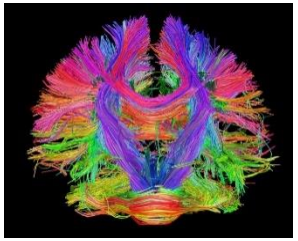



CS-570
Statistical Signal Processing

Lecture 4: Basics of Convex Optimization Algorithms

Spring Semester 2019

Grigorios Tsagkatakis

Today's Objectives

- Convex Optimization Algorithms

Disclaimer: Material used:

- Convex Optimization for Signal Processing and Communications - From Fundamentals to Applications, C.Y. Chi, W.C. Li, C.H. Lin
- Signal Processing and Networking for Big Data Applications, Z Han, M Hong, D Wang, 2017
- Convex Optimization – S. Boyd and L. Vandenberghe http://web.stanford.edu/~boyd/cvxbook/

Standard form of optimization

$$\begin{aligned} \min \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \ i = 1, \ldots, p \end{aligned}$$

- Problem domain $\mathcal{D} = \left\{ \bigcap_{i=0}^{m} \mathbf{dom}\ f_i \right\} \bigcap \left\{ \bigcap_{i=1}^{p} \mathbf{dom}\ h_i \right\}$
- Feasible set

$$\mathcal{C} = \left\{ \mathbf{x} \mid \mathbf{x} \in \mathcal{D}, \ f_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m, \ h_i(\mathbf{x}) = 0, \ i = 1, \ldots, p \right\}$$

- A problem is feasible if there exists at least one $\mathrm{x} \in C$; infeasible if $C = \emptyset$
- The optimization problem is said to be a convex optimization problem if the objective function and the set C are convex

Optimal solution

The optimal value $p^\star$ of the optimization problem is defined as

$$p^\star = \inf_{\mathbf{x}\in\mathcal{C}} f_0(\mathbf{x}) = \inf \{ f_0(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C} \}.$$

A point x is locally optimal (or a local minimizer) if there is an r > 0 such $f_0(x) = \inf \{ f_0(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C}, \|\mathbf{x} - x\|_2 \leq r \}$

A feasible point x with $f_0(\mathbf{x}) \leq p^\star + \epsilon \ (\text{where } \epsilon > 0)$ is called *ϵ-suboptimal*

Optimality criterion

- Assume that f_0 is differentiable, and that the associated optimization problem with the constraint set C given by

$$\begin{aligned} \min \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \ f_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m, \end{aligned}$$

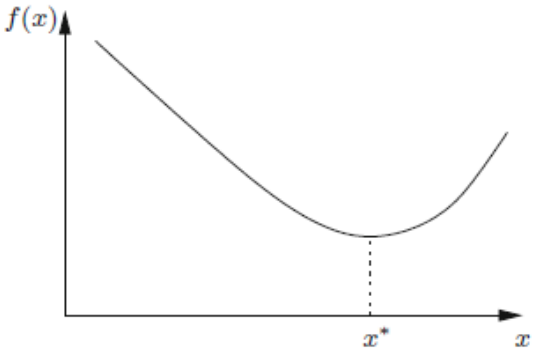
  is convex

- Then a point $\mathrm{x} \in C$ is optimal if and only if

$$\nabla f_0(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \geq 0, \ \forall \mathbf{y} \in \mathcal{C}.$$

Basic optimization method

Basic case $\underset{x}{\text{minimize}}\, f(x),\ x \in \mathbb{R},\ f \in C^2.$



Clearly x^* occurs where the slope is zero, i.e. where

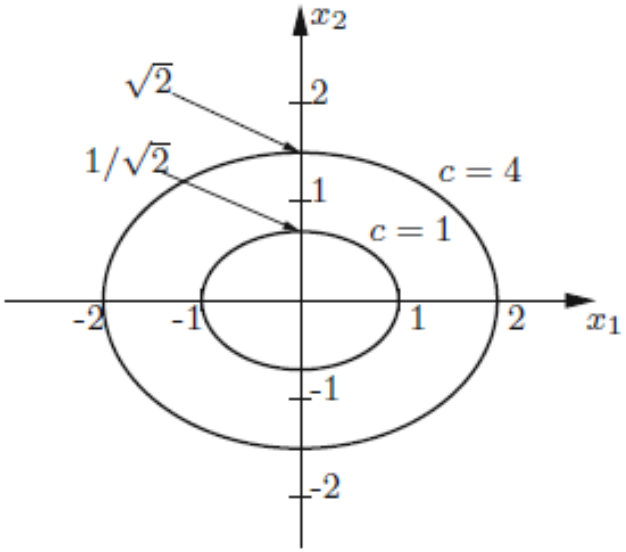
$$f'(x) = \frac{df(x)}{dx} = 0,$$

non-negative curvature is necessary at x^*, i.e. it is required that the second order condition

