nu_i (\sum_a M_{ai} - 1) - \frac{\gamma}{2}\sum_{ai} M_{ai}^2 + \frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai} \quad (1)$$

This form of the energy function has a *self-amplification* term with a parameter γ, two Lagrange parameters μ and ν for constraint satisfaction, an $x \log x$ barrier function which ensures positivity of M_{ai} and a deterministic annealing control parameter β. The QAP benefit matrix $C_{ai;bj}$ is preset based on the chosen problem, for example, graph matching or TSP. The deterministic annealing pseudocode follows:

Initialize β to β_0, M_{ai} to $\frac{1}{N} + \xi_{ai}$

Begin A: Deterministic annealing. Do A until $\beta \geq \beta_f$

  Begin B: Relaxation. Do B until all M_{ai} converge or number of iterations $> I_B$

  $Q_{ai} \leftarrow \sum_{bj} C_{ai;bj} M_{bj} + \gamma M_{ai}$

    Begin Softassign:

    $M_{ai} \leftarrow \exp(\beta Q_{ai})$

      Begin C: Do C until all M_{ai} converge

      Update M_{ai} by normalizing the rows:

      $M_{ai} \leftarrow \frac{M_{ai}}{\sum_i M_{ai}}$

      Update M_{ai} by normalizing the columns:

      $M_{ai} \leftarrow \frac{M_{ai}}{\sum_a M_{ai}}$

      End C

    End Softassign

  End B

$\beta \leftarrow \beta_r \beta$

End A

The softassign is used for constraint satisfaction. The softassign is based on Sinkhorn's theorem [5] but can be independently derived as coordinate ascent on the Lagrange parameters μ and ν. Sinkhorn's theorem ensures that we obtain a doubly stochastic matrix by the simple process of alternating row and column normalizations. The QAP algorithm above was developed using the graduated assignment heuristic [2] with no proof of convergence until now.

We simplify the objective function in (1) by collecting together all terms quadratic in M_{ai}. This is achieved by defining

$$C^{(\gamma)}_{ai;ai} = C_{ai;ai} + \gamma. \quad (2)$$

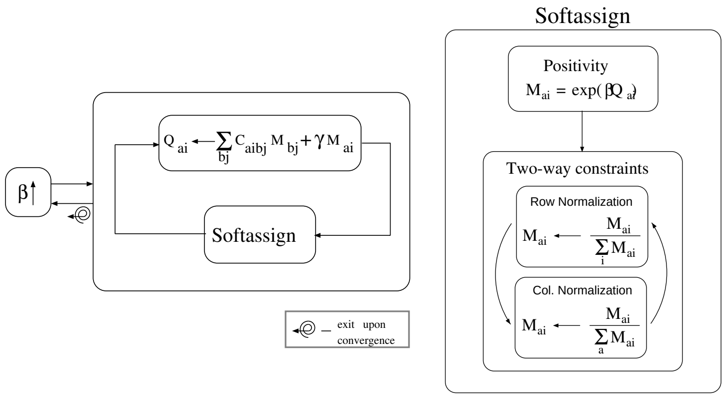

Figure 1: Left: The softassign quadratic assignment algorithm. Right: Softassign

Then we use an *algebraic transformation* [4] to transform the quadratic form into a more manageable linear form:

$$-\frac{X^2}{2} \rightarrow \min_{\sigma}\left(-X\sigma + \frac{1}{2}\sigma^2\right) \tag{3}$$

Application of the algebraic transformation (in a vectorized form) yields:

$$E_{\mathrm{qap}}(M,\sigma,\mu,\nu) = -\sum_{aibj} C^{(\gamma)}_{ai;bj} M_{ai}\sigma_{bj} + \frac{1}{2}\sum_{aibj} C^{(\gamma)}_{ai;bj}\sigma_{ai}\sigma_{bj}$$
$$+\sum_a \mu_a(\sum_i M_{ai} - 1) + \sum_i \nu_i(\sum_a M_{ai} - 1) + \frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai} \tag{4}$$

Minimizing (4) w.r.t. σ, we get

$$\sum_{bj} C^{(\gamma)}_{ai;bj} M_{bj} = \sum_{bj} C^{(\gamma)}_{ai;bj}\sigma_{bj} \Rightarrow \sigma_{ai} = M_{ai} \tag{5}$$

is a minimum, provided certain conditions hold which we specify below.

