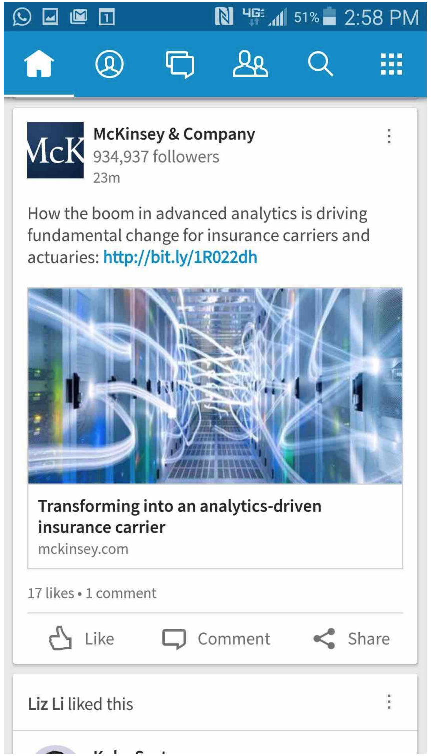
Large-Scale QCQP via Low-Discrepancy Sequences

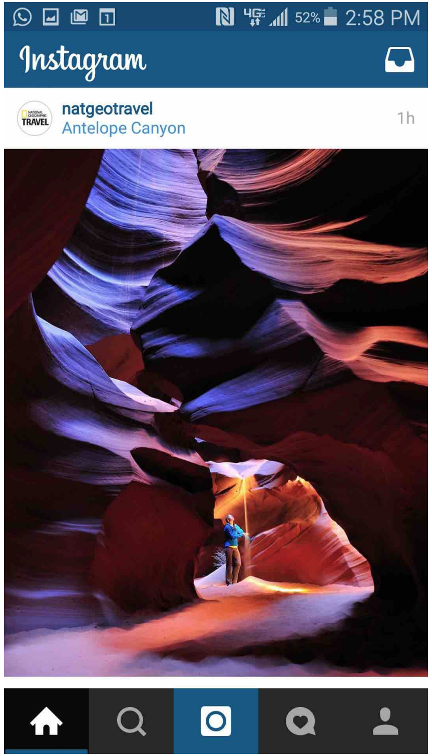
Kinjal Basu, Ankan Saha, Shaunak Chatterjee

BayLearn 2017

Motivation



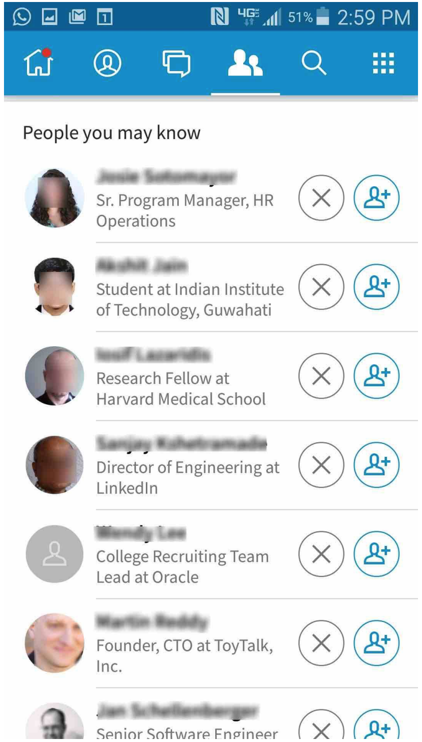
2:58 PM
McKinsey & Company
934,937 followers
23m
How the boom in advanced analytics is driving fundamental change for insurance carriers and actuaries: http://bit.ly/1R022dh
Transforming into an analytics-driven insurance carrier
mckinsey.com
17 likes • 1 comment
Like
Comment
Share
Liz Li liked this