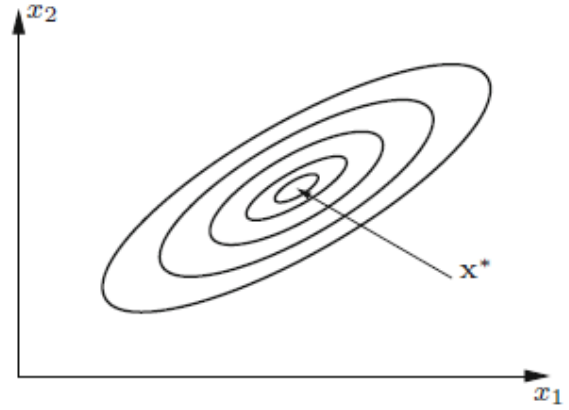
$$f''(x) = \frac{d^2 f(x)}{dx^2} > 0$$

must hold at x^* for a strong local minimum.

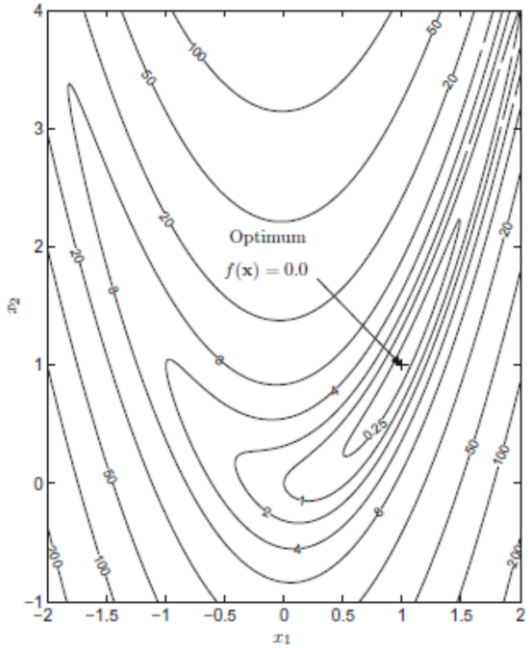
Higher dimensions



Contour representation of
$f(x) = x_1^2 + 2x_2^2$



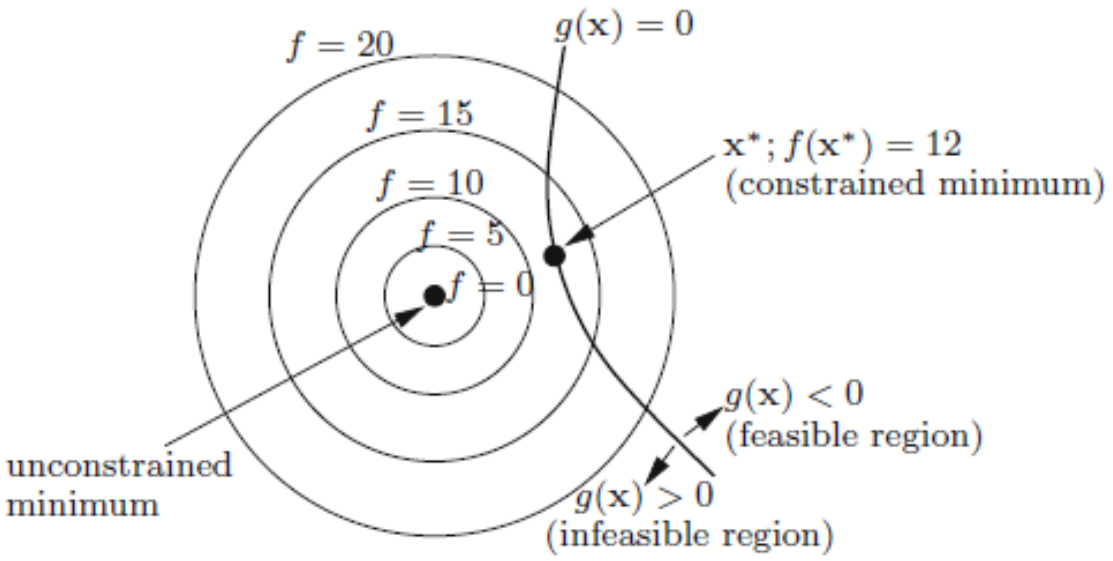
General quadratic function



Contour of the 2D Rosenbrock function
$f(x) = 10(x_2 - x_1^2)^2 + (1 - x_1\ )^2$

Inequality constrained problems

$$\min f(\mathbf{x})$$
$$\text{such that } g(\mathbf{x}) \leq 0.$$



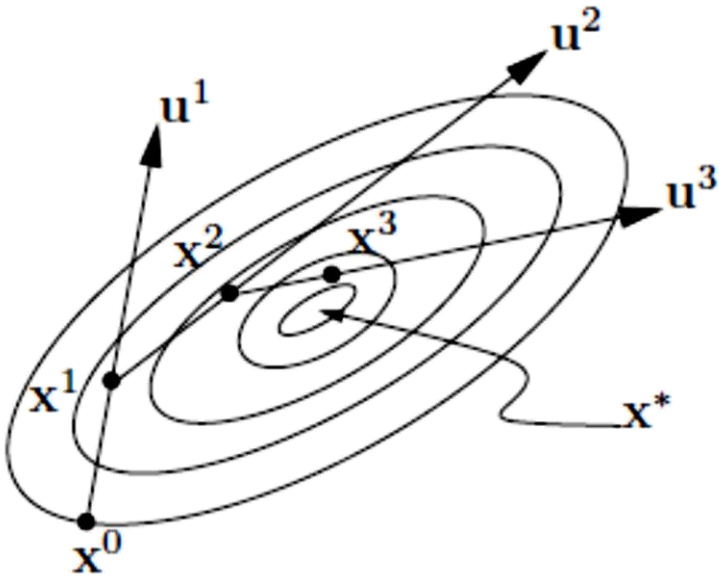
Unconstrained minimization

Direct search algorithms

- These algorithms require an initial estimate to the optimum point, denoted by $\mathbf{x}^0$.
- With this estimate as starting point, the algorithm generates a sequence of estimates $\mathbf{x}^0$, $\mathbf{x}^1$, $\mathbf{x}^2$, . . . , by successively searching *directly* from each point in a direction of *descent* $\mathbf{u}^{i+1}$ to determine the next point.
- The process is terminated if either no further progress is made, or if a point $\mathbf{x}^k$ is reached (for smooth functions) at which the first necessary condition $\nabla f(\mathbf{x}^k) = \mathbf{0}$ is sufficiently accurately satisfied

Line search descent methods

- Descend direction $\left.\frac{df(\mathbf{x}^i)}{d\lambda}\right|_{\mathbf{u}^{i+1}} = \nabla^T f(\mathbf{x}^i)\mathbf{u}^{i+1} < 0.$



First order methods

- Line search descent methods use the gradient vector $\nabla f(\mathbf{x})$ to determine the search direction for each iteration
- The simplest and most famous of these methods is the *method of steepest descent*, first proposed by Cauchy in 1847.

