1 QAP, the problem

QAP, and alternative descriptions, see 1

$$
\begin{aligned}
&\min_X f(X) = \mathrm{tr}(A^\top X B X^\top) \\
&= \mathrm{tr}(X^\top A^\top X B) && x = \mathrm{vec}(X) \\
&= \langle \mathrm{vec}(X), \mathrm{vec}(A^\top X B)\rangle \\
&= \langle \mathrm{vec}(X), B^\top \otimes A^\top \cdot \mathrm{vec}(X)\rangle \\
&= x^\top (B^\top \otimes A^\top) x \\
\textbf{s.t.}& \\
&X \in \Pi_n
\end{aligned}
$$

is the optimization problem on permutation matrices:

$$\Pi_n = \left\{X \in \mathbb{R}^{n\times n} \mid Xe = X^\top e = e, X_{ij} \in \{0,1\}\right\}$$

The convex hull of permutation matrices, the Birkhoff polytope, is defined:

$$D_n = \left\{X \in \mathbb{R}^{n\times n} \mid Xe = X^\top e = e, X \geq 0\right\}$$

for the constraints, also equivalently:

$$\mathrm{tr}(XX^\top) = \langle x, x\rangle_F = n, X \in D_n$$

1.1 Differentiation

$$
\begin{aligned}
&\nabla f = A^\top X B + A X B^\top \\
&\nabla \mathrm{tr}(XX^\top) = 2X
\end{aligned}
$$

2 $\mathcal{L}_p$ regularization

various form of regularized problem:

- $\mathcal{L}_0$: $f(X) + \sigma ||X||_0$ is exact to the original problem for efficiently large σ 1, but the problem itself is still NP-hard.
- $\mathcal{L}_p$: also suggested by 1, good in the sense:
  - strongly concave and the global optimizer must be at vertices

  - **local optimizer is a permutation matrix** if σ, ϵ satisfies some condition. Also, there is a lower bound for nonzero entries of the KKT points

$$\min_{X \in D_n} F_{\sigma,p,\epsilon}(X) := f(X) + \sigma \|X + \epsilon 1\|_p^p$$

- $\mathcal{L}_2$, and is based on the fact that $\Pi_n = D_n \bigcap \{X : \mathrm{tr}(XX^\top) = n\}$, 2

$$\min_X f(X) + \mu_0 \cdot \mathrm{tr}\left(XX^\top\right)$$

2.1 $\mathcal{L}_2$

2.1.1 naive

$$\begin{aligned}&\min_X \mathrm{tr}(A^\top XBX^\top) + \mu_0 \cdot \mathrm{tr}(XX^\top)\\=&x^\top(B^\top \otimes A^\top + \mu \cdot \mathbf{e}_{n\times n})x\end{aligned}$$

this implies a LD-like method. (but not exactly)

2.1.2 a better naive

$$\min_X \mathbf{tr}(M^\top SM)$$

$$M = \begin{pmatrix} XB \\ AX \end{pmatrix}, \ S = \begin{pmatrix} \mathbf{0} & \frac{1}{2}\mathbf{I} \\ \frac{1}{2}\mathbf{I} & \mathbf{0} \end{pmatrix}$$

factorizing matrix S by scale factor δ

$$R^\top R = S + \delta I$$

2.2 $\mathcal{L}_2 + \mathcal{L}_1$ penalized formulation

Motivated by the formulation using trace:

$$\begin{aligned}&\min_X \mathrm{tr}(A^\top XBX^\top)\\ \mathbf{s.t.}&\\ &\mathrm{tr}(XX^\top) - n = 0\\ &X \in D_n\end{aligned}$$

using absolute value of $\mathcal{L}_2$ penalty and by the factor that $\forall X \in D_n,\ \mathrm{tr}(XX^\top) \leq n$, we have:

$$
\begin{aligned}
F_\mu &= f + \mu \cdot |\mathrm{tr}(XX^\top) - n| \\
&= \mathrm{tr}(A^\top XBX^\top) + \mu \cdot n - \mu \cdot \mathrm{tr}(XX^\top)
\end{aligned}
$$

For sufficiently large penalty parameter μ, the problem solves the original problem.

The penalty method is very likely to become a concave function (even if the original one is convex), and thus it cannot be directly solved by conic solver.

