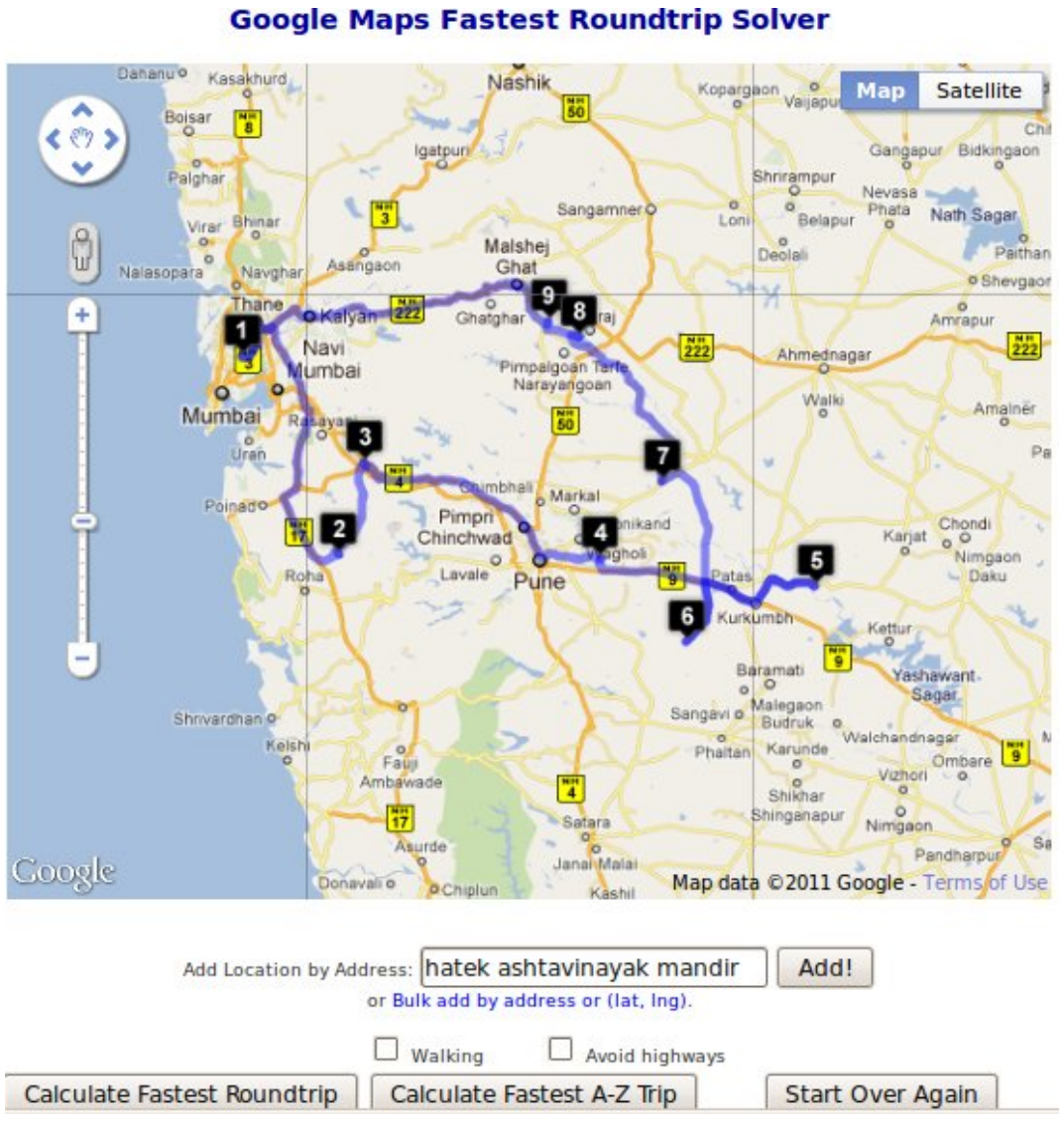
Introductory Lecture (CS709)
Convex Optimization

Instructor: J. Saketha Nath

CSE, IIT-Bombay

22-Jul-2011

Examples ...



http://gebweb.net/optimap/

Examples ...