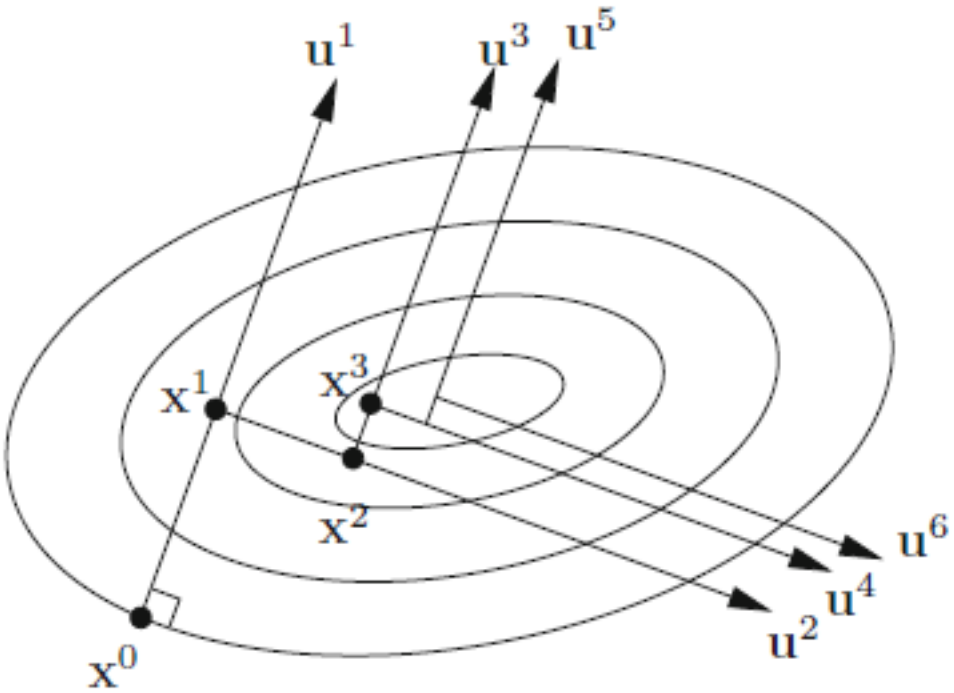
Given $\mathbf{x}^0$, do for iteration $i = 1, 2, \ldots$ until convergence:

1. set $\mathbf{u}^i = \dfrac{-\nabla f(\mathbf{x}^{i-1})}{\|\nabla f(\mathbf{x}^{i-1})\|}$
2. set $\mathbf{x}^i = \mathbf{x}^{i-1} + \lambda_i \mathbf{u}^i$ where λ_i is such that

$$F(\lambda_i) = f(\mathbf{x}^{i-1} + \lambda_i \mathbf{u}^i) = \min_{\lambda} f(\mathbf{x}^{i-1} + \lambda \mathbf{u}^i) \text{ (line search).}$$

steepest descent

Orthogonal zigzagging behaviour of the steepest descent method



Second order linear search descent methods

These methods are based on Newton's method for solving $\nabla f(\mathbf{x}) = \mathbf{0}$

- Given x^0
- $$\mathbf{x}^i = \mathbf{x}^{i-1} - \mathbf{H}^{-1}(\mathbf{x}^{i-1})\nabla f(\mathbf{x}^{i-1}),\ i = 1, 2, \ldots \qquad (2.17)$$

Modified version

- At iteration $\mathbf{u}^i = \boldsymbol{\Delta} = -\mathbf{H}^{-1}(\mathbf{x}^{i-1})\nabla f(\mathbf{x}^{i-1})$
- Then find $\min_{\lambda} f(\mathbf{x}^{i-1} + \lambda \mathbf{u}^i)$
- And set $\mathbf{x}^i = \mathbf{x}^{i-1} + \lambda_i \mathbf{u}^i$

Iterative Descent Methods

$$\mathbf{x}^{r+1} = \mathbf{x}^r + \alpha_r \mathbf{d}^r, \; r = 0, 1, \cdots$$

where, if $\nabla f(\mathbf{x}^r) \neq 0$, the direction $\mathbf{d}^r$ satisfies $\nabla f(\mathbf{x}^r)\mathbf{d}^r < 0$, and α^r is a positive stepsize

- General Case: Gradient descent methods

$$\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha_r \mathbf{D}^r \nabla f(\mathbf{x}^r), \; r = 0, 1, \cdots$$

  where $\mathbf{D}^r$ is a positive definite matrix

- Special case I: Steepest descent

$$\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha_r \nabla f(\mathbf{x}^r), \; r = 0, 1, \cdots$$

- Special case II: Newton's method

$$\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha_r \left(\nabla^2 f(\mathbf{x}^r)\right)^{-1} \nabla f(\mathbf{x}^r), \; r = 0, 1, \cdots$$

Convergence criteria

In practice the algorithm is terminated if some convergence criterion is satisfied. Usually termination is enforced at iteration i if one, or a combination, of the following criteria is met:

(i) $\|\mathbf{x}^i - \mathbf{x}^{i-1}\| < \varepsilon_1$

(ii) $\|\nabla f(\mathbf{x}^i)\| < \varepsilon_2$

(iii) $|f(\mathbf{x}^i) - f(\mathbf{x}^{i-1})| < \varepsilon_3.$

where ε_1, ε_2 and ε_3 are prescribed small positive tolerances.

