Chapter 1

Introduction
Classic optimization algorithms

\[ \min f(x) \]

two steps of designing an optimization algorithm:

- find a descent direction
- take a small step

two examples:

- gradient descent method:
  \[ x^{k+1} = x^k - t_k \nabla f(x^k) \]

- Newton’s method
  \[ x^{k+1} = x^k - t_k (\nabla^2 f(x^k))^{-1} \nabla f(x^k) \]
Example: linear regression

from a set of observations

\[ a_1^\top x = b_1, \ a_2^\top x = b_2, \ldots, a_m^\top x = b_m, \]

where \( a_i^\top = (a_{i1}, a_{i2}, \ldots, a_{in}) \in \mathbb{R}^n \), want to find the best \( x \)
The optimization formulation (a least-squares problem)

\[
\min \frac{1}{2} \| Ax - b \|_2^2
\]

where \( a_i^\top \) is the \( i \)-th row of matrix \( A \)

- gradient method: \( x^{k+1} = x^k - t_k A^\top (Ax^k - b) \)
- Newton’s method: \( x^{k+1} = x^k - t_k (A^\top A)^{-1} A^\top (Ax^k - b) \)
Structured optimization

$\ell_1$ norm regularized linear regression (LASSO in statistics, Compressed sensing in signal processing)

$$\min \left(\frac{1}{2}\|Ax - b\|_2^2 + \tau \|x\|_1\right)$$

where $\|x\|_1 = \sum_i |x_i|$

- nonsmooth problem
- gradient method? Newton’s method?
- subgradient method? (too slow)
- how to take advantage of the structure?
Why structured optimization?

- the needs of big data analytics
- extract useful information from high-dimensional massive data
- a priori information to the data or the solution

Challenges in big data analytics

- large scale
- incomplete
- dense
- noisy
- nonsmooth
LASSO (sparse data fitting)

$$\min \left( \frac{1}{2} \|Ax - b\|_2^2 \right) \quad \text{s.t.,} \quad \|x\|_1 \leq T$$
Compressed sensing

Signal $u$ is sparse under transformation $W$, i.e., $x = Wu$ is sparse. Reconstruct $u$ from limited observations $b = Ru$.

$$\min \|Wu\|_0, \text{ s.t. } Ru = b$$
$$\min \|Wu\|_1, \text{ s.t. } Ru = b$$

where $\|x\|_0$ denotes the number of nonzero components of $x$. 
MRI: Magnetic Resonance Imaging

- Is it possible to cut the scan time into half?
- MR images are sparse under some wavelet transform \( W \)
- solve \( \min_u \|W u\|_1 \), s.t. \( Ru = b \), where \( R \) is partial discrete Fourier transform
Portfolio Selection

- \( r_i \): random variable, the rate of return for stock \( i \)
- \( x_i \): the relative amount invested in stock \( i \)
- Return: \( r = r_1 x_1 + r_2 x_2 + \ldots + r_n x_n \)
- expected return: \( R = E(r) = \sum E(r_i) x_i = \sum \mu_i x_i \)
- Risk: \( V = Var(r) = \sum_{ij} \sigma_{ij} x_i x_j = x^\top \Sigma x \)

\[
\begin{align*}
\text{min} & \quad (1/2) x^\top \Sigma x \\
\text{s.t.} & \quad \sum_i \mu_i x_i \geq r_0 \\
& \quad \sum_i x_i = 1 \\
& \quad x_i \geq 0, \ i = 1, \ldots, n
\end{align*}
\]
Recommendation System
Recommendation System: Netflix

- Rating movies

- Get recommendations
Recommendation System: Netflix Prize

- 480,189 users rate 17,770 movies
- over 100 million ratings (1.2%)
- Data: incomplete rating matrix
- Decision: predict the missing ratings
- 1 million dollar prize
Recommendation System: Rating Matrix Completion

<table>
<thead>
<tr>
<th>User</th>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>9  5  6  8</td>
</tr>
<tr>
<td>8</td>
<td>8  4  9  ?</td>
</tr>
<tr>
<td>1</td>
<td>3  ?  5  4</td>
</tr>
<tr>
<td>8</td>
<td>?  10 4  1</td>
</tr>
</tbody>
</table>