Course–Room allocation:

- Course registration and room location given
- minimize shifting distance
- room capacity constraints

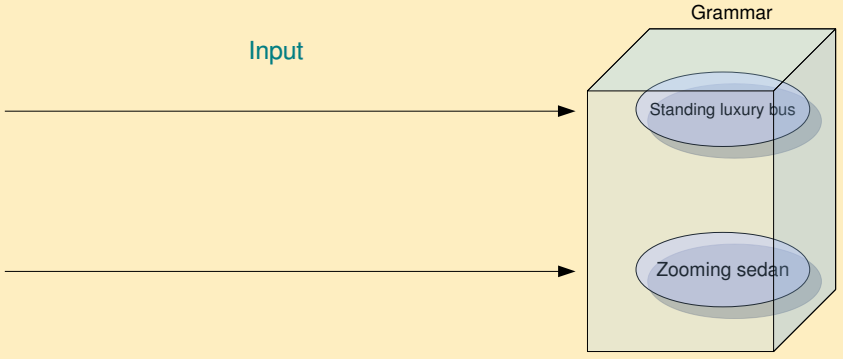
Examples ...

Topology/Material Optimization:

- Topology which min. stress
- Given boundary conditions Show Video

Examples ...

Machine Learning:



$F : I \rightarrow T$

(Overloaded slow moving bus)

(Robots admiring a new car)

Everything is an optimization problem!

— Stephen Boyd

Computational Feasibility is Key!

Review of e.g.:

- Google map prob. — TSP — hard
- Course-room prob. — LP (easy), QAP (hard)
- Topology/mat. opt. — generic (hard), SOCP (easy)
- Machine Learning — generic (hard), SDP (easy)

Characterize *easy* problems?

Characterize *easy* problems?