Constrained convex optimization

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0, \quad i = 1, \cdots, m \\ & g_j(x) \leq 0, \quad j = 1, \cdots, n \end{aligned}$$

- **Reminder**: The problem is called convex problem if
  1. $f(x)$ is a convex function
  2. $h_i(x)$ is an affine function, i.e., $h_i(x) = Ax + b$
  3. $g_j(x)$ is a convex function

Lagrange multipliers

The Lagrangian can be formed using the Lagrangian multipliers $\lambda_i \geq 0$ and $\nu_i \in \mathbb{R}$

$$L(x, \lambda, \nu) = f(x) + \sum_{j=1}^{n} \lambda_j g_j(x) + \sum_{i=1}^{m} \nu_i h_i(x)$$

The Lagrangian dual function

$$L^*(\lambda, \nu) = \inf_{x \in X} L(x, \lambda, \nu) = \inf_{x \in X} f(x) + \sum_{j=1}^{n} \lambda_j g_j(x) + \sum_{i=1}^{m} \nu_i h_i(x)$$

The Dual Problem

$$\max_{\lambda, \nu} \quad L^*(\lambda, \nu), \quad \text{s.t. } \lambda \geq 0$$

λ_i and ν_i's can be viewed as "prices" for violating the constraints

Duality

Let f^* be the optimal value of $f(x)$

The Lagrangian dual L^* is

1. A concave function: even when the original problem is not convex
2. A lower bound: for $\lambda \geq 0$, $L^*(\lambda, \nu) \leq f^*$

Let d^* be the optimal objective of the dual

Weak duality: $d^* \leq f^*$

1. Always true
2. Non-trivial lower bound for hard problems
3. Useful in approximation algorithms

Strong duality: $d^* = f^*$

1. Does not hold in general
2. If holds, sufficient to solve the dual

KKT Condition

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0, \quad i = 1, \cdots, m \\ & g_j(x) \leq 0, \quad j = 1, \cdots, n \end{aligned}$$

Any optimal and dual pairs $\tilde{x}$ and $(\tilde{\lambda}, \tilde{\nu})$ must satisfy

$$\nabla f(\tilde{x}) + \sum_{j=1}^{n} \tilde{\lambda}_j \nabla g_j(\tilde{x}) + \sum_{i=1}^{m} \tilde{\nu}_i \nabla h_i(\tilde{x}) = 0_{K \times 1}$$

$g_j(\tilde{x}) \leq 0, \forall j = 1, ..., n,$ (primal feasibility)

$h_i(\tilde{x}) = 0, \forall i = 1, ..., m,$ (primal feasibility)

$\tilde{\lambda}_j \geq 0, \forall j = 1, ..., n,$ (dual feasibility)

$g_j(\tilde{x}) \times \tilde{\lambda}_j = 0, \forall j$ (complementarity).

Alternating Directions Method of Multipliers

The ADMM algorithm solves the following convex program

$$\begin{aligned} \min \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c} \\ & \mathbf{x} \in X, \quad \mathbf{z} \in Z \end{aligned}$$

$\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$ are the variables

f, g are two convex function, possible nonsmooth

$\mathbf{A}$, $\mathbf{B}$ are two known matrices, $\mathbf{c}$ is a known vector

Note: Two blocks of variables; separable in the objective, coupled by a linear equation

Main Benefit: Capable of dealing with $\mathbf{x}$ and $\mathbf{z}$ separately (in a BCD manner)

The ADMM Algorithm

Consider the Equality-Constrained convex problem

$$\min_{\mathbf{x}\in X,\mathbf{z}\in Z} \quad f(\mathbf{x})+g(\mathbf{z}), \quad \text{s.t. } \mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}=\mathbf{c}$$

Augmented Lagrangian

$$L_\rho(\mathbf{x},\mathbf{z};\mathbf{y}) = f(\mathbf{x})+g(\mathbf{z})+\langle \mathbf{y}, \mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}-\mathbf{c}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}-\mathbf{c}\|^2$$

