COMP4680 Notes

Advanced Topics in Machine Learning: Optimization

2023 S2

0 Introduction

A mathematical optimisation problem requires us to minimize an objective function $f_0(x)$, subject to constraint functions $f_i(x) \leq b_i$ for $i = 1, \ldots, m$.

A solution, or optimal point, x^*, has the smallest value of f_0 among all vectors that satisfy the constraints.

There are three main types of problems which can be solved to different extents:

0.1 Least-squares Problems

Least-squares problems attempt to minimize $\|Ax - b\|_2^2$ for some matrix A and vector b.

There exists an analytical solution $x^* = (A^T A)^{-1} A^T b$, there are reliable and efficient algorithms with computation time $O(n^2 k)$ when $A \in \mathbb{R}^{k \times n}$, less if A satisfies certain structures.

0.2 Convex Optimization Problems

Convex optimization problems are optimization problems where both the objective and constraint functions are convex, and is a superset of least-squares problems.

There is no analytical solution, but there are algorithms with computation time $\max(O(n^3), O(n^2 m), F)$ where F is the cost of evaluating f_i's and their second derivatives.

0.3 Nonconvex Optimization Problems

There is no general way to solve nonconvex optimization problems: they all involve some kind of compromise.

We may use local optimization methods (nonlinear programming), which is fast and finds a local minima around an initial guess, but may not be the global minima.

Or we may use global optimization methods, which finds the global solution but requires exponential time complexity.

1 Preliminaries

1.1 Sets

A set, denoted as $S = \{a_1, \ldots, a_n\}$, is a collection of distinct objects.

Some common notations:

- $a \in S$ denotes a is an element of S
- $S \subseteq T$ denotes S is a subset of T, that is, every element of S is also an element of T
- $S \cup T$ denotes the union of S and T, that is, the set of all elements that are in S or T
- $S \cap T$ denotes the intersection of S and T, that is, the set of all elements that are in both S and T
- $S \times T$ denotes the Cartesian product of S and T, that is, the set of all ordered pairs (s, t) where $s \in S$ and $t \in T$
- $S \setminus T$ denotes the set difference of S and T, that is, the set of all elements that are in S but not in T

Some common sets:

- $\mathbb{R}$ is the set of real numbers
- $\mathbb{R}^n$ is the set of n-dimensional real vectors
- $\mathbb{R}^{m \times n}$ is the set of $m \times n$ real matrices
- $\mathbb{C}$ is the set of complex numbers
- $\mathbb{Z}$ is the set of integers
- $\mathbb{R}_+$ is the set of nonnegative real numbers
- $\mathbb{R}_{++}$ is the set of positive real numbers
- $\emptyset$ is the empty set
- $[a, b]$ is the closed interval from a to b (i.e. $\{x \in \mathbb{R} \mid a \leq x \leq b\}$)
- (a, b) is the open interval from a to b (i.e. $\{x \in \mathbb{R} \mid a < x < b\}$)
- $[a, b)$ and $(a, b]$ are half-open intervals, defined similarly

Open and Closed Sets

A subset $S \subseteq \mathbb{R}$ is **open** if for every $x \in S$, there exists $\epsilon > 0$ such that if $\|y - x\|_2 < \epsilon$, then $y \in S$.

A subset $S \subseteq \mathbb{R}$ is **closed** if its complement $\mathbb{R} \setminus S$ is open.

A subset $S \subseteq \mathbb{R}$ is **bounded** if there exists $M > 0$ such that $\|a - b\|_2 \leq M$ for all $a, b \in S$.

Infimum and Supremum

The **infimum** of a set $S \subseteq \mathbb{R}$, written as $\inf(S)$, is the largest $y \in \mathbb{R}$ such that $y \leq x$ for all $x \in S$. If no such y exists, we say $\inf(S) = -\infty$.

The **supremum** of a set $S \subseteq \mathbb{R}$, written as $\sup(S)$, is the smallest $y \in \mathbb{R}$ such that $y \geq x$ for all $x \in S$. If no such y exists, we say $\sup(S) = \infty$.

We define $\inf(\emptyset) = \infty$ and $\sup(\emptyset) = -\infty$.

1.2 Functions

A function $f : A \to B$ is a mapping from its **domain** A to its **codomain** B.

If $U \subseteq A$ and $V \subseteq B$, we define the **image** of U under f as $f(U) = \{f(x) \mid x \in U\} \subseteq B$, and the **preimage** of V under f as $f^{-1}(V) = \{x \in A \mid f(x) \in V\} \subseteq A$.

1.3 Vector Spaces

A vector space V is a set with two operations, vector addition and scalar multiplication, that satisfy the following axioms:

- $x + y = y + x$ (commutativity of vector addition)
- $(x + y) + z = x + (y + z)$ (associativity of vector addition)
- $x + \mathbf{0} = x$ (additive identity)
- $\forall x \in V, \exists y \in V$ such that $x + y = 0$, we write y as $-x$ (additive inverse)
- $\alpha(x + y) = \alpha x + \alpha y$ (right distributivity)
- $(\alpha + \beta)x = \alpha x + \beta x$ (left distributivity)
- $1x = x$ (multiplicative identity)

We define the **zero vector** as a vector with all elements equal to 0, and the **ones vector** as a vector with all elements equal to 1.

Euclidean Norm

The Euclidean norm of a vector $\mathbf{v} = (v_1, \ldots, v_n)$ is

$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + x_n^2}$$

$\|\mathbf{v}\|_2$ measures the length of $\mathbf{v}$.

The norm satisfies:

- $\|\alpha\mathbf{v}\| = |\alpha|\|\mathbf{v}\|$

- $\|\mathbf{u}+\mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ (triangle inequality)
- $\|\mathbf{v}\| \geq 0$ and $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = \mathbf{0}$ (positive definiteness)

There are other norms such as $\| \quad \|_1$ and $\| \quad \|_\infty$.

Inner Products

The inner product of two vectors $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{v} = (v_1, \ldots, v_n)$ is defined by

$$\langle \mathbf{u}, \mathbf{v} \rangle = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n.$$

The inner product satisfies:

- $\langle \alpha \mathbf{u}, \mathbf{v} \rangle = \alpha \langle \mathbf{u}, \mathbf{v} \rangle$
- $\langle \mathbf{u}_1 + \mathbf{u}_2, \mathbf{v} \rangle = \langle \mathbf{u}_1, \mathbf{v} \rangle + \langle \mathbf{u}_2, \mathbf{v} \rangle$
- $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
- $\langle \mathbf{v}, \mathbf{v} \rangle = \|\mathbf{v}\|_2^2$

Subspaces

A subspace of a vector space is a subset of the vector space that is also a vector space.

Independence

A set of vectors $v_1, \ldots, v_n$ is (linearly) independent if and only if $\alpha_1 v_1 + \cdots + \alpha_n v_n = 0$ implies $\alpha_1 = \cdots = \alpha_n = 0$.

Conversely, if a set of vectors is linearly dependent, we can write one of the vectors as a linear combination of the others.

Bases

The set of vectors $\{v_1, \ldots, v_n\}$ form a basis of a vector space V if

- they are linearly independent
- they span V, that is, every vector in V can be written as a linear combination of the vectors in the set

Equivalently, $\{v_1, \ldots, v_n\}$ form a basis for V if every $v \in V$ can be uniquely expressed as $v = \alpha_1 v_1 + \cdots + \alpha_n v_n$.

We define the **dimension** of a vector space V to be the number of vectors in any basis of V.

The standard basis of $\mathbb{R}^n$ is the set of vectors $\{e_1, \ldots, e_n\}$ where e_i is the vector with a 1 in the i^{th} position and 0 elsewhere.

1.4 Matrices

A matrix $A \in \mathbb{R}^{m \times n}$ is a rectangular array of real numbers with m rows and n columns.

We write A_{ij} for the entry in the i^{th} row and j^{th} column of A.

A $n \times 1$ matrix is called a (column) **vector**, and a $1 \times n$ matrix is called a row **vector**.

We say a matrix is **diagonal** if its nonzero entries are all on the main diagonal (top left to bottom right).

The **zero matrix**, denoted $\mathbf{0}_{m \times n}$, is the matrix with all entries equal to zero.

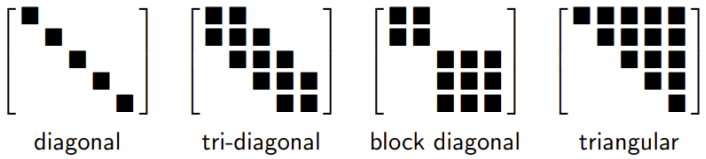
The **identity matrix**, denoted $\boldsymbol{I}_n$, is the $n \times n$ matrix with ones on the main diagonal and zeros elsewhere.

Special Types of Matrices

A matrix is **triangular** if all its entries above or below the main diagonal are zero. In particular, we refer to a matrix as **upper triangular** if all its entries below the main diagonal are zero, and **lower triangular** if all its entries above the main diagonal are zero.

A matrix is **block diagonal** if it is diagonal and each diagonal entry is itself a matrix.

A matrix is **tri-diagonal** if it has nonzero entries only on the main diagonal and the diagonals immediately above and below the main diagonal.



Matrix Transpose

Transpose, denotes as ^{T}, flips a matrix over is main diagonal, i.e. if A is an $m \times n$ matrix then A^T is an $n \times m$ matrix. It satisfies the following properties:

- $(A^T)^T = A$
- $(AB)^T = B^T A^T$
- $(A + B)^T = A^T + B^T$

If a matrix A satisfies $A = A^T$ we say A is **symmetric**.

If a matrix A satisfies $A = -A^T$ we say A is **anti-symmetric**.

Every square matrix A can be written as the sum of a symmetric part and an anti-symmetric part:

$$A = \underbrace{\frac{1}{2}\left(A + A^T\right)}_{\text{symmetric}} + \underbrace{\frac{1}{2}\left(A - A^T\right)}_{\text{anti-symmetric}}$$

Notation for Symmetric Matrices

We write

- $\mathbb{S}^n$ for the set of symmetric $n \times n$ matrices
- $\mathbb{S}^n_+ = \{X \in \mathbb{S}^n \mid X \preceq 0\}$ for the positive semi-definite $n \times n$ matrices

$$X \in \mathbb{S}^n_+ \Leftrightarrow z^T X z \geq 0 \text{ for all } z.$$

- $\mathbb{S}^n_{++} = \{X \in \mathbb{S}^n \mid X \prec 0\}$ for the positive definite $n \times n$ matrices

Matrix Addition

Two matrices of the same size can be added together: we simply add the corresponding elements in each matrix.