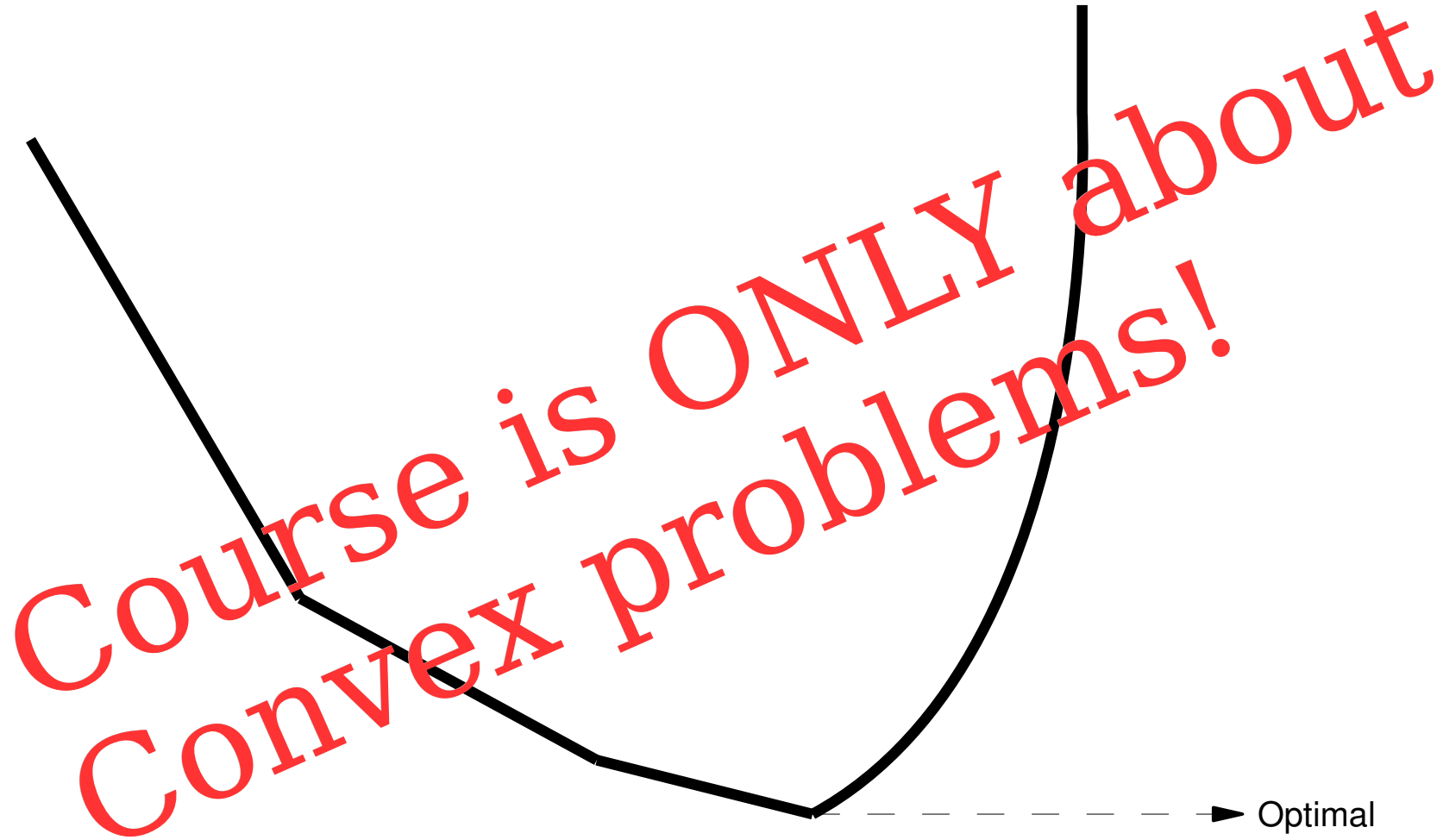
- Un-answered

Characterize *easy* problems?

- Un-answered
- Convex problems are definitely *easy* (appear in real-world)

Characterize *easy* problems?

- Un-answered
- Convex problems are definitely *easy* (appear in real-world)
- Not all non-convex are difficult (Unimodal [Invex], Eigen-Value-Prob)


Course is ONLY about
Convex problems!
Optimal

If easy why study?

- Easy does NOT mean trivial

If easy why study?

- Easy does NOT mean trivial
- Convexity inherent in many real applications

If easy why study?

- Easy does NOT mean trivial
- Convexity inherent in many real applications
- Some can be convexified:
  - Re-casting opt. prob.
  - Assumptions, relaxations

If easy why study?

- Easy does NOT mean trivial
- Convexity inherent in many real applications
- Some can be convexified:
  - Re-casting opt. prob.
  - Assumptions, relaxations
- Knowledge of solvers

Formal Syllabus

I. Theory

- Convex Analysis: Convex Sets, Convex Functions, Calculus of convex functions
- Optimality of Convex Programs: 1st order nec. and suff. conditions, KKT conditions
- Duality: Lagrange and Conic duality

II. Standard Convex Programs and Applications

- Linear and Quadratic Programs
- Conic Programs: QCQPs, SOCPs, SDPs

III. Optimization Techniques

- Smooth Problems: (proj.) Gradient descent, Nesterov's accelerated method, Newton's methods
- Nonsmooth Problems: (proj.) Subgradient descent
- Special topics: Active set and cutting planes methods

Mode of Teaching

- Focus on opt. rather than appl.
- Formal, mathematical development
- Projects help in applying

Evaluation

S.No.	Exam	Weightage	Date
1.	End-Semester	30%	16^{th}-28^{th} Nov'11
2.	Mid-Semester	15%	12^{th}-17^{th} Sep'11
3.	Two Quizes	10+10%	19^{th} Aug'11, 14^{th} Oct'11
4.	Project	20%	15^{th}Oct-14^{th}Nov'11
5.	Bonus, Surprise tests	15%	Anytime