In the first part of the proof, we show that setting $\sigma_{ai} = M_{ai}$ is guaranteed to decrease the energy function. Restated, we require that

$$\sigma_{ai} = M_{ai} = \arg\min_{\sigma}\left(-\sum_{aibj} C^{(\gamma)}_{ai;bj} M_{ai}\sigma_{bj} + \frac{1}{2}\sum_{aibj} C^{(\gamma)}_{ai;bj}\sigma_{ai}\sigma_{bj}\right) \tag{6}$$

If $C^{(\gamma)}_{ai;bj}$ is positive definite in the subspace spanned by M, then $\sigma_{ai} = M_{ai}$ is a minimum of the energy function $-\sum_{aibj} C^{(\gamma)}_{ai;bj} M_{ai}\sigma_{bj} + \frac{1}{2}\sum_{aibj} C^{(\gamma)}_{ai;bj}\sigma_{ai}\sigma_{bj}$.

At this juncture, we make a crucial assumption that considerably simplifies the proof. Since this assumption is central, we formally state it here: "*M is always constrained to be a doubly stochastic matrix.*" In other words, for our proof of convergence, we require the softassign algorithm to return a doubly stochastic matrix (as Sinkhorn's theorem guarantees that it will) instead of a matrix which is merely close to being doubly stochastic (based on some reasonable metric). We also require the variable σ to be a doubly stochastic matrix.

Since M is always constrained to be a doubly stochastic matrix, $C^{(\gamma)}_{ai;bj}$ needs to be positive definite in the linear subspace of rows and columns of M summing to one. The value of γ should be set high enough such that $C^{(\gamma)}_{ai;bj}$ does not have any negative eigenvalues in the subspace spanned by the row and column constraints.

To derive a more explicit criterion for γ, we first define a matrix r in the following manner:

$$r \stackrel{\text{def}}{=} I_n - \frac{1}{n}ee' \tag{7}$$

where I_n is the $n \times n$ identity matrix, e is the vector of all ones and the "prime" indicates a transpose operation. The matrix r has the property that any vector rs with s arbitrary will sum to zero. We would like to extend such a property to cover matrices whose row and column sums stay fixed. To achieve this, take the Kronecker product of r with itself:

$$R \stackrel{\text{def}}{=} r \otimes r \tag{8}$$

R has the property that it will annihilate all row and column sums [1]. Form a vector m by concatenating all the columns of the matrix M together into a single column [$m = \text{vec}(M)$]. Then the vector Rm has the equivalent property of the "rows" and "columns" summing to zero. Hence the matrix RCR (where C is the matrix equivalent of $C^{(\gamma)}_{ai;bj}$) satisfies the criterion of annihilated row and column sums in any quadratic form; $m'RCRm = (Rm)'C(Rm)$.

The parameter γ is chosen such that all eigenvalues of RCR are non-negative:

$$\gamma = -\min_{\lambda} \lambda(RCR) + \epsilon \tag{9}$$

where $\epsilon > 0$ is a small quantity. We cannot always efficiently compute the largest negative eigenvalue of the matrix RCR. Since the original $C_{ai;bj}$ is four dimensional, the dimensions of RCR are $N^2 \times N^2$ where N is the number of elements in one set. Fortunately, as we show later, for specific problems it's possible to break up RCR into its constituents thereby making the calculation of the largest negative eigenvalue of RCR more efficient. We return to this point in Section 3.

The second part of the proof involves demonstrating that the softassign operation also decreases the objective in (4). (Note that the two Lagrange parameters μ and ν are specified by the softassign algorithm [5]).

$$M = \text{Softassign}(Q, \beta) \text{ where } Q_{ai} = \sum_{bj} C^{(\gamma)}_{ai;bj}\sigma_{bj} \tag{10}$$

Recall that the step immediately preceding the softassign operation sets $\sigma_{ai} = M_{ai}$. We are therefore justified in referring to σ_{ai} as the "old" value of M_{ai}. For convergence, we have to show that $E_{\text{qap}}(\sigma, \sigma) \geq E_{\text{qap}}(M, \sigma)$ in (4).