Matrix Multiplication

The product of two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$ is an $m \times p$ matrix with elements

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}.$$

Matrix multiplication satisfies:

- $(AB)C = A(BC)$ (associativity)
- $A(B + C) = AB + AC$ (left distributivity)
- $(A + B)C = AC + BC$ (right distributivity)

but matrix multiplication is not commutative: $AB \neq BA$ generally.

Null Space

The null space of a matrix $A \in \mathbb{R}^{m \times n}$ is defined as

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\}.$$

$\mathcal{N}(A)$ can be interpreted as

- the set of all vectors mapped to zero by $y = Ax$
- the set of all vectors orthogonal to the rows of A

Range Space

The range space of a matrix $A \in \mathbb{R}^{m \times n}$ is defined as

$$\mathcal{R}(A) = \{Ax \mid x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m.$$

$\mathcal{R}(A)$ can be interpreted as

- the set of all vectors that can be "hit" by $y = Ax$
- the span of the columns of A
- the set of all vectors y such that $Ax = y$ has a solution

Orthogonal Complement

The orthogonal complement of $V \subseteq \mathbb{R}^n$ is defined as

$$V^\perp = \left\{x \mid z^T x = 0 \text{ for all } z \in V\right\}.$$

We have $V \oplus V^\perp = \mathbb{R}^n$.
A result from the Fundamental Theorem of Linear Algebra states that $\mathcal{N}(A) = \mathcal{R}(A^T)^\perp$.

Rank

The rank of a matrix $A \in \mathbb{R}^{m \times n}$ is

$$\text{rank}(A) = \dim \mathcal{R}(A).$$

- $\text{rank}(A) = \text{rank}(A^T)$
- $\text{rank}(A)$ is the maximum number of independent columns (or rows) of A. Hence $\text{rank}(A) \leq \min\{m, n\}$.
- $\text{rank}(A) + \dim\mathcal{N}(A) = n$ (rank-nullity)

We say a matrix A is **full rank** if $\text{rank}(A) = \min\{m, n\}$.
The rank of the product of two matrices satisfies

$$\text{rank}(AB) \leq \min\left\{\text{rank}(A), \text{rank}(B)\right\}.$$

If $A \in \mathbb{R}^{m \times n}$ has rank r then A can be factored as BC with $B \in \mathbb{R}^{m \times r}$ and $C \in \mathbb{R}^{r \times n}$.

Trace

The trace of a square matrix $A \in \mathbb{R}^{n \times n}$ is the sum of its diagonal entries, i.e.

$$\mathrm{tr}(A) = \sum_{j=1}^{n} A_{jj}.$$

Trace satisfies the following properties:

- $\mathrm{tr}(A) = \mathrm{tr}(A^T)$
- $\mathrm{tr}(\alpha A + \beta B) = \alpha \mathrm{tr}(A) + \beta \mathrm{tr}(B)$
- if AB is square then $\mathrm{tr}(AB) = \mathrm{tr}(BA)$

Determinant

The determinant of a square matrix $A \in \mathbb{R}^{n \times n}$ is a function $\det : \mathbb{R}^{n \times n} \to \mathbb{R}$ that satisfies the following properties:

- $\det \boldsymbol{I} = 1$
- $\det \alpha A = \alpha^n \det A$
- swapping any two rows/columns changes the sign of the determinant
- $\det AB = \det A \det B$

We can interpret the determinant as the volume of the parallelepiped spanned by the rows (or columns) of A.

Matrix Inverse

The inverse of a square matrix $A \in \mathbb{R}^{n \times n}$ is a matrix A^{-1} such that

$$AA^{-1} = A^{-1}A = \boldsymbol{I}$$

A matrix is **invertible** (i.e. has an inverse) if and only if $\det A \neq 0$. This is equivalent to:

- the columns/rows of A form a basis for $\mathbb{R}^n$
- $y = Ax$ has a unique solution for all $x \in \mathbb{R}^n$
- A is full-rank (i.e. $\mathcal{N}(A) = \{0\}$ and $\mathcal{R}(A) = \mathbb{R}^n$)
- $\det A^T A = \det AA^T \neq 0$

Cauchy-Schwarz Inequality

For any vectors $x, y \in \mathbb{R}^n$, we have that

$$\left|x^T y\right| \leq \|x\|_2 \|y\|_2.$$

The angle between vectors in $\mathbb{R}^n$ is given by

$$\theta = \cos^{-1}\left(\frac{x^T y}{\|x\|_2 \|y\|_2}\right).$$

- If x and y are aligned then $x^T y =$

Eigenvalues and Eigenvectors

$\lambda \in \mathbb{C}$ is an eigenvalue of $A \in \mathbb{R}^{n \times n}$ if

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

Equivalently, there exists a non-zero $v \in \mathbb{C}^n$ such that $(\lambda I - A)v = 0$, or $Av = \lambda v$. Any such v here is called an eigenvector of A, associated with eigenvalue λ.

The eigenvalues of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ are real. Moreover, there exists a set of orthogonal eigenvectors $q_1, \ldots, q_n$ such that $Aq_i = \lambda_i q_i$ and $q_i^T q_j = 0$ if $i \neq j$.

In matrix form, there is an orthonormal Q such that $A = Q\Lambda Q^T$.

Norm Matrices

A matrix norm is a function $\| \quad \| : \mathbb{R}^{m \times n} \to \mathbb{R}$ that, similar to vector norms, satisfy linearity, positive definiteness, and the triangle inequality.

- Induced norms: $\|A\| = \sup\{\|Ax\| \mid x \in \mathbb{R}^n, \|x\| \leq 1\}$
- Frobenius norm: $\|A\|_F = \sqrt{\left\{\sum_{ij} a_{ij}^2\right\}}$
- Nuclear norm: $\|A\|_* = \sum_i \sigma_i(A) = \mathrm{tr}(\sqrt{A^T A})$

Square matrices also satisfy the sub-multiplicative property:

$$\|AB\| \leq \|A\|\|B\|.$$

1.5 Matrix Factorization

LU Factorization

Every nonsingular matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$A = PLU$$

where P is a permutation matrix, L is unit lower triangular, and U is upper triangular and non-singular.

Cholesky Factorization

Every symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$A = LL^T$$

where L is lower triangular and non-singular with positive diagonal elements.

Singular Value Decomposition

Any matrix A can be decomposed as

$$A = U\Sigma V^T$$

where $A \in \mathbb{R}^{m \times n}$ has rank r, $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{n \times r}$ which satisfy $U^T U = I$ and $V^T V = I$, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

Since $A^T A = V\Sigma^2 V^T$ we have v_i are the eigenvectors of $A^T A$. Similarly, u_i are the eigenvectors of AA^T.

We can use SVD to interpret a linear map $y = Ax$ as follows:

- we compute coefficients of x along the input directions $v_1, \ldots, v_r$
- scale the coefficients by σ_i
- re-constitute along the output directions $u_1, \ldots, u_r$

Here, v_1 is the most sensitive input direction, and u_1 is the highest gain output direction.

Matrix Calculus

We can compute partial derivatives of a function $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ as

$$\frac{\partial f(x)}{\partial x_{ij}} = \lim_{\alpha \to 0} \frac{f(x + \alpha e_i e_j^T) - f(x)}{\alpha}.$$

We can also compute the gradient (Jacobian) of f as

$$\nabla_A f(A) = \begin{pmatrix} \frac{\partial f}{\partial A_{11}} & \frac{\partial f}{\partial A_{12}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \frac{\partial f}{\partial A_{21}} & \frac{\partial f}{\partial A_{22}} & \cdots & \frac{\partial f}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \frac{\partial f}{\partial A_{m2}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{pmatrix}$$

Partial derivatives are linear:

- $\nabla_A(f + g) = \nabla_A f + \nabla_A g$
- $\nabla_A(tf) = t\nabla_A f$

Chain rule and product rule also extend to matrix calculus.

In vector calculus, the **Hessian** of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the matrix of second-order partial derivatives of f, i.e.

$$\nabla_x^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

1.6 Probability Theory

A probability distribution is a function that maps outcomes of an experiment to probabilities:

- for discrete variables we have **probability mass functions**
- for continuous variables we have **probability density functions**

The **mean** or **expected value** of a random variable is the sum of possible values weighted by their probabilities:

$$\mathbb{E}[X] = \int_x xP(X = x)\,\mathrm{d}x$$

The **variance** of a random variable X is $\mathbb{E}[(X - \mathbb{E}[X])^2]$.

1.7 Geometric Concepts

Lines

A line through two points x_1 and x_2 has the equation $x = \theta x_1 + (1 - \theta)x_2$.

The **line segment** between x_1 and x_2 is the set of points $x = \theta x_1 + (1 - \theta)x_2$ for $0 \leq \theta \leq 1$.

Affine Sets

An affine set contains the line through any two distinct points in the set: if $x_1, x_2 \in S$ then $\theta x_1 + (1 - \theta)x_2 \in S$.

Every affine set can be expressed as the solution set of a system of linear equations.

Convex Sets

A convex set contains the line segment between any two distinct points in the set: if $x_1, x_2 \in S$ then $\theta x_1 + (1 - \theta)x_2 \in S$ for $0 \leq \theta \leq 1$.

Common examples:

- nonnegative orthant: $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x_i \geq 0\}$
- positive semidefinite matrices: $\mathbb{S}^n_+ = \left\{X \in \mathbb{R}^{n \times n} \mid z^T X z \geq 0, z \in \mathbb{R}^n\right\}$

Convex Combinations and Hulls

A convex combination of $x_1, \ldots, x_k$ is any point of the form

$$x = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_k x_k$$

where $\theta_1 + \theta_2 + \ldots + \theta_k = 1$ and $\theta_i \geq 0$ for all i.

The convex hull of a set S, $\text{conv}(S)$, is the set of all convex combinations of points in S.