Audit req: 100% attendance

Top Reasons for taking CS709

- I love it
- I like it
- I will use it

Top Reasons for junking CS709

Top Reasons for junking CS709

- Allergic to or afraid of Math

Top Reasons for junking CS709

- Allergic to or afraid of Math
- Want to take A for that B . . . for that CS709

Top Reasons for junking CS709

- Allergic to or afraid of Math
- Want to take A for that B . . . for that CS709
- Friend said u can sleep and still pass

Top Reasons for junking CS709

- Allergic to or afraid of Math
- Want to take A for that B . . . for that CS709
- Friend said u can sleep and still pass
- If u think u are ... (play video)

Convex Optimization (CS709)

Instructor: Saketh

Contents

Lecture 1

- Closer look at an optimization problem
  - Took 3 examples of real-world optimization problems
  - Converted the physical description into a formal "Mathematical Program" (MP) or an "Optimization Problem" (OP). Noted that this conversion is non-unique, usually an "art", and is not the focus in this course.
  - Looked at the key ingredients of a formal MP
    1. Variable space ($\mathcal{X}$) — the domain in which the variable lives, together with the algebraic operations/structures it is endowed with. Primarily resposible for the key results in optimization theory including duality and optimization techniques. The focus in this course is on "finite-dimensional *Hilbert* spaces" (which we will see are *equivalent* to the Euclidean space)
    2. Feasibility/Constraint set ($\mathcal{F}$) — we will study special subsets[1] of vectors, which have nice properties and are easy to deal with. The focus in this course is on MPs with Feasibility set as "convex sets".
    3. Objective Function ($\mathcal{O}$) — the focus is on "convex functions" from $\mathcal{X} \mapsto \mathbb{R}$. We will study some algebraic, topological and calculus properties of convex functions.
  - We formally defined an MP (all combinations with $\min, \inf, \max, \sup$; here P represents the parameters to the MP):

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & \mathcal{O}(x; P) \\ \text{s.t.} \quad & x \in \mathcal{F}(P) \end{aligned} \tag{1.1}$$

[1] For us, subset itself means subset or equal to.

- Identified and defined the related problem[2] (argmin/argmax):

$$\arg\min_{x \in \mathcal{X}} \quad \mathcal{O}(x; P) \qquad \text{s.t.} \quad x \in \mathcal{F}(P) \tag{1.2}$$

- In course of the lectures, we will:
  1. analyze these ingredients (this subject goes with the name "convex analysis")
  2. analyze MPs with convex objective functions and convex Feasibility sets in finite dimensional "Hilbert spaces" (Euclidean spaces for now) — which are called as Convex Programs (CPs). Some of the key questions we will answer are: when is an MP *bounded*, *solvable*? Can we characterize an *optimal solution*? Is it unique? etc.
  3. understand the very important and useful notion of duality which gives ways of arriving at *equivalent* optimization problems for the given problem — this may lead to deep insights into the problem/solution-structure or may lead to efficient solving techniques.
  4. Study standard CPs for which off-the-shelf generic solvers are available.
  5. Study special (scalable?) optimization techniques which work on generic CPs.
- We started revising vector spaces[3]:
  - Given a non-empty set V endowed with two operations $+$ (vector addition: $+ : V \times V \mapsto V$) and $\cdot$ (scalar multiplication: $\cdot \; : \; \mathbb{R} \times V \mapsto V$), if $(V, +)$ form an abelian group, and the operator $\cdot$ is commutative, associative and identity element exists, and the distributive laws governing interaction of $+$ and $\cdot$ hold, then the triplet $\mathcal{V} = (V, +, \cdot)$ is called a vector space and elements of V are called as vectors.
  - We gave a lot of examples of vector spaces — those with matrices, polynomials, functions etc.
  - We identified *linear combination* as an important operation (V is closed under lin. comb. by axioms).
  - We outlined results about basis etc. We will discuss more in the next lecture.

