Large-Scale QCQP via Low-Discrepancy Sequences

Kinjal Basu, Ankan Saha, Shaunak Chatterjee

BayLearn 2017
Motivation
Similar Ranking Problems

Examples:

(a) People You May Know (PYMK)

(b) Notifications
Multi-Objective Optimization

- Most ranking / recommendation problems try to balance conflicting metrics.
- Increase in one causes decrease in another.
  - **Feed**: Increase engagement but not drop revenue
  - **Notifications / Emails**: Decrease sends but do not drop sessions.
Optimization Formulation - Notification Example

- Minimize sends such that expected sessions does not drop.
- $x_{ij}$ - Probability of sending item $j$ to user $i$
- $q_{ij}$ - Prior probability of sending item $j$ to user $i$
- $s_{ij}$ - Probability that the user will visit given item $j$ is sent to user $i$

\[
\text{Minimize } \sum_{i=1}^{I} \sum_{j=1}^{J} x_{i,j} + \frac{\gamma}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} (x_{i,j} - q_{i,j})^2 \\
\text{subject to } \sum_{i=1}^{I} \sum_{j=1}^{J} x_{i,j} s_{i,j} \geq C \\
0 \leq x_{i,j} \leq 1
\]
Quadratically Constrained Quadratic Problem (QCQP)

• The probability $s_{ij}$ that user will visit depends not only on the current notification but also on many previous notifications sent.
  • For example total sends till the last visit.
  • No. of flashy UI pushes, etc

• If approximate it as a linear function, say $s = Px$

\[ s_{ij} = p^1_{ij}x_{i1} + p^2_{ij}x_{i2} + \ldots + p^J_{ij}x_{iJ} \]
Quadratically Constrained Quadratic Problem (QCQP)

• Original Problem:

\[
\begin{align*}
\text{Minimize} \quad & \sum_{i=1}^{I} \sum_{j=1}^{J} x_{i,j} + \frac{\gamma}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} \left( x_{i,j} - q_{i,j} \right)^2 \\
\text{subject to} \quad & \sum_{i=1}^{I} \sum_{j=1}^{J} x_{i,j} s_{i,j} \geq C \\
& 0 \leq x_{i,j} \leq 1
\end{align*}
\]

• Derived Problem:

\[
\begin{align*}
\text{Minimize} \quad & x^T 1 + \frac{\gamma}{2} \|x - q\|^2 \\
\text{subject to} \quad & x^T P x \geq C \\
& 0 \leq x \leq 1
\end{align*}
\]
Overview

• Challenges

• Method of solving the Large-Scale QCQP
  • Approximation
  • Sampling techniques

• Theoretical Guarantees

• Empirical Validation
Challenges of Solving a QCQP

• It is NP hard in the general framework.

• Usual Solutions
  • Semidefinite Programming
  • Relaxation Linearization Technique

• Both convert the problem from $n$ variables to $O(n^2)$ and hence becomes prohibitively expensive for large $n$. 
Methodology

• **Main Idea**: QCQP to QP Approximation. Solving the QP by state-of-the-art methods.

• Original Problem:

\[
\text{Minimize } \quad (x - a)^T A (x - a) \\
\text{subject to } \quad (x - b)^T B (x - b) \leq \tilde{b}, \\
C x = c.
\]

Constraint Set:
QCQP to QP Approximations

• Derived problem becomes:

\[
\begin{align*}
& \text{Minimize} & \quad (x - a)^T A (x - a) \\
& \text{subject to} & \quad (x - b)^T B (x_j - b) \leq \tilde{b} \quad \text{for } j = 1, \ldots, N \\
& & \quad Cx = c.
\end{align*}
\]

• Given a fixed N, the accuracy of the solution solely depends on the choice of the points \( x_1, \ldots, x_N \).
Why not Random Points?

• If you are lucky it can be good, but if not it can be arbitrarily bad.

• Can even get an unbounded region
Optimal Choice of Points

• Low-Discrepancy Points
Theoretical Results

Theorem 1: \( \lim_{N \to \infty} \| x^*(N) - x^* \| = 0 \)

Theorem 2: If \( x^*(N) \) converges to \( x^* \) in the rate \( O(g(N)) \), then

\[
|f(x^*(N)) - f(x^*)| \leq C_2 g(N)
\]

and the function \( g(N) \) can be explicitly identified for different constraint domains.
## Comparative Study

**Table 1: The optimal objective value and convergence time**

<table>
<thead>
<tr>
<th>$n$</th>
<th>Our method</th>
<th>Sampling on $[0, 1]^n$</th>
<th>Sampling on $S^n$</th>
<th>SDP relaxation</th>
<th>RLT relaxation</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3.00 (4.61s)</td>
<td>2.99 (4.74s)</td>
<td>2.95 (6.11s)</td>
<td>3.07 (0.52s)</td>
<td>3.08 (0.51s)</td>
<td>3.07 (0.49)</td>
</tr>
<tr>
<td>10</td>
<td>206.85 (5.04s)</td>
<td>205.21 (5.65s)</td>
<td>206.5 (5.26s)</td>
<td>252.88 (0.53s)</td>
<td>252.88 (0.51s)</td>
<td>252.88 (0.51)</td>
</tr>
<tr>
<td>20</td>
<td>6291.4 (6.56s)</td>
<td>4507.8 (6.28s)</td>
<td>5052.2 (6.69s)</td>
<td>6841.6 (2.05s)</td>
<td>6841.6 (1.86s)</td>
<td>6841.6 (0.54)</td>
</tr>
<tr>
<td>50</td>
<td>99668 (15.55s)</td>
<td>15122 (18.98s)</td>
<td>26239 (17.32s)</td>
<td>1.11 x 10^5 (4.31s)</td>
<td>1.08 x 10^5 (2.96s)</td>
<td>1.11 x 10^5 (0.64)</td>
</tr>
<tr>
<td>100</td>
<td>1.40 x 10^6 (58.41s)</td>
<td>69746 (1.03m)</td>
<td>1.24 x 10^6 (54.69s)</td>
<td>1.62 x 10^6 (30.41s)</td>
<td>1.52 x 10^6 (15.36s)</td>
<td>1.62 x 10^6 (2.30s)</td>
</tr>
<tr>
<td>1000</td>
<td>2.24 x 10^7 (14.87m)</td>
<td>8.34 x 10^6 (15.63m)</td>
<td>9.02 x 10^6 (15.32m)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>10^5</td>
<td>3.10 x 10^8 (25.82m)</td>
<td>7.12 x 10^7 (24.59m)</td>
<td>8.39 x 10^7 (27.23m)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>10^6</td>
<td>3.91 x 10^9 (38.30m)</td>
<td>2.69 x 10^8 (39.15m)</td>
<td>7.53 x 10^8 (37.21m)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Comparative Study

![Graph showing comparative study results with log (Relative Square Error) on the y-axis and Dimension of the Problem on the x-axis. The graph compares methods like Low Discrepancy Sampling, Uniform Sampling on Square, and Uniform Sampling on Sphere.]
Future Work

• Comparison with
  • Commercial solvers such as CPLEX.
  • Large-Scale SDP solvers based on ADMM.

• Finding explicit bounds for common domains.

• Finding accurate rates of comparison by considering the growth of the eigenvalues of the matrices.
Summary & Takeaways

• This method gives a highly scalable solution to the QCQP problem, with a theoretical guarantee of convergence.

• Rather than using random points it is better to use low-discrepancy points since random points can lead to arbitrarily bad results.

• Can be used in several applications which were blocked because of the scale of the problem.
Thank you for your attention!