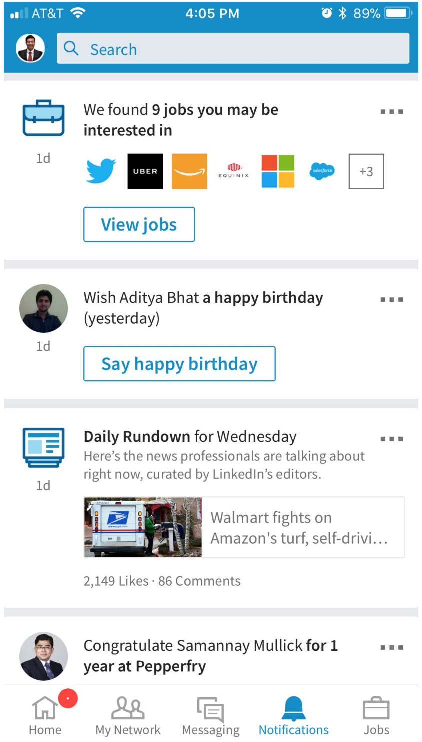


2:58 PM
Instagram
natgeotravel
Antelope Canyon
1h

2:57 PM
Search
Nobby. Watch the trailer for #TheBrothersGrimsby. In theaters March 11.
36,192 likes 4,607 comments 19,796 shares
6,851,422 views
Like
Comment
Share
Manchester United with Adnan Januzaj and Cristiano Ronaldo.
1 hr •
Enjoy these three super Reds strikes at the Stadium of Light...

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

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{Minimize}} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J} x_{i,j} + \frac{\gamma}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{i,j} - q_{i,j})^2 \\ \text{subject to} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J} x_{i,j}s_{i,j} \geq C \\ & 0 \leq x_{i,j} \leq 1 \end{aligned}$$

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 $\boldsymbol{s} = \mathbf{P}\mathbf{x}$

- $$s_{ij} = p_{ij}^1 x_{i1} + p_{ij}^2 x_{i2} + \ldots + p_{ij}^J x_{iJ}$$

Quadratically Constrained Quadratic Problem (QCQP)

- Original Problem:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{Minimize}} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J} x_{i,j} + \frac{\gamma}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{i,j} - q_{i,j})^2 \\ \text{subject to} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J} x_{i,j}s_{i,j} \geq C \\ & 0 \leq x_{i,j} \leq 1 \end{aligned}$$

- Derived Problem:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{Minimize}} \quad & \mathbf{x}^T\mathbf{1} + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{q}\|^2 \\ \text{subject to} \quad & \mathbf{x}^T\mathbf{P}\mathbf{x} \geq C \\ & 0 \leq \mathbf{x} \leq 1 \end{aligned}$$

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.

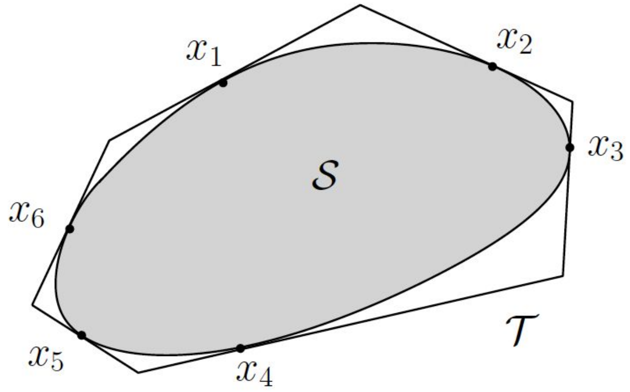
Methodology

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

- Original Problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & (\mathbf{x}-\mathbf{a})^T\mathbf{A}(\mathbf{x}-\mathbf{a}) \\ \text{subject to} \quad & (\mathbf{x}-\mathbf{b})^T\mathbf{B}(\mathbf{x}-\mathbf{b}) \leq \tilde{b}, \\ & \mathbf{C}\mathbf{x} = \mathbf{c}. \end{aligned}$$

Constraint Set:



QCQP to QP Approximations

- Derived problem becomes:

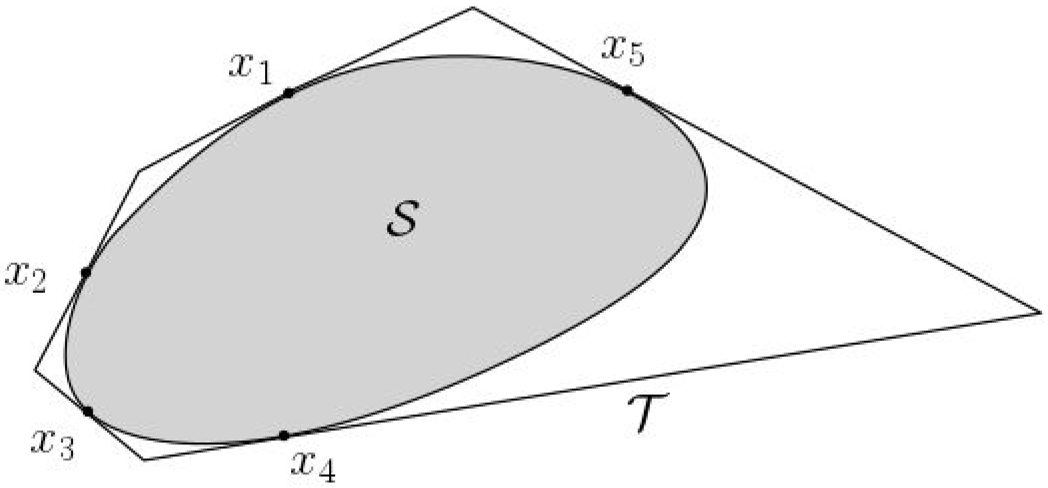
$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & (\mathbf{x}-\mathbf{a})^T \mathbf{A} (\mathbf{x}-\mathbf{a}) \\ \text{subject to} \quad & (\mathbf{x}-\mathbf{b})^T \mathbf{B} (\mathbf{x}_j - \mathbf{b}) \leq \tilde{b} \qquad \text{for } j = 1, \ldots, N \\ & \mathbf{C}\mathbf{x} = \mathbf{c}. \end{aligned}$$

- 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?

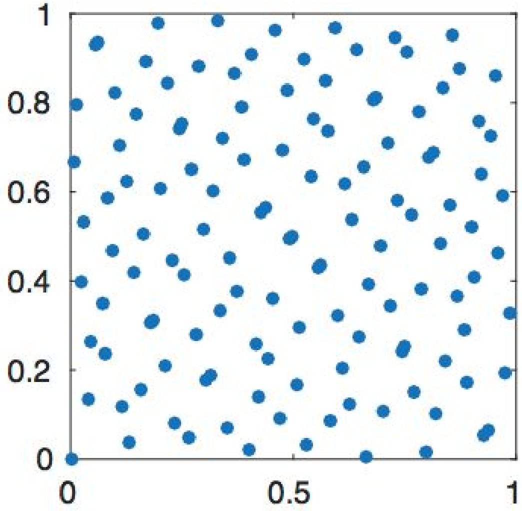
- If you are lucky it can be good, but if not it can be arbitrarily bad.