[2] Please revise notions of maximum, minimum, GLB(infimum), LUB(supremum) and their existance results, atleast for sets of real numbers. `http://en.wikipedia.org/wiki/Supremum` should be enough.

[3] Go through pages 1–13 in [Sheldon Axler, 1997]. Also go through related exercises.

Lecture 2

- After a short recap, we continued with our question does every set of vectors V, have a subset of vectors, say B, such that linear span of B, LIN(B), i.e., the set of all vectors which can be expressed as linear combinations of those in B, is equal to V? Obviously such sets exist (for example take $B = V$ itself). Such sets are called as spanning sets.
- A vector space is finite-dimensional if there exists a spanning set of finite size.
- Given any $v \in V$ and B a spanning set of V, v can be written as lin. comb. of vectors in B. We observed that if v is represented using a Euclidean vector with components as the coeff. of lin. comb., then we get a compact representation of vectors. Not only this, lin. comb. of vectors in V can be got by simply applying the same lin. comb. to the corresponding Euclidean vector representations.
- We said that it will be great if i) the spanning set is small (smallest). (Then the proposed representation will be highly compact) ii) the proposed representation is one-to-one.
- We proved[1] that answer to both goals is the same: a Basis, which is a linearly independent, spanning set. A linearly independent set is a set of vectors whose non-trivial (not all zero) lin. comb. can never give a trivial vector (zero vector). The common size of any basis is called the dimensionality of the vector space.
- Hence a basis is like a pair of goggles, through which the **vector space** looks "simple". The key result we showed is that every finite dimensional vector space has a basis and hence is essentially as simple as an Euclidean vector space.

[1] Refer pages 21-36 in [Sheldon Axler, 1997].

- We noted that there may be subsets of a set of vectors which themselves form a vector space with the corresponding $+$ and $\cdot$ — such a subset is called a linear set or linear variety and the resulting vector space is called a subspace of the original vector space. In lectures, we may interchangebly use the terms subspace and linear set (as long as it does'nt create much confusion).

- We studied some examples of subspaces in various vector spaces and noted their basis.

- A basis gives an inner/constitutional/compositional/primal description (a description of an object with help of parts in it) of the vector space it spans.

- Euclidean vector spaces are interesting not only because of lin. comb., but also because notions of dot-products, distances, projections and other such interesting operations exist. In order to make abstract vector spaces interesting, we added a new operator $<>: V \times V \mapsto \mathbb{R}$, called the inner-product, which satisfies positive-definiteness, symmetry and linearity properties and extends the idea of a dot-product in Euclidean spaces[2]. A vector space endowed with a valid inner-product is called an inner-product space.

- We gave many examples[3] of inner-products with Euclidean vectors, Matrices and polynomials.

- Encouraged by the results that a "basis" is the right goggles for the given vector space, we asked the question is a "basis" the right one for an inner-product space too? The answer is negative. However, we showed that if the basis is a special one: orthonormal basis, where every pair of vectors in the basis is orthogonal (i.e., inner-product is zero) and each vector in basis is of unit length. (Length/norm of a vector is defined as the square-root of the inner-product of the vector with itself). Then, the inner-product between two vectors can be computed by simply taking dot-product of the corresponding Euclidean representations. In other words, an orthonormal basis is like a pair of goggles, through which an inner-product space looks "simple".

[2] Refer pg 98-101 of [Sheldon Axler, 1997] for definition and examples.

[3] We discovered the positive-definite (pd) matrices (if M is pd, we denote it by $M \succ 0$) naturally while giving examples of inner-products. Refer http://en.wikipedia.org/wiki/Positive-definite_matrix. It is helpful if one is familiar with all results pertaining to pd matrices, especially the one about its eigen value decomposition: $M \succ 0 \Leftrightarrow M = L\Lambda L^\top$, where L is an orthonormal matrix and Λ is a diagonal matrix with positive entries. The entries in the diagonal matrix are called eigen-values and columns in the orthonormal matrix are called as eigen-vectors.

