Approximate Graph Matching With Convex Optimization

Joseph Lubars
Graph Matching

Social Network Data

Anonymous Communication Data

Alice
Bob
Carol

Alice
Bob
Carol
Very Complicated For Larger Graphs
Theoretical Setting

• Objective: \[ \text{argmin}_{P \in \Pi} \| A - PBP^T \|_F^2 \]

• \( A, B \): The adjacency matrices of our (order \( n \)) graphs
• \( \Pi \): The set of \( n \times n \) permutation matrices
• Interpretation: Minimize the symmetric difference in edges
Equivalent Problems

• Original Objective: \[ \arg\min_{P \in \Pi} \|A - PB^T P^T \|^2_F \]

• Different Norm: \[ \arg\min_{P \in \Pi} \|A - PB^T P^T \|_1 \]

• Multiply by \( P \): \[ \arg\min_{P \in \Pi} \|AP - PB\|^2_F \]

• Trace Formulation: \[ \arg\max_{P \in \Pi} Tr[PB^T P^T A] \]

• Proof: \[ \|A - PB^T P^T \|^2_F = Tr[(A - PB^T P^T)^T (A - PB^T P^T)] \]
  \[= Tr[A^T A - A^T PB^T P^T - PB^T P^T A + PB^T B^T P^T] \]
Our Convex Relaxation

\[
\arg\min_{P \in D} \| AP - PB \|^2_F
\]

• $D$ is the set of doubly stochastic matrices
• This is a convex linearly constrained quadratic program (LCQP)
Why is it Convex?

• $\|AP - PB\|_F^2 = Vec(AP - PB)^T Vec(AP - PB)$

Use the identity $(B^T \otimes A)Vec(X) = Vec(AXB)$:
• $Vec(AP) = (I \otimes A)Vec(P)$
• $Vec(PB) = (B^T \otimes I)Vec(P)$

$$\|AP - PB\|_F^2 = Vec(P)^T Q Vec(P)$$

$Q = (I \otimes A - B^T \otimes I)^T (I \otimes A - B^T \otimes I)$
Let’s Solve It!

- Plug it into CVXPY, get $P_D = \arg\min_{P \in D} \|AP - PB\|_F^2$.
Getting a Permutation

• Need to project back to the set of permutation matrices:

$P^* = \arg\min_{P \in \Pi} \|P - P_D\|_F^2 = \arg\max_{P \in \Pi} \text{Tr}[P^T P_D]$

• This is called the “linear assignment problem”

• Solvable in $n^3$ time using the Hungarian Algorithm
Recap

1. Start with objective: \( \arg\min_{P \in \Pi} \|A - PB P^T \|_F^2 \)

2. Convex relaxation: \( P_D = \arg\min_{P \in D} \|AP - PB \|_F^2 \)

3. The above is a quadratic program; solve it with a standard solver

4. Project back to permutations: \( P^* = \arg\min_{P \in \Pi} \|P - P_D \|_F^2 \)
Biggest Problems

1. Solving the QP $\min_{P \in D} \text{Vec}(P)^T Q \text{Vec}(P)$ is slow
   - $P$ has $n^2$ variables, the matrix $Q$ is $n^2 \times n^2$
   - We will use the Frank-Wolfe algorithm instead

2. The solution to the relaxed problem may be too far away from the optimal solution to the original problem
   - If $A$ and $B$ are isomorphic, then $\min_{P \in D} ||AP - PB||_F^2 = \min_{P \in \Pi} ||AP - PB||_F^2$
   - Otherwise, I have no guarantees.
   - I’ll try to do some discrete optimization to correct errors
Frank-Wolfe (Conditional Gradient Method)

Our objective is: \( F(P) = \text{Vec}(P)^T Q \text{Vec}(P) \)

We can calculate the gradient:

\[
\nabla F(P) = 2Q \text{Vec}(P) = 2\text{Vec}(A^2 P - A^T P B^T - A P B + P B^2)
\]

This can be calculated efficiently, and we proceed as follows:

1. Initialize \( P_0 \) randomly
2. Solve \( \widehat{P}_k = \arg\min_{S \in D} \text{Vec}(S)^T \nabla F(P) \)
3. Set \( \gamma = \frac{2}{t+2} \)
4. Update \( P_{k+1} = (1 - \gamma) P_k + \gamma \widehat{P}_k \)
Correcting the Permutation

• Once we have our permutation: \( P^* = \text{argmin}_{P \in \Pi} \| P - P_D \|_F^2 \)

• We would like to iteratively improve upon our solution:

For \( k = 1 \ldots \text{max\_iterations} \):

For \( i = 1 \ldots n \):

Find \( i' = \text{argmin}_{j \in [n]} F((i, j) \ast P) \)

\[ P := (i, i') \ast P \] (swap \( i \) and \( i' \) in \( P \))
Algorithm Performance
Not Covered/Further Work

• Can the projection \( \min_{P \in D} \|P - P_0\|_F^2 \) be done efficiently (for PGD)?
• Are there any guarantees on the non-convex ‘swap’ algorithm?
• What about other convex relaxations (e.g. orthogonal matrices)?
Questions?