Convex Cones

A **conic combination** of points x_1 and x_2 is any point of the form

$$x = \theta_1 x_1 + \theta_2 x_2$$

with $\theta_1, \theta_2 \geq 0$.

A cone is a set containing all non-negative multiples of its points (i.e. if $x \in C$ then $\alpha x \in C$ for all $\alpha \geq 0$).

A convex cone is a set containing all conic combinations of its points.

Hyperplanes and Halfspaces

A hyperplane is a set of the form $\left\{x \mid a^T x = b\right\}$ with $a \neq 0$.

A halfspace is a set of the form $\left\{x \mid a^T x \leq b\right\}$ with $a \neq 0$.

In the 3D case, a plane is a hyperplane while a halfspace is everything on one side of the plane.

Hyperplanes are affine and convex, and halfspaces are convex.

Euclidean Balls and Ellipsoids

A Euclidean ball with center x and radius r is a set $B(x, r) = \{y \mid \|y - x\|_2 \leq r\}$.

An ellipsoid is a set of the form

$$\left\{y \mid (y - x)^T P^{-1} (y - x) \leq 1\right\}$$

with $P \in \mathbb{S}^n_{++}$ (symmetric positive definite).

Alternatively, we can represent a ball as

$$B(x, r) = \{x + ru \mid \|u\|_2 \leq 1\}$$

and an ellipsoid as

$$\{x + Au \mid \|u\|_2 \leq 1\}$$

with A a square, nonsinguar matrix.

Norm Balls and Cones

A norm ball with center x and radius r is the set $\{y \mid \|y - x\| \leq r\}$.

A norm cone is the set $\{(x, t) \mid \|x\| \leq t\}$. Norm balls and cones are convex.

Polyhedra

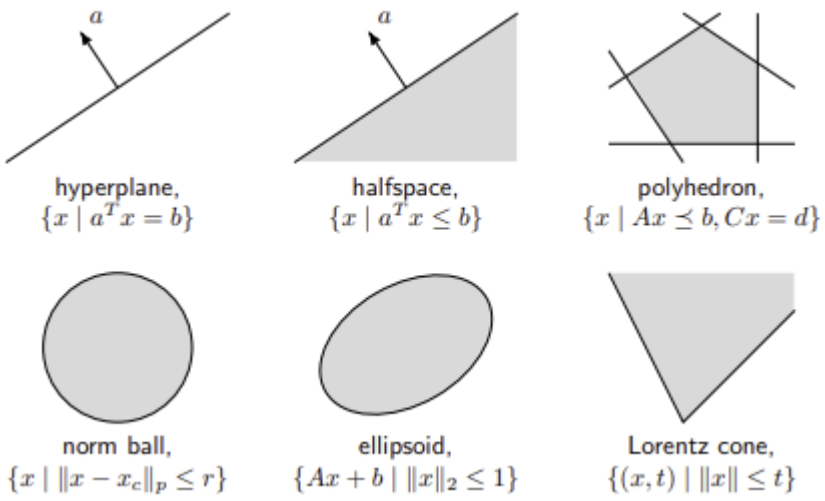
A polyhedron is the solution set of finitely many linear inequalities and equalities

$$Ax \preceq b, Cx = d$$

where $\preceq$ is componentwise inequality.

So a polyhedron is the intersection of a finite number of halfspaces and hyperplanes. Polyhedra are convex sets.

Summary of Convex Sets



Obtaining Convex Sets

We can either show a set C is convex by applying the definition, or obtain C using the following properties:

- the intersection of any number of convex sets is convex
- the image of a convex set under an affine map is convex (recall affine maps are of the form $f(x) = Ax + b$)
- the preimage of a convex set under an affine map is convex
- perspective functions preserve convexity: these are functions of the form

$$P(x, t) = x/t, t > 0$$

- linear-fractional functions preserve convexity: these are functions of the form

$$f(x) = \frac{Ax + b}{c^T x + d}, c^T x + d > 0$$

Generalized Inequalities

A convex cone $K \subseteq \mathbb{R}^n$ is a **proper cone** if

- K is closed (contains its boundary)
- K is solid (has nonempty interior)
- K is pointed (contains no line)

Examples include,

- the nonnegative orthant
- positive semidefinite cone $\mathbb{S}_+^n$
- nonnegative polynomials on $[0, 1]$

$$K = \left\{x \in \mathbb{R}^n \mid x_0 + x_1 t + x_2 t^2 + \cdots + x_n t^n \geq 0, t \in [0, 1]\right\}$$

A generalized inequality is a relation of the form

$$x \preceq_K y \Leftrightarrow y - x \in K$$

and

$$x \prec_K y \Leftrightarrow y - x \in \text{int}(K)$$

where $\text{int}(K)$ denotes the interior of K.

- In the case of the nonnegative orthant $K = \mathbb{R}_+^n$, we have componentwise inequality $x \preceq y \Leftrightarrow x_i \leq y_i$ for all i.
- In the case of the positive semidefinite cone $K = \mathbb{S}_+^n$, we have $X \preceq Y \Leftrightarrow Y - X \in \mathbb{S}_+^n$.

In these cases, we omit the subscript K and write $x \preceq y$ and $X \preceq Y$.

Minimum and Minimal Elements

As $\preceq$ is not a linear order, we have to define

- $x \in S$ is the minimum element of S with respect to $\preceq$ if $x \preceq y$ for all $y \in S$
- $x \in S$ is the minimal element of S with respect to $\preceq$ if there is no $y \in S$ such that $y \prec x$

1.7.1 Separating Hyperplane Theorem

If C and D are nonempty disjoint convex sets, there exists $a \neq 0, b$ such that

$$a^T x \leq b \text{ for } x \in C, \quad , a^T x \geq b \text{ for } x \in D$$

That is, there exists a hyperplane separating any two convex sets C and D. (Strict separation requires additional assumptions, e.g. C is closed, D is singleton)

Supporting Hyperplane Theorem

A supporting hyperplane to a set C at a boundary point x_0 is

$$\left\{x \mid a^T x = a^T x_0\right\}$$

where $a \neq 0$ and $a^T \leq a^T x_0$ for all $x \in C$.

This can be thought of as a tangent hyperplane.

The theorem states that there exists a supporting hyperplane at any point on the boundary of a convex set.

Dual Cones and Generalized Inequalities

The dual cone of a cone K is

$$K^* = \left\{y \mid y^T x \geq 0 \text{ for all } x \in K\right\}$$

Examples:

- For $K = \mathbb{R}^n_+$, we have $K^* = \mathbb{R}^n_+$
- For $K = \mathbb{S}^n_+$, we have $K^* = \mathbb{S}^n_+$
- For $K = \{(x,t) \mid \|x\|_2 \leq t\}$, we have $K^* = \{(x,t) \mid \|x\|_2 \leq t\}$
- For $K = \{(x,t) \mid \|x\|_1 \leq t\}$, we have $K^* = \{(x,t) \mid \|x\|_\infty \leq t\}$

We say a cone is self-dual if it is its own dual.

Dual cones of proper cones are proper, hence defining generalized inequalities:

$$y \succeq_{K^*} 0 \Leftrightarrow y^T x \geq 0 \text{ for all } x \succeq_K 0$$

We can define minimum and minimal elements using these dual inequalities:

- x is the minimum element of S iff for all $\lambda \succ_{K^*} 0$, x is the unique minimizer of $\lambda^T z$ over S
- x is a minimal element of S if it minimizes $\lambda^T z$ for some $\lambda \succ_{K^*} 0$
- if x is a minimal element of a convex set S, then there exists a nonzero $\lambda \succeq_{K^*} 0$ such that x minimizes $\lambda^T z$ over S

2 Convex Functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\mathrm{dom} f$ is convex and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$

for all $x, y \in \mathrm{dom} f$ and $0 \leq \theta \leq 1$.

- f is concave if $-f$ is convex
- f is strictly convex if the inequality holds strictly
- f is strongly convex if it has minimum positive curvature everywhere

For example, all affine functions and norms on $\mathbb{R}^n$ are convex.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the function $g : \mathbb{R} \to \mathbb{R}$ defined by $g(t) = f(x+tv)$ is convex for any $x \in \mathrm{dom} f$, $v \in \mathbb{R}^n$. That is, a function f is convex if and only if it is convex on every line.

2.1 Extended-value Extension

The extended-value extension $\tilde{f}$ of f is

$$\tilde{f}(x) = \begin{cases} f(x), & x \in \mathrm{dom} f \\ \infty, & x \notin \mathrm{dom} \end{cases}.$$

This simplifies our definition of convexity, the requirement that the domain is convex is no longer needed.

2.2 First-order condition

f is differentiable if $\mathrm{dom} f$ is open and the gradient

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$$

exists at each $x \in \mathrm{dom} f$.

If f is differentiable with convex domain, f is convex iff

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \text{ for all } x, y \in \mathrm{dom} f.$$

(First-order/linear approximation is an underestimate)

2.3 Second-order conditions

f is twice differentiable if dom$f$ is open and the hessian $\nabla^2 f(x) \in \mathbb{S}^n$

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

exists at each $x \in \text{dom} f$ for $i, j = 1, \ldots, n$.

If f is twice differentiable with convex domain, f is convex iff $\nabla^2 f(x) \succeq 0$ for all $x \in \text{dom} f$.

Furthermore, if $\nabla^2 f(x) \succ 0$ then f is strictly convex, while if $\nabla^2 f(x) \succ nI$ for some $n > 0$, then f is strongly convex.

2.4 Epigraph and Sublevel set

The α-sublevel set of $f : \mathbb{R}^n \to \mathbb{R}$ is

$$C_\alpha = \{x \in \text{dom} f \mid f(x) \leq a\}.$$

Sublevel sets of convex functions are convex.

The epigraph of $f : \mathbb{R}^n \to \mathbb{R}$ is

$$\text{epi} f = \left\{(x, t) \in \mathbb{R}^{n+1} \mid f(x) \leq t\right\}.$$

f is convex iff epi$f$ is convex.

2.5 Jensen's Inequality

If f is convex, then

$$f(\mathbb{E}[z]) \leq \mathbb{E}[f(z)]$$

for any random variable z.