Minimizing (4) w.r.t. M_{ai}, we get

$$\frac{1}{\beta}\log M_{ai} = \sum_{bj} C^{(\gamma)}_{ai;bj}\sigma_{bj} - (\mu_a + \nu_i) - \frac{1}{\beta} \tag{11}$$

From (11), we see that

$$\frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai} = \sum_{aibj} C^{(\gamma)}_{ai;bj} M_{ai}\sigma_{bj} - \sum_a \mu_a \sum_i M_{ai} - \sum_i \nu_i \sum_a M_{ai} - \frac{1}{\beta}\sum_{ai} M_{ai} \tag{12}$$

and

$$\frac{1}{\beta}\sum_{ai}\sigma_{ai}\log M_{ai} = \sum_{aibj} C^{(\gamma)}_{ai;bj}\sigma_{ai}\sigma_{bj} - \sum_a \mu_a \sum_i \sigma_{ai} - \sum_i \nu_i \sum_a \sigma_{ai} - \frac{1}{\beta}\sum_{ai}\sigma_{ai} \tag{13}$$

From (12) and (13), we get (after some algebraic manipulations)

$$E_{\rm qap}(\sigma,\sigma) - E_{\rm qap}(M,\sigma) =$$

$$-\sum_{aibj} C^{(\gamma)}_{ai;bj}\sigma_{ai}\sigma_{bj} - \left(-\sum_{aibj} C^{(\gamma)}_{ai;bj} M_{ai}\sigma_{bj}\right) + \frac{1}{\beta}\sum_{ai}\sigma_{ai}\log\sigma_{ai} - \frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai}$$

$$= \frac{1}{\beta}\left(\sum_{ai}\sigma_{ai}\log\frac{\sigma_{ai}}{M_{ai}} - \sum_{ai}\sigma_{ai} + \sum_{ai} M_{ai}\right) \geq 0 \tag{14}$$

by the non-negativity of the (slightly generalized) Kullback-Leibler measure. We have shown that the change in energy *after σ has been initialized with the "old" value of M* is non-negative. We require that σ and M are always doubly stochastic via the action of the softassign operation. Consequently, the terms involving the Lagrange parameters μ and ν can be eliminated from the energy function (4). Setting $\sigma = M$ followed by the softassign operation decreases the objective in (4) after excluding the terms involving the Lagrange parameters.

We summarize the essence of the proof to bring out the salient points. At each temperature, the quadratic assignment algorithm executes the following steps until convergence is established.

> **Step 1:** $\sigma_{ai} \leftarrow M_{ai}$.
>
> **Step 2:**
>
> > **Step 2a:** $Q_{ai} \leftarrow \sum_{bj} C^{(\gamma)}_{ai;bj}\sigma_{bj}$.
> > **Step 2b:** $M \leftarrow \mathrm{Softassign}(Q,\beta)$.
>
> Return to Step 1 until convergence.

Our proof is based on demonstrating that an appropriately designed energy function decreases in both Step 1 and Step 2 (at fixed temperature). This energy function is Equation (4) after excluding the Lagrange parameter terms.

> **Step 1:** Energy decreases due to the positive definiteness of $C^{(\gamma)}_{ai;bj}$ in the *linear* subspace spanned by the row and column constraints. γ has to be set high enough for this statement to be true.
>
> **Step 2:** Energy decreases due to the non-negativity of the Kullback-Leibler measure and due to the restriction that M (and σ) are doubly stochastic.

3 Applications

3.1 Quadratic Assignment

The QAP benefit matrix is chosen such that the softassign algorithm will not converge without adding the γ term in (1). To achieve this, we randomly picked a unit vector v of dimension N^2. The benefit matrix C is set to $-vv'$. Since C has only one negative eigenvalue, the softassign algorithm cannot possibly converge. We ran the softassign algorithm with $\beta_0 = 1$, $\beta_r = 0.9$ and $\gamma = 0$. The energy difference plot on the left in Figure 2 shows the energy never decreasing with increasing iteration number. Next, we followed the recipe for setting γ exactly as in Section 2. After projecting C into the subspace of the row and column constraints, we calculated the largest negative eigenvalue of the matrix RCR which turned out to be -0.8152. We set γ to 0.8162 ($\epsilon = 0.001$) and reran the softassign algorithm. The energy difference plot shows (Figure 2) that the energy never increases. We have shown that a proper choice of γ leads to a convergent algorithm.

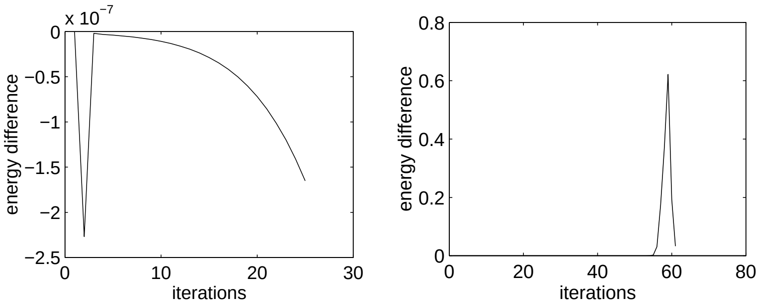

Figure 2: **Energy difference plot.** Left: $\gamma = 0$ and Right: $\gamma = 0.8162$. While the change in energy is always negative when $\gamma = 0$, it is always non-negative when $\gamma = 0.8162$. The negative energy difference (on the left) implies that the energy function increases whereas the non-negative energy difference (on the right) implies that the energy function never increases.