2.2.1 Projected gradient

```
# code see
qap.models.qap_model_l2.l2_exact_penalty_gradient_proj
```

Suppose we do projection on the penalized problem F_μ

derivatives

$$
\begin{aligned}
\nabla_X F_\mu &= A^\top XB + AXB^\top - 2\mu X \\
\nabla_\mu F_\mu &= n - \mathrm{tr}(XX^\top) \\
\nabla_\Lambda F_\mu &= -X
\end{aligned}
$$

projected derivative problem PD, a quadratic program

$$
\begin{aligned}
&\min_D ||\nabla F_\mu + D||_F^2 \\
&\textbf{s.t.} \\
&\quad De = D^\top e = 0 \\
&\quad D_{ij} \geq 0 \quad \textsf{if:}\ X_{ij} = 0
\end{aligned}
$$

or equivalently, a linear program

$$
\begin{aligned}
&\min_D \nabla F_\mu \bullet D \\
&\textbf{s.t.} \\
&\quad De = D^\top e = 0 \\
&\quad D_{ij} \geq 0 \quad \textsf{if:}\ X_{ij} = 0 \\
&\quad ||D|| \leq 1
\end{aligned}
$$

- There is no degeneracy, great.

2.2.2 Remark

integrality of the solution Computational results show that the *residue of the trace*: $|n - \mathrm{tr}(XX^\top)|$ is almost zero, this means the algorithm converges to an integral solution. (even without any tuning of penalty parameter μ)

Prove that it is exact if μ is sufficiently large, the model converges to an integral solution.

PF. outline a. the model uses exact penalty function, as μ, actually $\{\mu_k\}$ become sufficiently large, the penalty method solves the original problem. **b**. if $\{\mu_k\}$ become sufficiently large, the penalized objective will be concave, so that the optimal solution should be attained at the vertices.

quality of the solution it is however hard to find a global optimum, and gradient projection as defined above converges to a local integral solution and then stops, see instances with gap $> 10\%$.

analytic representation for projection in projected gradient method, let the space of D, (e is the vector of 1s)

$$\mathcal{D} = \{D \in \mathbb{R}^{n\times n} :\ De = D^\top e = 0;\ D_{ij} = 0,\ \forall (i,j) \in M\}$$

is there a way to formulate the set for F such that $\langle F, D\rangle_F = 0,\ \forall D \in \mathcal{D}$, can we find an analytic representation?

***what if projection is zero?** dual problem for PD

- α, β, Λ are Lagrange multipliers, $\mathbf{I}$ is the identity matrix for active constraints of the $X \geq 0$ where $\mathbf{I}_{ij} = 1$ if $X_{ij} = 0$

$$\begin{aligned}
& L_d = 1/2 \cdot ||\nabla F_\mu + D||_F^2 - \alpha^\top De - \beta^\top D^\top e - \Lambda \circ \mathbf{I} \bullet D \\
\text{KKT}: & \\
& \nabla F + D - ae^\top - e\beta^\top - \Lambda \circ \mathbf{I} = 0 \\
& \Lambda \geq 0
\end{aligned}$$

Suppose at iteration k projected gradient $D_k = 0$, then the KKT condition for

We relax one condition for active inequality for some $e = (i,j), e \in M$ such that $X_e = 0$, a new optimal direction for problem PD is achieved at $\hat{D}$, we have:

$$\hat{D}_{ij} - (\alpha_i + \beta_j) + (\hat{\alpha}_i + \hat{\beta}_j) - \Lambda_{ij} = 0, \quad e = (i,j)$$

3 Computational Results

The experiments are done on dataset of QAPLIB, also see paper [3]

The $\mathcal{L}_2$ penalized formulation with code `qap.models.qap_model_l2.l2_exact_penalty_gradient_proj`, solved by gradient projection of module `qap.models.qap_gradient_proj` can solve almost all instances, except for very large ones ($\geq$ 256), it should be better since it now uses Mosek as backend to solve orthogonal projections, line search for step-size, and so on.

current benchmark

Table 1: L_2 + L_1 penalized gradient projection

	value	rel_gap	trace_res	runtime
count	115.000	114.000	115.000	115.000
mean	58087685.641	0.300	2.068	12.905
std	201707868.192	0.544	20.853	84.748
min	0.000	0.000	0.000	0.039
25%	3962.001	0.024	0.000	0.125
50%	97330.031	0.091	0.000	0.215
75%	3207963.510	0.255	0.000	1.584
max	1289655958.000	4.000	223.278	885.492

Reference

[1] B. Jiang, Y.-F. Liu, and Z. Wen, “L_p-norm regularization algorithms for optimization over permutation matrices,” *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2284–2313, 2016.

[2] Y. Xia, “An efficient continuation method for quadratic assignment problems,” *Computers & Operations Research*, vol. 37, no. 6, pp. 1027–1032, 2010.

[3] R. E. Burkard, S. E. Karisch, and F. Rendl, “QAPLIB–a quadratic assignment problem library,” *Journal of Global optimization*, vol. 10, no. 4, pp. 391–403, 1997.