When z is a point distribution at x and y, we obtain our convexity condition

$$f(\theta x + (1 - \theta) y) \leq \theta f(x) + (1 - \theta) f(y).$$

2.6 Obtaining Convex Functions

We can show a function f is convex using the definition, by applying some conditions above, or use the following properties:

- αf is convex if f is convex with $\alpha \geq 0$
- if f_1, f_2 are convex, $f_1 + f_2$
- $f(Ax + b)$ is convex if f is convex
- if f_1, f_2 are convex, $\max(f_1, f_2)$ is convex

- if $f(x,y)$ is convex, $\sup_y f(x,y)$ is convex
- if g, h are convex and $\tilde{h}$ is nondecreasing in each argument, $h \circ g$ is convex
- if $f(x,y)$ is convex and C is convex, $\inf_{y \in C} f(x,y)$ is convex
- if f is convex, $g(x,t) = tf(x/t)$ is convex

2.7 The Conjugate Function

The conjugate of a function f is

$$f^*(y) = \sup_x (y^T x - f(x)).$$

The conjugate of a function is always convex.

2.8 Types of Convexity

Quasiconvexity

A function $f : \mathbb{R}^n \to \mathbb{R}$ is quasiconvex if $\text{dom} f$ is convex and the sublevel sets

$$S_\alpha = \{x \in \text{dom} f \mid f(x) \leq a\}$$

are convex for all α.

Alternatively, f satisfies

$$f(\theta x + (1-\theta)y) \leq \max\{f(x), f(y)\}.$$

We define f as quasiconcave if $-f$ is quasiconvex, and f as quasilinear if it's both quasiconvex and quasiconcave.

First order condition: if f is differentiable with convex domain, f is quasiconvex iff

$$f(y) \leq f(x) \Rightarrow \nabla f(x)^T (y - z) \leq 0.$$

Log Convexity

A positive function f is log-concave if $\log f$ is concave:

$$f(\theta x + (1-\theta)y) \geq f(x)^\theta f(y)^{1-\theta}.$$

f is log-convex if $\log f$ is convex.

- if f is twice differentiable then it is log-concave iff

  $$f(x)\nabla^2 f(x) \succeq \nabla f(x) \nabla f(x)^T$$

  for all x.

- the product of log-concave functions is log-concave
- if $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is log-concave, then

$$g(x) = \int f(x, y)\, \mathrm{d}y$$

  is log-concave. (For example, we can apply this to the convolution formula.)

Convexity with respect to Generalized Inequalities

$f : \mathbb{R}^n \to \mathbb{R}^m$ is K-convex if $\mathrm{dom} f$ is convex and

$$f(\theta x + (1-\theta)y) \preceq_K \theta f(x) + (1-\theta) f(y).$$

3 Convex Optimization

Recall: the general optimization problem has the form
minimize $f_0(x)$ subject to $f_i(x) \leq 0$ and $h_i(x) = 0$.

The f_i and h_i are called **explicit constraints**, and introduce **implicit constraints** restricting x to the domain of each f_i, h_i. We say a problem is **unconstrained** if there are no explicit constraints.

A point x is **feasible** if $x \in \mathrm{dom} f_0$ and it satisfies the constraints. A **solution** or **optimal point**, x^* has the smallest value of f_0 among all feasible x.

We may also solve **feasibility problems** where $f_0 = 0$, and our goal is to find any feasible x.

The **optimal value** of the problem is

$$p^* = \inf_{x \in \mathrm{dom} f_0} \{f_0(x) \mid f_i(x) \leq 0, h_i(x) = 0\}.$$

- $p^* = f_0(x^*)$ when x^* exists
- $p^* = \infty$ if the problem is infeasible
- $p^* = -\infty$ if the problem is unbounded below

A point x is **locally optimal** if there is an $R > 0$ such that $z = x$ is optimal with the additional constraint $\|z - x\|_2 \leq R$.

We say an optimization problem is **convex** if f_0, f_i are convex and h_i are affine. If the objective function f_0 is **quasiconvex**, we say the problem is quasiconvex.

Any local optimal point of a convex optimization problem is globally optimal.

3.1 Optimality Criteria

If f_0 is differentiable, then x is optimal iff $\nabla f_0(x)^T (y - x) \geq 0$ for all feasible y.

For example:

- For an unconstrained problem, x is optimal iff $\nabla f_0(x) = 0$
- For an equality constrained problem (i.e. subject to $Ax = b$), x is optimal iff $\nabla f_0(x) + A^T \lambda = 0$ for some λ
- If we want to minimize f_0 subject to $x \succeq 0$, x is optimal iff $\nabla f_0(x)$ is nonnegative in the i-th component when $x_i = 0$ and zero otherwise

3.2 Equivalent Problems

Two problems are equivalent if the solution of one is readily obtained from the solution of the other, and vice versa. Some common transformations that preserve convexity are:

- Eliminating equality constraints, we can replace an equality constraint $Ax = b$ with the solution $x = Fz + x_0$ where x_0 is a particular solution and the columns of F span $\mathcal{N}(A)$. Then, we may write the problem to minimize $f_0(Fz + x_0)$.
- Introducing equality constraints, converse to the above.
- Introducing slack variables for linear inequalities, we can replace an inequality constraint $a^T x \leq b$ with $a^T x + s = b$ and $s \geq 0$.
- Epigraph form, we can replace the objective minimize $f_0(x)$ with minimizing t with constraint $f_0(x) - t \leq 0$.
- Variable reduction, we can replace the problem minimize $f_0(x_1, x_2)$ with minimize $g(x)$ with $g(x) = \inf_{x_2} f_0(x_1, x_2)$.

3.3 Convex representation of Quasiconvex objective function

If f_0 is quasiconvex, there exists a family of functions ϕ_t such that ϕ_t is convex in x for fixed t, and

$$f_0(x) \leq t \Leftrightarrow \phi_t(x) \leq 0.$$

Then, we can solve a convex feasibility problem,

$$\phi_t(x) \leq 0, f_i(x) \leq 0, Ax = b$$

where we can conclude either $t \geq p^*$ (feasible) or $t \leq p^*$ (infeasible).

Chebyshev Center

The Chebyshev center of

$$\mathcal{P} = \left\{x \mid a_i^T x \leq b_i\right\}$$

is the center of the largest inscribed ball

$$\mathcal{B} = \left\{x_c + u \mid \|u\|_2 \leq r\right\}.$$

3.4 Families of Convex optimization problems

Linear Program

A linear program is an optimization problem of the form

$$\text{minimize } c^T x \text{ subject to } Gx \preceq h, Ax = b.$$

- Convex problem with affine objective and constraint functions
- Feasible set is a polyhedron
- There always exists a solution at a vertex

For example, the diet problem of minimizing costs c_j of n food each with nutritional value a_{ij} of nutrient i, subject to at least b_i of nutrient i has the form

$$\text{minimize } c^T x \text{ subject to } Ax \succeq b.$$

Linear-fractional Program

A linear-fractional program is an optimization problem of the form

$$\text{minimize } \frac{c^T x + d}{e^T x + f} \text{ subject to } Gx \preceq h, Ax = b.$$

This is a quasiconvex optimization problem which can be solved by bisection, and equivalent to a linear program

$$\text{minimize } c^T y + dz \text{ subject to } Gy \preceq hz, Ay = bz, e^T y + fz = 1, z \geq 0.$$

A generalized linear-fractional program has the form

$$f_0(x) = \max \frac{c_i^T x + d_i}{e_i^T x + f_i}$$

which is also a quasiconvex optimization problem and can be solved by bisection.

Quadratic Program

A quadratic program is an optimization problem of the form

$$\text{minimize } \frac{1}{2} x^T P x + q^T x + r \text{ subject to } Gx \preceq h, Ax = b.$$

For example, least squares and curve fitting are quadratic programs, and so is a linear program with random cost (minimize $\mathbb{E}[c^T x] + \gamma \mathbb{V}[c^T x]$ where γ is a risk parameter).

Quadratically Constrained Quadratic Program

A quadratically constrained quadratic program is an optimization problem of the form

$$\text{minimize } \frac{1}{2} x^T P_0 x + q_0^T x + r_0 \text{ subject to } \frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0, i = 1, \ldots, m, Ax = b.$$

Second-order Cone Program

A second-order cone program is an optimization problem of the form

$$\text{minimize } f^T x \text{ subject to } \|A_i x + b_i\|_2 \leq c_i^T x + d_i, i = 1, \ldots, m, Fx = g.$$

where $A_i \in \mathbb{R}^{n_i \times n}, F \in \mathbb{R}^{p \times n}$.

- The inequalities are called second-order cone constraints: $(A_i x + b_i, c_i^T x + d_i)$ is a second-order cone on $\mathbb{R}^{n_i+1}$
- When $n_i = 0$, it reduces to LP; if $c_i = 0$ it reduces to QCQP.

Robust Linear Programming

The parameters in optimization problems are often uncertain, there are two common approaches:

- Deterministic model: constraints holds for all a_i in a small neighbourhood around the true value: i.e. the constraints are $a_i^T x \leq b_i$ for all $a_i \in \mathcal{E}_i$.
- Stochastic model: a_i is a random variable, and the constraints must hold with some probability η.

The deterministic model can be converted to a SOCP by defining

$$\mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\}.$$

Then, we can convert the constraint to $\bar{a}_i^T x + \left\|P_i^T x\right\|_2 \leq b_i$.

The stochastic model can be converted to a SOCP by converting the constraint to $\bar{a}_i^T x + \Phi^{-1}(\eta)\|\sigma_i x\|_2 \leq b_i$, where Φ is the standard normal CDF, and σ_i is the standard deviation.

Geometric Program

A **monomial** function is a function of the form

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$$

defined on $\mathbb{R}_{++}^n$.

A **posynomial** function is a sum of monomials.

A geometric program is an optimization problem of the form

$$\text{minimize } f_0(x) \text{ subject to } f_i(x) \leq 1, h_i(x) = 1$$

with f_i posynomial and h_i monomial.

We can convert geometric programs to convex problems by taking the logarithm of the objective and constraints, and then solving the resulting convex problem.

Generalized Inequality Constraints

We can have optimization problems with generalized inequality constraints, for example a constraint of the form $f_i(x) \preceq_K 0$ where K is a proper cone.