Method of multipliers

$$(\mathbf{x}^{r+1},\mathbf{z}^{r+1}) = \arg\min_{\mathbf{x},\mathbf{z}} L_\rho(\mathbf{x},\mathbf{z};\mathbf{y}^r)$$

$$\mathbf{y}^{r+1} = \mathbf{y}^r + \rho\left(\mathbf{A}\mathbf{x}^{r+1}+\mathbf{B}\mathbf{z}^{r+1}-\mathbf{c}\right)$$

The ADMM Algorithm

The steps of the ADMM Algorithm is given below

$$\mathbf{x}^{r+1} = \arg\min_{\mathbf{x}\in X} L_\rho(\mathbf{x}, \mathbf{z}^r; \mathbf{y}^r)$$
$$\mathbf{z}^{r+1} = \arg\min_{\mathbf{z}\in Z} L_\rho(\mathbf{x}^{r+1}, \mathbf{z}; \mathbf{y}^r)$$
$$\mathbf{y}^{r+1} = \mathbf{y}^r + \rho\left(\mathbf{A}\mathbf{x}^{r+1} + \mathbf{B}\mathbf{z}^{r+1} - \mathbf{c}\right)$$

- Divide and conquer: Optimize $\mathbf{x}$ and $\mathbf{z}$ once (coordinate descent on L_ρ), then update the dual variable
- The primal problem is no longer solved exactly (where the efficiency comes from)

Optimality of ADMM

- Lagrangian

$$L(\mathbf{x}, \mathbf{z}; \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \langle \mathbf{y}, \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} \rangle$$

- KKT condition (suppose X, Z are both the whole space, no other constraints on $\mathbf{x}$, $\mathbf{z}$)

$$-\mathbf{A}^T\mathbf{y}^* \in \partial f(\mathbf{x}^*), \quad -\mathbf{B}^T\mathbf{y}^* \in \partial g(\mathbf{z}^*)$$
$$\mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{z}^* - \mathbf{c} = 0$$

- ADMM updates (optimality condition at each iteration)

$$-\mathbf{A}^T\mathbf{y}^{r+1} + \mathbf{A}^T\left(\mathbf{B}\mathbf{z}^{t+1} - \mathbf{B}\mathbf{z}^t\right) \in \partial f(\mathbf{x}^{r+1})$$
$$-\mathbf{B}^T\mathbf{y}^{r+1} \in \partial g(\mathbf{z}^{r+1})$$

- Optimality is achieved if the following are satisfied:

$$\mathbf{A}^T\left(\mathbf{B}\mathbf{z}^{t+1} - \mathbf{B}\mathbf{z}^t\right) = 0$$
$$\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{c} = 0$$

Convergence

- The ADMM converges under very mild condition
- Let us define the residue as

$$\mathbf{r}^r = \mathbf{A}\mathbf{x}^r + \mathbf{B}\mathbf{z}^r - \mathbf{c}$$

- **Claim**: Suppose that the problem is convex and feasible, then the following is true
  1. **Residue convergence**: $\mathbf{r}^k \to 0$ as $k \to \infty$
  2. **Objective convergence**: $f(\mathbf{x}^k) + g(\mathbf{z}^t) \to p^*$ as $k \to \infty$
  3. **Dual variable convergence**: $\mathbf{y}^k \to \mathbf{y}^*$ as $k \to \infty$, where $\mathbf{y}^*$ is the optimal dual solution

Matrix factorizations

LU: A square matrix $\mathbf{A}$ can be written $\mathbf{A} = \mathbf{LU}$ where $\mathbf{L}$ is lower-triangular and $\mathbf{U}$ is upper triangular. The main use is in the solution of a system of equations

$$\mathbf{Ax} = \mathbf{LUx} = \mathbf{b}$$

Cholesky: A Hermitian positive-definite matrix $\mathbf{A}$ can be factored as $\mathbf{A} = \mathbf{LL}^H$ where $\mathbf{L}$ is lower-triangular. It is used in simulations to compute a vector noise of specified covariance.