3.2 TSP

The TSP objective function is written as follows: Given N cities,

$$E_{\text{tsp}}(M) = \sum_{aij} d_{ij} M_{ai} M_{(a\oplus 1)j} = \text{trace}(DM'TM) \tag{15}$$

where the symbol $\oplus$ is used to indicate that the summation in (15) is taken modulo N, d_{ij} (D) is the inter-city distance matrix and M is the desired permutation matrix. T is a matrix whose $(i,j)^{\text{th}}$ entry is $\delta_{(i\oplus 1)j}$ (δ_{ij} is the Kronecker delta function). Equation (15) is transformed into the $m'Cm$ form:

$$E_{\text{tsp}}(m) = \text{trace}(m'(D \otimes T)m) \tag{16}$$

where $m = \text{vec}(M)$. We identify our general matrix C with $2D \otimes T$.

For convergence, we require the largest negative eigenvalue of

$$RCR = 2(r \otimes r)(D \otimes T)(r \otimes r) = 2(rDr) \otimes (rTr) = 2(rDr) \otimes (rT) \tag{17}$$

The eigenvalues of rT are bounded by unity. The eigenvalues of rDr will depend on the form of D. Even in Euclidean TSP the values will depend on whether the Euclidean distance or the distance squared between the cities is used.

3.3 Graph Matching

The graph matching objective function is written as follows: Given M and N node graphs with adjacency matrices G and g respectively,

$$E_{\text{gm}}(M) = -\frac{1}{2} \sum_{aibj} C_{ai;bj} M_{ai} M_{bj} \tag{18}$$

where $C_{ai;bj} = 1 - 3|G_{ab} - g_{ij}|$ is the compatibility matrix [2]. The matching constraints are somewhat different from TSP due to the presence of slack variables [2]. This makes no difference however to our projection operators. We add an extra row and column of zeros to g and G in order to handle the slack variable case. Now G is $(M+1) \times (M+1)$ and g is $(N+1) \times (N+1)$. Equation (18) can be readily transformed into the $m'Cm$ form. Our projection apparatus remains unchanged. For convergence, we require the largest negative eigenvalue of RCR.

Subgraph isomorphism is a special case of weighted graph matching.

$$E_{\text{sgi}}(M) = -\frac{1}{2}\sum_{aibj} G_{ab} g_{ij} M_{ai} M_{bj} = -\frac{1}{2}\text{trace}(M'GMg) \tag{19}$$

This can be converted into our standard form via Kronecker products.

$$E_{\text{sgi}} = -\frac{1}{2} m'(g \otimes G) m \tag{20}$$

From this C is identified as $g \otimes G$.

$$RCR = (r \otimes r)(g \otimes G)(r \otimes r) = (rgr) \otimes (rGr) \tag{21}$$

γ is calculated from the eigenvalues of rgr and rGr.

4 Conclusion

We have derived a convergence proof for the softassign quadratic assignment algorithm and specialized to the cases of TSP and graph matching. An extension to graph partitioning follows along the same lines as graph matching. Central to our proof is the requirement that the QAP matrix M is always doubly stochastic. As a by-product, the convergence proof yields a criterion by which the free self-amplification parameter γ is set. We believe that the combination of good theoretical properties and experimental success of the softasign algorithm make it the technique of choice for quadratic assignment neural optimization.

References

[1] A. H. Gee, S. V. B. Aiyer, and R. W. Prager. An analytical framework for optimizing neural networks. *Neural Networks*, 6(1):79–97, 1993.

[2] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:377–388, 1996.

[3] S. Gold and A. Rangarajan. Softassign versus softmax: Benchmarks in combinatorial optimization. In *NIPS 8*. MIT Press, 1996.

[4] E. Mjolsness and C. Garrett. Algebraic transformations of objective functions. *Neural Networks*, 3:651–669, 1990.

[5] A. Rangarajan, S. Gold, and E. Mjolsness. A novel optimizing network architecture with applications. *Neural Computation*, 8(5):1041–1060, 1996.

[6] A. L. Yuille and J. J. Kosowsky. Statistical physics algorithms that converge. *Neural Computation*, 6(3):341–356, May 1994.