Specifically, the conic form problem has affine objective and constraints takes the form

$$\text{minimize } c^T x \text{ subject to } Fx + g \preceq_K 0, Ax = b$$

extends linear programming to nonpolyhedral cones.

Semidefinite Program

A semidefinite program is an optimization problem of the form

$$\text{minimize } c^T x \text{ subject to } x_1F_1 + x_2F_2 + \cdots + x_nF_n \preceq 0, Ax = b.$$

The inequality constraint is called linear matrix inequality (LMI).

Only one LMI constraint is needed, as we can combine smaller matrices M_1, M_2 into a block diagonal matrix $\text{diag}(M_1, M_2)$.

We can convert LP and SOCP to SDP as follows:

- The LP constraint $Ax \preceq b$ is equivalent to $\text{diag}(Ax - b) \preceq 0$.
- The SOCP constraint $\|A_i x + b_i\|_2 \leq c_i^T x + d_i$ is equivalent to

  $$\begin{pmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{pmatrix} \succeq 0$$

  by Schur complement.

Eigenvalue Minimization

The eigenvalue minimization problem asks to minimize the maximum eigenvalue of $A(x) = A_0 + x_1A_1 + \cdots + x_nA_n$.

This is equivalent to the SDP

$$\text{minimize } t \text{ subject to } A(x) \preceq tI.$$

Matrix Norm Minimization

The matrix norm minimization problem asks to minimize $\|A(x)\|_2$ where $A(x) = A_0 + x_1A_1 + \cdots + x_nA_n$.

This is equivalent to the SDP

$$\text{minimize } t \text{ subject to } \begin{pmatrix} tI & A(x) \\ A(x)^T & tI \end{pmatrix} \succeq 0.$$

Vector Optimization

In a vector optimization problem, we say a point x is

- **optimal** if $f_0(x)$ is the minimum value of all feasible x
- **Pareto optimal** if $f_0(x)$ is a minimal value of all feasible x

In particular, we can have multicriterion objective functions of the form $f_0 = (F_1, \ldots, F_q)$. Then,

- x^* is optimal if

$$y \text{ feasible } \Rightarrow f_0(x^*) \preceq f_0(y)$$

- x^{po} is Pareto optimal if

$$y \text{ feasible }, f_0(y) \preceq f_0(x^{\text{po}}) \Rightarrow f_0(x^{\text{po}}) = f_0(y)$$

To find Pareto optimal points, we can convert this to a scalar problem by choosing $\lambda \succ_{K^*} 0$ and writing the objective as $\lambda^T f_0(x)$.

If x is optimal for the scalar problem, then it is Pareto optimal for the vector problem.

We can find almost all Pareto optimal points by varying λ.

In the case of multicriterion problems, λ is a set of weightings for $F_1, \ldots, F_q$.

4 Duality

Every optimization problem has a corresponding dual problem, whose optimal value is a lower bound on the optimal value of the primal problem.

Take a general optimization problem of the form

$$\text{minimize } f_0(x) \text{ subject to } f_i(x) \leq 0, h_i(x) = 0.$$

The Lagrangian is

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

The Lagrange dual function is

$$g(\lambda, \nu) = \inf_{x \in \text{dom} f_0} L(x, \lambda, \nu).$$

Here, g is concave, and $g(\lambda, \nu) \leq f_0(x)$ for all x feasible.

4.1 The Dual Problem

The Lagrange dual problem is to maximize $g(\lambda, \nu)$ subject to $\lambda \succeq 0$.

This finds the best lower bound on p^*, the optimal value of the primal problem.

4.2 Strong and Weak Duality

Weak Duality is the condition that $d^* \leq p^*$, which holds for any dual problem. This can be used to find nontrivial lower bounds to problems.

Strong Duality is the condition that $d^* = p^*$, which (usually) holds for convex problems. There are conditions called **constraint qualifications** that guarantee strong duality on convex problems.

Slater's Constraint Qualification

Strong duality holds for a convex problem if it is strictly feasible, i.e. for the problem

$$\text{minimize } f_0(x) \text{ subject to } f_i(x) \leq 0, Ax = b$$

we require

$$x \in \text{int}\mathcal{D} : f_i(x) < 0, Ax = b.$$

This also guarantees that the dual optimum is attained.

Karush-Kuhn-Tucker Conditions

The KKT conditions are:

1. Primal Feasible: $f_i(x) \leq 0, h_i(x) = 0$,
2. Dual Feasible: $\lambda \succeq 0$,
3. Complementary Slackness: $\lambda_i f_i(x) = 0$,
4. Gradient of Lagrangian vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$$

If x, λ, ν satisfy the KKT conditions for a convex problem, then they must be optimal.

Reformulating Problems and Duality

Sometimes, we can reformulate a problem to make the dual problem easier to solve or more interesting.

For example, the optimization problem to minimize $f_0(Ax + b)$ has a constant dual function. Instead, we can write the problem as

$$\text{minimize } f_0(y) \text{ subject to } y = Ax + b.$$

Problems with Generalized Inequalities

Given a problem of the form

$$\text{minimize } f_0(x) \text{ subject to } f_i(x) \preceq_{K_i} 0, h_i(x) = 0,$$

the Lagrange multiplier $f_i(x) \preceq_{K_i} 0$ is a vector $\lambda_i \in \mathbb{R}^{k_i}$. So, the Lagrangian is

$$L(x, \lambda_1, \ldots, \lambda_m, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i^T f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

and the dual function is defined similarly.

If each $\lambda_i \succeq_{K_i^*} 0$, then $g(\lambda_1, \ldots, \lambda_m, \nu) \leq p^*$.

5 Applications in Machine Learning

5.1 Norm Approximation

The problem to minimize $\|Ax - b\|$ can be interpreted as:

- Geometric: find the closest point in $\mathcal{R}(A)$ to b
- Estimation: if we have a model $y = Ax + v$ where y are measurements, x is unknown and v is noise, we can find the best estimate of x given $y =$
- Optimal design: if x are design variables and Ax is result, we can find the design that best approximates the desired result b

5.2 Penalty Function Approximation

The problem is to minimize $\phi(r_1) + \ldots + \phi(r_n)$ subject to $r = Ax - b$.

There are many choices for ϕ:

- Quadratic ($\phi(u) = u^2$)
- Deadzone-linear ($\phi(u) = \max\{0, |u| - \epsilon\}$) with width ϵ
- Log-barrier ($\phi(u) = -a^2 \log(1 - (u/a)^2)$) defined on $|u| < a$ and is ∞ otherwise.
- Huber Penalty Function:

$$\phi_{\text{hub}}(u) = \begin{cases} u^2, & |u| \leq M \\ M(2|u| - M), & |u| > M \end{cases}$$

  less sensitive to outliers.

5.3 Least-norm Problems

The problem to minimize $\|x\|$ subject to $Ax = b$ can be interpreted as

- Geometric: find the closest point in the affine set $\{x \mid Ax = b\}$ to the origin
- Estimation: $b = Ax$ are measurements of x, find the smallest (most plausible) estimate consistent with measurements
- Design: x are design variables, b are required results, find the smallest (most efficient) design that satisfies the requirements

5.4 Regularized Approximation

The problem to minimize $(\|Ax - b\|, \|x\|)$ can be interpreted as

- Estimation: linear model $y = Ax + v$, with prior knowledge that x is small
- Design: small x is cheaper or more efficient, or $y = Ax$ is only valid for small x
- Robust Approximation: good approximation $Ax \approx b$ is less sensitive to errors in A than approximation with large x

Scalarized Problem

We can parametrize the problem as to minimize $\|Ax - b\| + \gamma\|x\|$. When $\gamma > 0$, the solution traces out the optimal tradeoff curve.

Alternatively in Tikhonov Regularization, we aim to minimize

$$\|Ax - b\|_2^2 + \delta\|x\|_2^2$$

can be solved as a least-squares problem:

$$\text{minimize} \quad \left\| \begin{pmatrix} A \\ \sqrt{\delta} I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2$$

to get solution $x^* = (A^T A + \delta I)^{-1} A^T b$.

5.5 Minimum Volume Ellipsoid Around a Set

The Lowner-John ellipsoid of a set C is the minimum volume ellipsoid $\mathcal{E}$ s.t. $C \subseteq \mathcal{E}$.

- We can parametrize $\mathcal{E}$ as $\mathcal{E} = \{v \mid \|Av + b\|_2 \leq 1\}$ and assume $A \in \mathbb{S}_{++}^n$
- The volume is proportional to $\det A^{-1}$, so we can minimize $\log\det A^{-1}$ subject to $C \subseteq \mathcal{E}$.

This generates a convex problem but evaluating the constraint can be hard.

If C is finite then the constraint is much simpler.

5.6 Maximum Volume Inscribed Ellipsoid

- We parametrize $\mathcal{E} = \{Bu + d \mid \|u\|_2 \leq 1\}$ and assume $B \in \mathbb{S}_{++}^n$
- The volume is proportional to $\det B$, so we can maximize $\log\det B$ subject to $\mathcal{E} \subseteq C$.

This generates a convex problem but evaluating the constraint can be hard.

Once again, if C is finite then the constraint is much simpler.

If $C \subseteq \mathbb{R}^n$ and if we shrink the Lowner-John ellipsoid by a factor of n, it is contained in C.

Similarly, if we expand the maximum volume inscribed ellipsoid by a factor of n, it contains C.

5.7 Centers

- Chebyshev center (center of largest inscribed ball), can be found by LP
- MVE (center of maximum volume inscribed ellipsoid)
- Analytic Center: for a problem

$$f_i(x) \leq 0, i = 1, \ldots, m, Fx = g$$

it is computed as the optimal point of

$$\begin{aligned} \text{minimize} \quad & -\sum_{i=1}^{m} \log(-f_i(x)) \\ \text{subject to} \quad & Fx = g \end{aligned}$$

This is not a property of the set, the same set can have different analytic centers. We can express the inner and outer ellipsoids in terms of the analytic center:

$$\begin{aligned} \mathcal{E}_{\text{inner}} &= \left\{ x \mid (x - x_{\text{ac}})^T \nabla^2 \phi(x_{\text{ac}})(x - x_{\text{ac}}) \leq 1 \right\} \\ \mathcal{E}_{\text{inner}} &= \left\{ x \mid (x - x_{\text{ac}})^T \nabla^2 \phi(x_{\text{ac}})(x - x_{\text{ac}}) \leq m(m-1) \right\} \end{aligned}$$

5.8 Maximum Likelihood Estimation

Suppose we are tasked with estimating a probability density $p_\theta(y)$ of some variable from data $y_1, \ldots, y_m$.