- Gram-Schmidt algorithm gives a way of getting an orthonormal basis from a basis[4]. As a result, every finite dimensional vector space has an orthonormal basis. Hence every finite dimensional inner-product space is essentially equivalent to the Euclidean inner-product space (with dot product as the inner product).

- Once the notion of inner-product exists, one can define the notions of, length/norm[5] of a vector, angle between vectors (cosine of angle between vectors u, v is defined as $\frac{\langle u,v \rangle}{||u||||v||}$) and projections[6] of vectors onto vectors or onto subspaces/subsets. Projection of a vector onto a subset is the vector in the subset which is closest to the vector in that inner-product space.

- In the next lecture we will see how inner-products can be used to give an outer/dual/*Veda* view of a subspace (and later on for different special subsets).

[4] Refer pg 108, 109 for this algorithm.

[5] Refer pg. 102-106 for definition and examples

[6] Refer http://en.wikipedia.org/wiki/Projection_%28linear_algebra%29

Lecture 3

- We began with illustrating (through examples) what the key benefits of the abstract study of vector/inner-product spaces, we did in couple of previous lectures, are from an optimization perspective:

  1. The optimization theory we are to study is generic and applies to problems with varied variable spaces and geometries.
  2. Alternatively, through orthonormal bases, one can reduce these varied problems to those in Euclidean space and work.

- We completed our discussion on (finite dimensional) inner-product spaces:

  - We explained the significance of a kernel in an inner-product: $\langle x, y\rangle_M = x^\top M y$ (for Euclidean vectors), $\langle f, g\rangle = \int f(x)g(x)w(x)\,dx$ (for functions/polynomials). M or w is called the kernel and determines the geometry. Usually the application decides what the kernel is.
  - Orthogonal complement of S is $S^\perp = \{x \mid \langle x, v\rangle = 0 \ \forall\ v \in S\}$. It is easy to show that $S^\perp$ is always a subspace.
  - With orthonormal basis it is easy to figure out the coefficients of linear combinations ... i.e., if $\{v_1, \ldots, v_n\}$ is an orthonormal basis, then for any vector x we have: $x = \sum_{i=1}^n \langle x, v_i\rangle v_i$.
  - We noted that there might be norms like $\|x\|_1 = \sum_i |x_i|$, which are not induced from inner-products. We can talk about vector spaces endowed with these norms – called as normed vector spaces or metric spaces (ofcourse every inner-product space is a normed vector space with the norm as that induced by the inner-product). However they are not attractive for us ... as inner-product gives notions of angles, projections, etc. which are fundamental to optimization algorithms. Moreover, the inner-products (as we shall see) are the key to duality theory.

- In any inner-product space we can define convergence of a sequence of vectors: Let $\{x_n\}$ be a sequence of vectors. If for any given $\epsilon > 0, \exists N \ni \forall n \geq N$, we have: $\|x - x_n\| < \epsilon$, then the sequence is said to converge to x i.e., $\{x_n\} \to x$. x is called the limit of the sequence $\{x_n\}$.
- We know that Euclidean spaces have no gaps (they are complete spaces) i.e., every Cauchy sequence converges. Because of our equivalence, all finite dim. inner-product spaces are also complete and hence qualify to be called as Hilbert spaces[1] (complete normed vector spaces are called as Banach spaces).
- We defined closed (or complete) sets and gave examples: A subset of vectors S is said to be closed iff every convergent sequence in S has its limit in S.
- We defined open sets and gave examples: A subset of vectors S is said to be open iff for all $v \in S$, $\exists r > 0 \ni B_r(x) = \{y \mid \|y - x\| \leq r\} \subset S$. We also gave examples of sets which are neither open nor closed and noted that the trivial vector space (containing only zero) and the full vector space are the only two subsets which are clopen i.e., both closed and open.
- A subset S is said to be bounded iff a ball of finite radius contains it. We gave examples. Recalled Bolzano-Weierstrass theorem[2].
- We defined compact sets as those which are closed and bounded.
- Given two inner-product spaces $\mathcal{V}_1 = (V_1, +_1, \cdot_1, \langle\rangle_1)$ and $\mathcal{V}_2 = (V_2, +_2, \cdot_2, \langle\rangle_2)$, we defined the direct sum of those, $\mathcal{V} = \mathcal{V}_1 \oplus \mathcal{V}_2$, which is another inner-product space defined as $\mathcal{V} = (V, +, \cdot, \langle\rangle)$, where $V = \{(v_1, v_2) \mid v_1 \in V_1, v_2 \in V_2\}$. Given two vectors $v = (v_1, v_2), w = (w_1, w_2) \in V$, we have: $v + w = (v_1 +_1 w_1, v_2 +_2 w_2)$, $\alpha \cdot v = (\alpha \cdot_1 v_1, \alpha \cdot_2 v_2)$ and $\langle v, w\rangle = \langle v_1, w_1\rangle_1 + \langle v_2, w_2\rangle_2$. This is the natural way of stacking up arbitrary spaces to form big space.