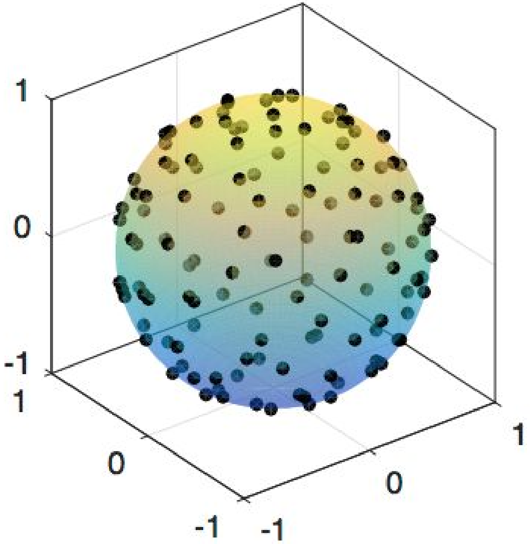
- Can even get an unbounded region

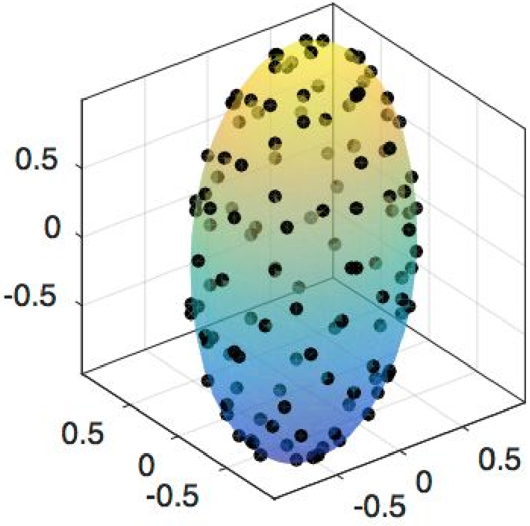


Optimal Choice of Points

- Low-Discrepancy Points







Theoretical Results

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

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

$$|f(\mathbf{x}^*(N)) - f(\mathbf{x}^*)| \leq C_2 g(N)$$

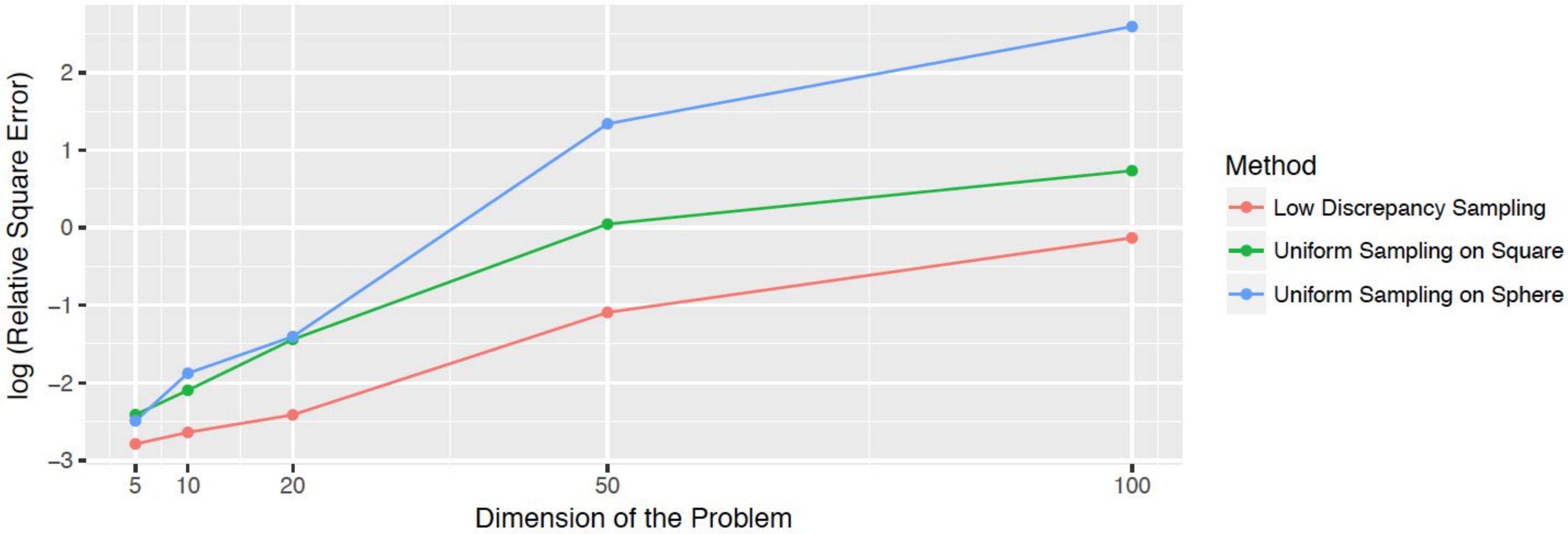
and the function g(N) can be explicitly identified for different constraint domains.

Comparative Study

Table 1: The optimal objective value and convergence time

n	Our method	Sampling on $[0,1]^n$	Sampling on $\mathbb{S}^n$	SDP relaxation	RLT relaxation	Exact
5	3.00 (4.61s)	2.99 (4.74s)	2.95 (6. 11s)	**3.07** (0.52s)	3.08 (0.51s)	**3.07** (0.49)
10	206.85 (5.04s)	205.21 (5.65s)	206.5 (5.26s)	**252.88** (0.53s)	**252.88** (0.51s)	**252.88** (0.51)
20	6291.4 (6.56s)	4507.8 (6.28s)	5052.2 (6.69s)	**6841.6** (2.05s)	**6841.6** (1.86s)	**6841.6** (0.54)
50	99668 (15.55s)	15122 (18.98s)	26239 (17.32s)	$\mathbf{1.11 \times 10^5}$ (4.31s)	1.08×10^5 (2.96s)	$\mathbf{1.11 \times 10^5}$ (0.64)
100	1.40×10^6 (58.41s)	69746 (1.03m)	1.24×10^6 (54.69s)	$\mathbf{1.62 \times 10^6}$ (30.41s)	1.52×10^6 (15.36s)	$\mathbf{1.62 \times 10^6}$ (2.30s)
1000	2.24×10^7 (14.87m)	8.34×10^6 (15.63m)	9.02×10^6 (15.32m)	NA	NA	NA
10^5	3.10×10^8 (25.82m)	7.12×10^7 (24.59m)	8.39×10^7 (27.23m)	NA	NA	NA
10^6	3.91×10^9 (38.30m)	2.69×10^8 (39.15m)	7.53×10^8 (37.21m)	NA	NA	NA

Comparative Study



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!