The maximum likelihood principle says we should choose the parameters that result in the highest probability of making these observations, in other words we maximize $\log p_\theta(y)$.

- $l(\theta) = \log p_\theta(y)$ is called the log-likelihood function
- usually decomposes over samples, $l(\theta) = \sum_{i=1}^{m} \log p_\theta(y_i)$
- if p_θ is log-concave then the problem is convex

Say we have a model $y = Ax + v$ where x is unknown, y are measurements and v is noise.

Then, we wish to maximize

$$l(x) = \sum_{i=1}^{m} \log p(y_i - a_i^T x)$$

5.9 Binary Classification Problem

Say we have training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$. We want to learn a density function $\mathbb{P}(y \mid x)$ that estimates the probability of 0 or 1 for an input.

Our classifier will be of the form $f(x) = a^T x + b$ with the sign being taken as the classification.

For example, we can use logistic regression, where we assume y has distribution

$$\mathbb{P}(y = 1 \mid x) = \frac{\exp(a^T x + b)}{1 + \exp(a^T x + b}.$$

The log-likelihood function is

$$\begin{aligned} l(a, b; \mathcal{D}) &= \log \left(\prod_{i:y_i=1} \frac{\exp(a^T x_i + b)}{1 + \exp(a^T x_i + b)} \prod_{i:y_i=0} \frac{1}{\exp(a^T x_i + b)} \right) \\ &= \sum_{i:y_i=1} (a^T x_i + b) - \sum_{i=1}^{m} \log(1 + \exp(a^T x_i + b)) \end{aligned}$$

which is concave in a and b.

Reguarization

To minimize overfitting we use regularization,

$$a^*, b^* = \text{argmax}_{a,b} l(a, b) - \lambda r(a, b)$$

where λ controls regularization strength.

5.10 Support Vector Machines

Given training data $\mathcal{D}$ as before, we wish to find a, b subject to $a^T x_i + b < 0$ for $y_i = -1$ and $a^T x_i + b > 0$ for $y_i = 1$.

We can define the geometric margin of (a, b) with respect to $\mathcal{D}$ to be

$$\gamma_i = y_i \left(\frac{1}{\|a\|} a^T x_i + \frac{b}{\|a\|} \right)$$

and $\gamma = \min \gamma_i$. ($\gamma < 0$ if we are wrong, and $\gamma > 0$ if we're right).

We can aim to write this as an optimization problem

$$\begin{aligned} \text{maximize} \quad & \gamma \\ \text{subject to} \quad & y_i \left(\frac{1}{\|a\|} a^T x_i + \frac{b}{\|a\|} \right) \geq \gamma \end{aligned}$$

but unfortunately this is very hard to solve because of the $\|a\|$ in the denominator.

Instead, we can make the substitution $\gamma = \frac{\gamma'}{\|a\|}$ and so the problem becomes

$$\begin{aligned} \text{maximize} \quad & \gamma' \\ \text{subject to} \quad & y_i \left(a^T x_i + b \right) \geq \gamma' \end{aligned}$$

where we can scale such that $\gamma' = 1$.

Finally, we notice that maximizing $\|a\|^{-1}$ is the same as minimizing $\|a\|^2$. So, we can write the problem as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|a\|^2 \\ \text{subject to} \quad & y_i \left(a^T x_i + b\right) \geq 1 \end{aligned}$$

This is the primal SVM problem and it is a quadratic program.

If we find a^* we can obtain b^* as

$$b^* = -\frac{1}{2}\left(\max_{i:y_i=-1} \langle a^*, x_i \rangle + \min_{i:y_i=1} \langle a^*, x_i \rangle\right).$$

We can introduce Lagrange multipliers α_i for each inequality constraint

$$\mathcal{L}(a, b, \alpha) = \frac{1}{2}\|a\|^2 - \sum_{i=1}^{m} \alpha_i (y_i (a^T x_i + b) - 1).$$

Taking derivatives we have

$$\begin{aligned} \nabla_a \mathcal{L}(a, b, \alpha) = a - \sum_{i=1}^{m} y_i \alpha_i x_i = 0 \Rightarrow a = \sum_{i=1}^{m} y_i \alpha_i x_i \\ \frac{\partial}{\partial b} \mathcal{L}(a, b, \alpha) = \sum_{i=1}^{m} y_i \alpha_i = 0 \Rightarrow \sum_{i=1}^{m} y_i \alpha_i = 0 \end{aligned}$$

We can substitute back to get the dual SVM optimization problem:

$$\text{maximize} \ -\frac{1}{2}\sum_{i,j=1}^{m} y_i y_j \langle x_i, x_j \rangle \alpha_i \alpha_j + \sum_{i=1}^{m} \alpha_i \quad \text{subject to} \quad \alpha_i \geq 0, \sum_{i=1}^{m} y_i \alpha_i = 0$$

Here, we can make predictions using

$$\begin{aligned} f(x) &= a^T x + b \\ &= \left(\sum_{i=1}^{m} \alpha_i y_i x_i\right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y_i \langle x_i, x \rangle + b \end{aligned}$$

The Kernel Trick

Sometimes we wish to learn the classifier with respect to some higher-dimensional features $\phi(x)$.

Let K denote the kernel corresponding to inner products in the higher-dimensional feature space, i.e.

$$K(x, z) = \phi(x)^T \phi(z)$$

Now we can replace $\langle k, z \rangle$ with $K(x, z)$ everywhere.

Regularization

We can add regularization to the SVM if the data contains outliers or is not linearly separable.

We can add slack variables ξ_i to the primal problem to allow for misclassification:

- $\xi_i \geq 0$ and the constraint becomes $y_i(a^T x_i + b) \geq 1 - \xi_i$
- the objective becomes $\frac{1}{2}\|a\|^2 + C\sum_{i=1}^{m} \xi_i$ where C is a regularization parameter
- the dual problem has constraints $0 \leq \alpha_i \leq C$

Sequential-Minimal-Optimization Algorithm

The SMO algorithm operates by starting with a feasible point α, and performing coordinate ascent on the dual SVM objective by

- Picking a pair of training examples (p, q)
- Solving the optimization problem with respect to α_p and α_q while keeping all other α_i fixed.

These updates can be computed very efficiently.

We need to change the constraint such that $\alpha_p y_p + \alpha_q y_q = c$ is constant.

Then, we can write $\alpha_p = y_p(c - \alpha_q y_q)$ and the problem is

$$\text{maximize } a\alpha_q^2 + b\alpha_q + c \quad \text{subject to} \quad L \leq \alpha_q \leq H.$$

Let $a^\dagger = \arg\max_\alpha a\alpha^2 + b\alpha + c$. Then

$$\alpha_q^* = \begin{cases} H, & \alpha^\dagger > H \\ \alpha^\dagger, & L \leq \alpha^\dagger \leq H \\ L, & \alpha^\dagger < L \end{cases}$$

and $\alpha_p^* = y_p(c - \alpha_q^* y_q)$.

6 Unconstrained Minimization

Assume the problem is to minimize $f(x)$, a convex, twice continuously differentiable function with a finite infimum.

We start at a point x_0 such that the sublevel set of x_0 is closed, which is often derived by showing that all sublevel sets are closed:

- If epi$f$ is closed, or
- If dom$f = \mathbb{R}^n$, or
- If $f(x) \to \infty$ as x approaches the boundary of dom$f$.

6.1 Strong Convexity

We can simplify the problem when f is strongly convex on S:

- $f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|y-x\|_2^2$ for all $x, y \in S$, so S is bounded
- $p^* > -\infty$ and for $x \in S$,
$$f(x) - p^* \leq \frac{1}{2\mu}\|\nabla f(x)\|_2^2$$
can be used as a stopping criterion.

6.2 Backtracking Line Search

To find the minima of $f(x + t\delta x)$, we can use an efficient algorithm where we start with $t = 1$ and repeatedly set $t = \beta t$ until

$$f(x + t\delta x) < f(x) + \alpha t \nabla f(x)^T \delta x$$

6.3 Direction of Descent

There are a few options for choosing δx:

- Gradient Descent: $\delta x = -\nabla f(x)$
- Steepest Descent: pick δx to minimize $\nabla f(x)^T \delta x$ subject to $\|\delta x\| = 1$, sometimes unnormalized by multiplying with $\|\nabla f(x)\|_*$
- Newton Step: $\delta x = -\nabla^2 f(x)^{-1} \nabla f(x)$ (minimizes second-order approximation), can be solved by Cholesky factorization

6.4 Stopping Condition

The Newton decrement

$$\lambda(x)^2 = \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$$

is a measure of the proximity of x to x^*.

6.5 Newton's Method

Newton's method uses the Newton step as the direction of descent, and the Newton decrement as a stopping condition.

6.6 Classical Convergence Analysis

Assume f is strongly convex on S with constant μ and $\nabla^2 f$ is Lipschitz continuous on S, meaning

$$\left\|\nabla^2(x) - \nabla^2 f(y)\right\|_2 \leq L\|x - y\|_2.$$

Then, there exist constants $\eta \in (0, \mu^2/L)$ and $\gamma > 0$ such that

- If $\|\nabla f(x)\|_2 \geq \eta$, then $f(x_{k+1}) - f(x_k) \leq -\gamma$
- If $\|\nabla f(x)\|_2 < \eta$, then

$$\frac{L}{2\mu^2}\|\nabla f(x_{k+1})\|_2 \leq \left(\frac{L}{2\mu}\|\nabla f(x_k)\|_2\right)^2.$$

The first phase is called the damped Newton phase, and each step the function value decreases by at least γ. So, it ends after at most $\frac{f(x_0)-p^*}{\gamma}$ iterations.