- Rating matrix is believed to be low rank

\[
\begin{align*}
\min_X \; \text{rank}(X) \quad &\text{s.t.} \quad X_{ij} = M_{ij}, \forall (i, j) \in \Omega \\
\min_X \; \|X\|_* \quad &\text{s.t.} \quad X_{ij} = M_{ij}, \forall (i, j) \in \Omega,
\end{align*}
\]

where \(\|X\|_* := \sum \sigma_i(X)\) is the nuclear norm of matrix \(X\), and \(\sigma_i(X)\) is the \(i\)-th singular value of \(X\).
Robust PCA

- Classical PCA:
  - Obtains a low-dimensional expression of high-dimensional data in an $\ell_2$ sense.
  - Dimensionality reduction:
    \[
    \min_L \|M - L\|_F, \quad \text{s.t. } \text{rank}(L) \leq k.
    \]
  - Solution given by an SVD: $L = \sum_{i=1}^{k} \sigma_i u_i v_i^\top$.
  - PCA is very sensitive to gross errors
  - Gross errors occur frequently in Web Data Analysis, Bioinformatics etc.
Robust PCA

• Robust PCA:
  - Decompose $M = L + S$
  - $L$: low-rank, regular ratings
  - $S$: sparse, manipulated ratings

• Matrix decomposition

\[
\min_{L,S} \text{rank}(L) + \rho \|S\|_0, \text{ s.t. } L + S = M.
\]

\[
\min_{L,S} \|L\|_* + \rho \|S\|_1, \text{ s.t. } L + S = M.
\]
Surveillance video background extraction
Shadow and specularity removal from face images
Sparse Inverse Covariance Estimation

• \((X_1, \ldots, X_p)\) multivariate Gaussian \(\mathcal{N}(\mu, \Sigma)\)

• \((\Sigma^{-1})_{ij} = \text{cov}(X_i, X_j \mid \text{rest})\). \((\Sigma^{-1})_{ij} = 0 \iff X_i \perp X_j \mid X_{-i,j}\).

• Suppose we have iid noise \(\epsilon_i \sim \mathcal{N}(0, 1)\) and linear model: \(x = z + \epsilon_1, \ y = z + \epsilon_2, \ z = \epsilon_3\).

• Covariance matrix and inverse covariance matrix:

\[
\Sigma = \begin{pmatrix}
2 & 1 & 1 \\
1 & 2 & 1 \\
1 & 1 & 1 \\
\end{pmatrix}, \quad \Sigma^{-1} = \begin{pmatrix}
1 & 0 & -1 \\
0 & 1 & -1 \\
-1 & -1 & 3 \\
\end{pmatrix}
\]

• Convex formulation (Yuan and Lin 2007; Banerjee, El Ghaoui and d’Aspremont 2007; Friedman, Hastie and Tibshirani 2008)

\[
\min_S \left\langle \hat{\Sigma}, S \right\rangle - \log \det S + \rho \| S \|_1.
\]
Latent variable covariance estimation

- $X$: observed variables; $Y$: latent variables (hidden factors)
- $(X, Y)$ jointly follow a multivariate Gaussian distribution.

- Covariance matrix and inverse covariance matrix
  \[
  \Sigma = \begin{bmatrix}
  \Sigma_X & \Sigma_{XY} \\
  \Sigma_{YX} & \Sigma_Y
  \end{bmatrix}
  \quad \Theta = \begin{bmatrix}
  \Theta_X & \Theta_{XY} \\
  \Theta_{YX} & \Theta_Y
  \end{bmatrix}
  \]

- Inverse covariance matrix
  \[
  \Sigma_X^{-1} = \Theta_X - \Theta_{XY} \Theta_Y^{-1} \Theta_{YX}
  \]

- Inverse covariance matrix = “sparse - low-rank”

- Convex formulation (Chandrasekaran et al., 2010)
  \[
  \min_{R,S,L} \quad \langle R, \hat{\Sigma}_X \rangle - \log \det(R) + \alpha \|S\|_1 + \beta \text{Tr}(L)
  \]
  \[
  \text{s.t.} \quad R = S - L, \quad R \succeq 0, \quad L \succeq 0.
  \]
Classification