QR: A general $m \times n$ matrix $\mathbf{A}$ can be factored as $\mathbf{A} = \mathbf{QR}$ where $\mathbf{QQ}^H = \mathbf{I}$ and $\mathbf{R}$ is upper-triangular. It is used in the solution of least-squares problems.

LU Decomposition

The LU decomposition can be applied to any $m \times m$ matrix $\mathbf{A}$. The algorithm is essentially Gaussian elimination. It should be implemented with permutations to provide numerical stability. The result is three matrices, $\mathbf{L}$, $\mathbf{U}$ and $\mathbf{P}$ such that

$$\mathbf{PA} = \mathbf{LU}$$

The permutation matrix $\mathbf{P}$ is orthogonal: $\mathbf{P}^T\mathbf{P} = \mathbf{I}$.

A system of equations $\mathbf{Ax} = \mathbf{b}$ can be solved in steps. First, let $\mathbf{Ax} = \mathbf{P}^T\mathbf{LUx} = \mathbf{P}^T\mathbf{Ly} = \mathbf{b}$ where $\mathbf{y} = \mathbf{Ux}$. This leads to the system of equations

$$\mathbf{Ly} = \mathbf{Pb} = \mathbf{c}$$

The equations $\mathbf{Ly} = \mathbf{c}$ can be solved by forward substitution. Then $\mathbf{x} = \mathbf{Uy}$ can be solved by back substitution.

The Matlab call is `[L,U,P]=lu(A)`

Read the Matlab documentation on the function `mldivide` to see how the various factorizations are used by the \ operator.

LU example

In this example we will solve the system of equations $\mathbf{Ax} = \mathbf{b}$ given by

$$\begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 1 \\ 6 & 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 13 \\ 18 \\ 48 \end{bmatrix}$$

First, factor $\mathbf{A}$ using `[L,U,P]=A`.

$$\mathbf{LU} = \begin{bmatrix} 1 & 0 & 0 \\ 1/6 & 1 & 0 \\ 1/2 & 1/3 & 1 \end{bmatrix} \begin{bmatrix} 6 & 3 & 5 \\ 0 & 3/2 & 19/6 \\ 0 & 0 & -23/9 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 5 \\ 1 & 2 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$\mathbf{PA} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 1 \\ 6 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 5 \\ 1 & 2 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$\mathbf{LUx} = \mathbf{PAx} = \mathbf{Pb} = \begin{bmatrix} 48 \\ 13 \\ 18 \end{bmatrix}$$

LU example (cont)

Let $\mathbf{y} = \mathbf{U}\mathbf{x}$, and first solve for $\mathbf{y}$. We can then solve for $\mathbf{x}$.

$$\mathbf{L}\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 \\ 1/6 & 1 & 0 \\ 1/2 & 1/3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 48 \\ 13 \\ 18 \end{bmatrix}$$

$y_1 = 48,\ y_2 = 13 - 48/6 = 5,\ y_3 = 18 - 48/2 - 5/3 = -23/3.$ Then $\mathbf{U}\mathbf{x} = \mathbf{y}$ yields

$$\begin{bmatrix} 6 & 3 & 5 \\ 0 & 3/2 & 19/6 \\ 0 & 0 & -23/9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 48 \\ 5 \\ -23/3 \end{bmatrix}$$

$x_3 = 3,\ x_2 = (5 - 3 \cdot 19/6) \cdot 2/3 = -3,\ x_1 = (48 - 3(-3) - 5(3))/6 = 7$

Once the LU factorization is obtained, the solution involves a forward substitution followed by a backward substitution.

Cholesky Factorization

When $\mathbf{A}$ is Hermitian and positive definite it can be decomposed as

$$\mathbf{A} = \mathbf{L}\mathbf{L}^H \tag{1}$$

This is a special case of LU factorization.

The Cholesky factorization is unique when the diagonal entries of $\mathbf{L}$ are required to be positive.

The Cholesky decomposition is mainly used for the numerical solution of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$. Writing $\mathbf{A}\mathbf{x} = \mathbf{b}$ as

$$\mathbf{L}\mathbf{y} = \mathbf{b} \quad \text{and} \quad \mathbf{L}^H\mathbf{x} = \mathbf{y}$$

allows solving a triangular form first for $\mathbf{y}$ and then for $\mathbf{x}$.