The second phase, the quadratically convergent phase, is where the function value decreases quadratically and

$$\frac{L}{2\mu^2}\|\nabla f(x_l)\|_2 \leq \frac{1}{2}^{2^l}.$$

So, the number of iterations needed until $f(x) - p^* \leq \varepsilon$ is bounded above by

$$\frac{f(x_0) - p^*}{\gamma} + \log_2 \log_2(\varepsilon_0/\varepsilon)$$

where ε_0 is constant depending on m, L, x_0.

In practice, the second phase is negligible and can be ignored.

7 Equality Constrained Minimization

Assume the problem is to minimize $f(x)$ subject to $Ax = b$, where f is convex and twice continuously differentiable.

Then, x^* is optimal if and only if there exists ν^* such that

$$\begin{aligned} \nabla f(x^*) + A^T\nu^* &= 0 \\ Ax^* &= b \end{aligned}$$

In the quadratic case where $f(x) = \frac{1}{2}x^TPx + q^Tx + r$, we have

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \nu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

which is called the KKT system.

The KKT matrix is nonsingular if and only if

$$Ax = 0, x \neq 0 \Rightarrow x^TPx > 0$$

or equivalently $P + A^TA \succeq 0$.

7.1 Eliminating Equality Constraints

Alternatively, we can rewrite the problem as a problem to minimize $f(Fz+\hat{x})$ with $\hat{x}$ a particular solution and F a basis for the nullspace of A.

7.2 Newton's Method for Equality Constrained Minimization

We can compute the newton step using the KKT system,

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}.$$

We can then compute the Newton decrement as

$$\lambda(x) = (-\nabla f(x)^T \delta x)^{1/2}.$$

Newton's method proceeds as before.

We can also extend this to infeasible points by solving the KKT system as

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta\nu \end{bmatrix} = -\begin{bmatrix} \nabla f(x) + A^T\nu \\ Ax - b \end{bmatrix}.$$

where $r = (f(x) + A^T\nu, Ax - b)$ is the residual.

Then, we can apply Newton's method by first minimizing the residual at each step, then minimizing the objective.

7.3 Solving KKT Systems

For a KKT system of the form

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

We can solve this by:

- LDL^T factorization
- Elimination (if H is nonsingular)

$$AH^{-1}A^T w = h - AH^{-1}g, Hv = -(g + A^T w)$$

- Elimination (if H is singular), we write it as

$$\begin{bmatrix} H + A^T Q A & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g + A^T Q h \\ h \end{bmatrix}$$

  with $Q \succeq 0$ for which $H + A^T Q A \succ 0$, and apply elimination.

7.4 Network Flow Optimization

$$\text{minimize } \sum_{i=1}^{n} \phi_i(x_i) \quad \text{subject to } Ax = b.$$

This is a directed graph with n arcs and $p+1$ nodes, x_i is the flow through arc i, and ϕ_i is the cost function for arc i (with $\phi_i''(x) > 0$).

A is the node-arc incidence matrix (with the last row removed), where $A_{ij} = 1$ if arc i leaves node j, $A_{ij} = -1$ if arc i enters node j, and $A_{ij} = 0$ otherwise. b is the reduced source vector.

Here, the KKT system has H diagonal, so we can solve it by elimination.

7.5 Analytic Center of Linear Matrix Inequality

$$\text{minimize } -\log\det X \quad \text{subject to } \operatorname{tr}(A_i X) = b_i, i = 1, \ldots, p$$

with variable $X \in \mathbb{S}^n$.

The optimality conditions are

$$X^* \succ 0, -(X^*)^{-1} + \sum_{j=1}^{p} \nu_j^* A_i = 0, \operatorname{tr}(A_i X^*) = b_i,$$

and we can derive the Newton equation

$$X^{-1}\delta X X^{-1} + \sum_{j=1}^{p} w_j A_i = X^{-1}, \operatorname{tr}(A_i \delta X) = 0,$$

which follows from the linear approximation $(X + \delta X)^{-1} \approx X^{-1} - X^{-1}\delta X X^{-1}$.

We can solve by

- Eliminating δX in the first equation by $\delta X = X - \sum_{j=1}^{p} w_j X A_j X$
- Substituting δX in the second equation

$$\sum_{j=1}^{p} \operatorname{tr}(A_i X A_j X) w_j = b_i$$

  which is a positive definite linear system with variable $w \in \mathbb{R}^p$.

8 Inner Point Methods

For an inequality constrained minimization problem

$$\text{minimize } f_0(x) \quad \text{subject to } f_i(x) \leq 0, Ax = b, i = 1, \ldots, m,$$

with f_i convex and twice continuously differentiable, $A \in \mathbb{R}^{p \times n}$ with $\text{rank}(A) = p$, p^* is finite and attained, and that the problem is strictly feasible: there exists an x on the interior of the feasible set.

8.1 Log Barrier

We can reformulate inequality constrained problems via an indicator function $I_{\mathbb{R}_-}$ which is 0 if $x \leq 0$ and ∞ otherwise.

However this is non-differentiable, so we can use the log barrier function as an approximation, the problem becomes

$$\text{minimize } f_0(x) - \frac{1}{t}\sum_{i=1}^{m} \log(-f_i(x)) \quad \text{subject to } Ax = b$$

for some $t > 0$. We can increase t to improve the approximation.

The set of solutions can be expressed in terms of t, and the set of solutions $x^*(t)$ is often called the central path.

8.2 Dual Points on Central Path

$x = x^*(t)$ if there exists a w such that

$$t\nabla f_0(x) + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x) + A^T w = 0, Ax = b$$

so $x^*(t)$ minimizes the Lagrangian

$$L(x, \lambda^*(t), \nu^*(t)) = f_0(x) + \sum_{i=1}^{m} \lambda_i^*(t) f_i(x) + \nu^*(t)^T (Ax - b)$$

where we define $\lambda_i^*(t) = 1/(-t f_i(x^*(t)))$ and $v^*(t) = w/t$.

This can also be understood via the KKT conditions,

(a) Primal Constraints: $f_i(x) \leq 0, Ax = b$

(b) Dual Constraints: $\lambda \succeq 0$

(c) Approximate Complementary Slackness: $-\lambda_i f_i(x) = 1/t$

(d) Gradient of Lagrangian Vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \nabla f_i(x) + A^T \nu = 0.$$

8.3 Phase I Methods

First, we find a strictly feasible x (inequalities must be strict).

A basic method is to solve the problem

$$\text{minimize } s \quad \text{subject to } f_i(x) \leq s, Ax = b.$$

If we find $s < 0$, then x is strictly feasible.

Sum of Infeasibilities

We can also solve the problem

$$\text{minimize } \mathbf{1}^T s \quad \text{subject to } s \succeq 0, f_i(x) \leq s_i, Ax = b.$$

8.4 Barrier Method

Given a strictly feasible $x, t := t_0 > 0$, $\mu > 1$ and tolerance $\varepsilon > 0$, repeat:

1. Compute $x^*(t)$ by minimizing $tf_0 + \phi$ subject to $Ax = b$
2. Update $x = x^*(t)$
3. Until $m/t < \varepsilon$
4. Set $t = \mu t$

Convergence

The number of outer (centering) iterations is exactly

$$\left\lceil \frac{\log\left(\frac{m}{\varepsilon t_0}\right)}{\log \mu} \right\rceil$$

plus the initial centering step.

The centering problem to minimize $tf_0(x) + \phi(x)$ is a convex problem, so we can use Newton's method to solve it.

8.5 Generalized Inequalities

Suppose now our inequalities take the form $f_i(x) \preceq_{K_i} 0$ where K_i is a proper cone.

For generalized inequalities, we can define a generalized logarithm $\psi : \mathbb{R}^q \to \mathbb{R}$ over a proper cone $K \subseteq \mathbb{R}^q$ as

- $\text{dom}\psi = \text{int}K$ and $\nabla^2\psi(y) \prec 0$ for $y \succ_K 0$,

- $\psi(sy) = \psi(y) + \theta \log s$ for $y \succ_K 0$ and $s > 0$, where θ is called the degree of ψ

For example, over $\mathbb{S}^n_+$ we have $\psi(Y) = \log\det Y$.

Some properties of this logarithm are:

- $\nabla\psi(y) \succeq_{K^*} 0$
- $y^T\nabla\psi(y) = \theta$

Then, we can define the log barrier for generalized inequalities similarly, as

$$\phi(x) = -\sum_{i=1}^{m} \psi_i(-f_i(x))$$

with ψ_i being the generalized logarithm for K_i. Then we define the central path similarly.

We may run the barrier method similarly, with stopping criterion $\sum \theta_i/t < \varepsilon$, where θ_i is the degree of ψ_i. The number of iterations is then

$$\left\lceil \frac{\log\left((\sum \theta_i)/(\varepsilon t_0)\right)}{\log \mu} \right\rceil.$$

9 Deep Learning

Deep learning involves using machine learning to optimize other machine learning parameters. That is, we wish to minimize $\sum L(f_\theta(x), y)$ for a model f_θ and loss function L. This objective is often nonconvex.

9.1 Multilayer Perceptron

This is a basic feedforward neural network, where we have input $x \in \mathbb{R}^{p_0}$, hidden layers $z_i \in \mathbb{R}^{p_i}$ and output $y \in \mathbb{R}^{p_n}$. The layers compute as

$$z_i = \tilde{f}_i(z_{i-1}) = f_i(A_i z_{i-1} + b_i)$$

where f_i is a non-linear activation function.

The network can then be viewed as a composition

$$y = (\tilde{f}_n \circ \cdots \circ \tilde{f}_2 \circ \tilde{f}_1)(x).$$

Activation Functions

There are a few common activation functions:

- Logistic function: $f(x) = 1/(1 + \exp(-x))$
- Hyperbolic tangent: $f(x) = \tanh(x) = (\exp(2x) - 1)(\exp(2x) + 1)$
- Rectified linear unit (ReLU): $f(x) = \max(0, x)$

Backpropagation

Suppose we wish to compute $\frac{\partial L}{\partial a_i}$ for some a_i, and we have $\frac{\partial L}{\partial z_{i+1}}$. Then we can use the chain rule to compute

$$\frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial z_{i+1}} \frac{\partial z_{i+1}}{\partial a_i}$$

and use the definition $z_{i+1} = f_{i+1}(A_{i+1} z_i + b_{i+1})$ to compute $\frac{\partial z_{i+1}}{\partial a_i}$.