Correlated Random Numbers If $\mathbf{R}$ is a symmetric covariance matrix then it has a Cholesky factorization $\mathbf{R} = \mathbf{L}\mathbf{L}^T$. Let $\mathbf{y} = \mathbf{L}\mathbf{x}$ where $\mathbf{x}$ are uncorrelated unit variance random numbers. Then $E[\mathbf{y}\mathbf{y}^T] = \mathbf{L}E(\mathbf{x}\mathbf{x}^T)\mathbf{L}^T = \mathbf{L}\mathbf{L}^T = \mathbf{R}$.

Eigenvector and Eigenvalue

- An eigenvector **x** of a linear transformation *A* is a non-zero vector that, when *A* is applied to it, does not change direction.
- Applying *A* to the eigenvector only scales the eigenvector by the scalar value λ, called an eigenvalue.

$$Ax = \lambda x, \quad x \neq 0.$$

Eigenvector and Eigenvalue

- We want to find all the eigenvalues of A:
- Which can we written as: $Ax = \lambda x, \quad x \neq 0.$
- Therefore: $Ax = (\lambda I)x \qquad x \neq 0.$

$$(\lambda I - A)x = 0, \quad x \neq 0.$$

Eigenvector and Eigenvalue

- We can solve for eigenvalues by solving: $(\lambda I - A)x = 0, \quad x \neq 0.$

- Since we are looking for non-zero **x**, we can instead solve the above equation as:

$$|(\lambda I - A)| = 0.$$

Properties

- The trace of a A is equal to the sum of its eigenvalues:

$$\mathrm{tr}A = \sum_{i=1}^{n} \lambda_i.$$

- The determinant of A is equal to the product of its eigenvalues

$$|A| = \prod_{i=1}^{n} \lambda_i.$$

- The rank of A is equal to the number of non-zero eigenvalues of A.
- The eigenvalues of a diagonal matrix D = diag(d_1, . . . d_n) are just the diagonal entries d_1, . . . d_n

Spectral theory

- We call an eigenvalue λ and an associated eigenvector an **eigenpair**.
- The space of vectors where (A − λI) = 0 is often called the **eigenspace** of A associated with the eigenvalue λ.
- The set of all eigenvalues of A is called its **spectrum**:

$$\sigma(A) = \{\lambda \in \mathbb{C} : \lambda I - A \text{ is singular}\}.$$

Spectral theory

- The magnitude of the largest eigenvalue (in magnitude) is called the spectral radius

$$\rho(A) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$$

Where C is the space of all eigenvalues of A

Spectral theory

- The spectral radius is bounded by infinity norm of a matrix:
- Proof: Let λ and v be an eigenpair of A:

$$\rho(A) = \lim_{k \to \infty} \|A^k\|^{1/k}$$

$$|\lambda|^k \|\mathbf{v}\| = \|\lambda^k \mathbf{v}\| = \|A^k \mathbf{v}\| \leq \|A^k\| \cdot \|\mathbf{v}\|$$

and since $\mathbf{v} \neq 0$ we have

$$|\lambda|^k \leq \|A^k\|$$

and therefore

$$\rho(A) \leq \|A^k\|^{\frac{1}{k}}.$$

Diagonalization

- An n × n matrix A is diagonalizable if it has n linearly independent eigenvectors.
- Most square matrices (in a sense that can be made mathematically rigorous) are diagonalizable:
  - Normal matrices are diagonalizable
  - Matrices with n distinct eigenvalues are diagonalizable

Lemma: Eigenvectors associated with distinct eigenvalues are linearly independent.

Diagonalization

- Eigenvalue equation:

$$AV = VD$$
$$A = VDV^{-1}$$

  - Where D is a diagonal matrix of the eigenvalues

$$\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

Diagonalization

- Eigenvalue equation:

$$AV = VD$$

$$A = VDV^{-1}$$

- Assuming all λ_i's are unique:

  - Remember that the inverse of an orthogonal matrix is just its transpose and the eigenvectors are orthogonal

$$A = VDV^{T}$$

Symmetric matrices

- Properties:
  - For a symmetric matrix A, all the eigenvalues are real.
  - The eigenvectors of A are orthonormal.

$$A = V D V^T$$

Some applications of Eigenvalues

- PageRank
- Schrodinger's equation
- PCA