Then, we can propagate gradients backwards through the network.

9.2 Automatic Differentiation

Automatic differentiation is an algorithm to produce code for computing derivatives of a function specified by a computer program. We assume the program is composed of a small set of elementary functions.

There are two ways to do this:

- Forward Mode: We propagate results on the first-order approximation $x + \delta x$ of the input x through the program.
- Reverse Mode: We compute the derivative based on the chain rule while reusing computations where possible.

To compute the gradient in either case, we use numerical approximation methods. Many traditional methods like Newton's method are too expensive, so we often use stochastic gradient descent instead to approximate the gradient on a sample of the data.

9.3 Universal Approximation Theorem

Let f be an activation function and g be any continuous function. Then there exists $A \in \mathbb{R}^{m \times n}$ and $b, c \in \mathbb{R}^m$ such that

$$\hat{g}(x) = c^T f(Ax + b)$$

approximates $g(x)$ everywhere.

However, it turns out that the number of hidden units needed to approximate g grows exponentially with the dimension of x.

9.4 Convolutional Neural Networks

A CNN is a neural network that takes an image ($3 \times W \times H$ tensor) and performs successive layers of

- Convolution, $(x * a)_{st} = \sum_{i,j} x_{s-i,t-j} a_{ij} + b$
- Non-linear transform, e.g. ReLU
- Pooling/downsampling, e.g. $z_{st} = \max \{x_{ij} \mid i, j \in \mathcal{N}_{st}\}$.

The final layers are often fully connected layers, similar to MLPs.

9.5 Recurrent Neural Networks

A recurrent neural network uses the partial output of a network back into itself to give a recurrent network, such as

$$(y_t, h_t) = f(x_t, h_{t-1}).$$

We can train recurrent networks by unrolling the network to a given time and back-propagating through time, such as

$$\begin{aligned}
(y_t, h_t) &= f(x_t, h_{t-1}; \theta) \\
(t_{t-1}, h_{t-1}) &= f(x_{t-1}, h_{t-2}; \theta) \\
&\vdots \\
(y_1, h_1) &= f(x_1, h_0; \theta)
\end{aligned}$$

Then we can again use the chain rule to compute $\frac{\partial L}{\partial \theta}$.

10 Differential Optimization

Take Stackelberg games, where we have a leader and a follower, where the market determines a price based on the combined supply of both players, and each player sequentially chooses some quantity to supply to maximize their profit.

In other words, the leader picks the supply knowing that the follower will pick optimally. If P is the demand function, C_1, C_2 are the cost functions and q_1, q_2 are the quantities supplied, then the leader solves

$$\begin{aligned} &\text{maximize } q_1 P(q_1 + q_2) - C_1(q_1) \\ &\text{subject to } q_2 \in \operatorname{argmax}_q qP(q_1 + q) - C_2(q). \end{aligned}$$

This is a simple example of a bi-level optimisation problem. A general bi-level optimisation problem is

$$\begin{aligned} &\text{minimize } L(x, y; \theta) \\ &\text{subject to } y \in \operatorname{argmin}_u f(x, u; \theta). \end{aligned}$$

To use gradient descent, we need to first compute the gradient of the lower-level solution, then use chain rule.

10.1 Parametrized Optimization

We may have a problem of the form

$$\begin{aligned} &\text{minimize } f(x, u) \\ &\text{subject to } h_i(x, u) = 0 \\ &\qquad\qquad g_j(x, u) \leq 0 \end{aligned}$$

where $x \in \mathbb{R}^n$ are parameters. Assume f is differentiable but not necessarily convex.

10.2 Imperative vs Declarative Nodes

We may think of two types of nodes in a computational graph:

- Imperative nodes gives an explicit input-output relations, like in most of our previous examples
- Declarative nodes have the input-output relationship specified by an optimization problem

For example, we may have a node in the problem of average pooling for finding the centroid of a set of points:

- Implicitly, $y = \frac{1}{n} \sum_{i=1}^n x_i$
- Declaratively, we may have $y = \operatorname{argmin}_u \sum_{i=1}^n \|x_i - u\|^2$

where the declarative specification may be useful if we wish to use a different loss function.

To compute the gradient of the declarative node, there are a few methods.

Unrolling Gradient Descent

We repeat $y_t \leftarrow y_{t-1} - \eta \frac{\partial f}{\partial y}(x, y_{t-1})$ until convergence. And the gradient is

$$\frac{\mathrm{d}y_t}{\mathrm{d}x} = \frac{\partial y_t}{\partial x} + \frac{\partial y_t}{\partial y_{t-1}} \frac{\mathrm{d}y_{t-1}}{\mathrm{d}x} = -\eta \frac{\partial^2 f}{\partial x \partial y}(x, y_{t-1}) + \left(I - \eta \frac{\partial^2 f}{\partial y^2}(x, y_{t-1}) \right) \frac{\mathrm{d}y_{t-1}}{\mathrm{d}x}.$$

Dini's Implicit Function Theorem

Consider the solution mapping associated with the equation $f(x, u) = 0$,

$$Y : x \mapsto \{u \in \mathbb{R}^m \mid f(x, u) = 0\}.$$

We are interested in how elements of $Y(x)$ change as a function of x.

Theorem. Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be differentiable in a neighbourhood of (x, u) and such that $f(x, u) = 0$, and let $\frac{\partial}{\partial u} f(x, u)$ be nonsingular. Then the solution mapping Y has a single-valued localization y around x for u which is differentiable in a neighbourhood $\mathcal{X}$ of x with Jacobian satisfying

$$\frac{\mathrm{d}y(x)}{\mathrm{d}x} = -\left(\frac{\partial f(x, y(x))}{\partial y} \right)^{-1} \frac{\partial f(x, y(x))}{\partial x}$$

for every $x \in \mathcal{X}$.

10.3 Differentiating Unconstrained Optimization Problems

Let $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be twice differentiable and let $y(x) \in \operatorname{argmin}_u f(x, u)$. Then if the Hessian is nonzero,

$$\frac{\mathrm{d}y(x)}{\mathrm{d}x} = -\left(\frac{\partial^2 f}{\partial y^2} \right)^{-1} \frac{\partial^2 f}{\partial x \partial y}.$$

10.4 Differentiating Equality Constrained Optimization Problems

Consider functions $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$. Let $y(x) \in \operatorname{argmin}_u f(x, u)$ subject to $h(x, u) = 0$.

Assume that $y(x)$ exists, f and h are twice differentiable in the neighbourhood of $(x, y(x))$, and that $\operatorname{rank}(\frac{\partial h(x,y)}{\partial y}) = q$. Then for H non-singular

$$\frac{\mathrm{d}y(x)}{\mathrm{d}x} = H^{-1} A^T (A H^{-1} A^T)^{-1} (A H^{-1} B - C) - H^{-1} B$$

where $A = \frac{\partial h(x,y)}{\partial y}$, $B = \frac{\partial^2 f(x,y)}{\partial x \partial y} - \sum_{i=1}^q \nu_i \frac{\partial^2 h_i(x,y)}{\partial x \partial y}$, $C = \frac{\partial h(x,y)}{\partial x}$ and $H = \frac{\partial^2 f(x,y)}{\partial y^2} - \sum_{i=1}^q \nu_i \frac{\partial^2 h_i(x,y)}{\partial y^2}$, and $\nu \in \mathbb{R}^q$ satisfies $\nu^T A = \frac{\partial f(x,y)}{\partial y}$.

10.5 Inequality Constraints

There are two main ways to deal with inequality constraints:

- We may replace inequalities with log-barrier functions
- We may treat them as equality constraints if they are active, and ignore them otherwise

10.6 Applications

10.6.1 Differentiable Least Squares

We need the solution to the least squares problem is $x^* = (A^T A)^{-1} A^T b$. We can compute the gradient of this as

$$\frac{\mathrm{d}x^*}{\mathrm{d}A_{ij}} = \frac{\mathrm{d}}{\mathrm{d}A_{ij}}(A^T A)^{-1} A^T b$$

and we can obtain

$$\left(\frac{\mathrm{d}L}{\mathrm{d}A}\right)^T = wr^T - x^*(Aw)^T$$

where $w^T = v^T(A^T A)^{-1}$ and $r^T = b^T - x^* A^T$.

10.6.2 Differentiable Eigendecomposition

Finding the eigenvector corresponding to the maximum eigenvalue of a real symmetric matrix $X \in \mathbb{R}^{m \times m}$ can be formulated as

$$\begin{aligned} &\text{maximize } u^T X u \\ &\text{subject to } u^T u = 1 \end{aligned}$$

whose optimality conditions are $X_Y = \lambda_{\max} y$ and $y^T y = 1$.

Taking derivatives we get

$$\frac{\mathrm{d}y}{\mathrm{d}X_{ij}} = -\frac{1}{2}(X - \lambda_{\max} I)^{\dagger}(E_{ij} + E_{ji})y.$$

10.6.3 Optimal Transport

One view of optimal transport is as a matching problem: given an $m \times n$ cost matrix M and a $m \times n$ probability matrix P of a match occurring, we wish to

$$\begin{aligned} &\text{minimize } \langle M, P \rangle + \frac{1}{\gamma}\langle P, \log P \rangle \\ &\text{subject to } P\mathbf{1} = r, P^T\mathbf{1} = c \end{aligned}$$

where the second term is an entropic regularization term and r and c satisfy $\mathbf{1}^T r = \mathbf{1}^T c = 1$.

The row and column sum constraints ensure that P is a doubly stochastic matrix (lies within the convex hull of permutation matrices).

The solution takes the form

$$P_{ij} = \alpha_i \beta_j e^{-\gamma M_{ij}}$$

and can be found using the Sinkhorn algorithm:

- Set $K_{ij} = e^{-\gamma M_{ij}}$ and $\alpha, \beta \in \mathbb{R}^n_{++}$,
- Iterate until convergence $\alpha \leftarrow r \oslash K\beta$, $\beta \leftarrow c \oslash K^T\alpha$, where $\oslash$ denotes componentwise division,
- Set $P = \text{diag}(\alpha) K \text{diag}(\beta)$.

For computing the gradient, we can either back-propagate through the Sinkhorn algorithm, or use implicit differentiation with chain rule.