- We began discussing the second ingredient of an MP, which is the Feasibility set, which (for us) is some subset of some finite dimensional inner-product space ($\mathbb{R}^N$). We defined some set operations like (arbitrary) union[3] and intersection[4], sum and difference of sets — for two sets A, B, their sum $A + B = \{z = x + y \mid x \in A, y \in B\}$ and their difference $A - B = \{z = x - y \mid x \in A, y \in B\}$. We begin with a subspace, which is a special subset.

[1] Refer http://en.wikipedia.org/wiki/Hilbert_space.
[2] Refer http://en.wikipedia.org/wiki/Bolzano%E2%80%93Weierstrass_theorem
[3] Refer http://en.wikipedia.org/wiki/Union_%28set_theory%29
[4] Refer http://en.wikipedia.org/wiki/Intersection_%28set_theory%29

- For a subspace S, a basis B gives a compositional/inner/primal view. A view made of parts inside an object. If dim. of S is n, then B has n entries. So this description requires n units (of space/memory etc.). Also this description employs lin. comb. as the fundamental operation.
- Whereas, one can also describe a subspace as that which is orthogonal to its orthogonal complement. More specifically, suppose we consider a matrix $M^\top$ whose columns are a basis for $S^\perp$. Then a dual/outer description of S is: the set of solutions to the homogeneous system of equations given by $Mx = 0$. This description does not involve vectors in the subspace of discussion. Also, this description employs inner products as the fundamental operation.
- If the entire vector space is of dim. N, then this description requires $N - n$ units (which is also the rank of the matrix M). Whenever $n < N/2$, the primal description seems efficient and whenever $n > N/2$, the dual description seems efficient.
- It is also easy to see the set of solutions of $Ax = 0$ always form a subspace (for any A). Hence solution set of homogeneous equations is a characterization for subspace.
- In the extreme case, where $n = N - 1$, the dual description looks like $a^\top x = 0$ (here, a is a vector), which is exactly the equation of a hyperplane through the origin (which we are familiar with from school days).
- Easy to show every subspace is a closed set. The trivial and full subspaces are also open.
- We can also show intersection of subspaces is a subspace. There are two approaches for proving this — the primal approach (which involves primal descriptions in proofs) and the dual approach (involves dual descriptions). This pattern repeats for many proofs which we study in this course. Below we outline both approaches for: given subspaces S_1, S_2 show that $S = S_1 \cap S_2$ is a subspace.
  - Primal: Strategy is to show S is closed under lin.comb. Let $v, w \in S \Rightarrow v, w \in S_1; v, w \in S_2$. Since S_1 is subspace, a (any) lin.comb. of v,w is in S_1 and since S_2 is also a subspace this lin.comb. also is in S_2 and hence the (any) lin.comb. must be in S.
  - Dual: S_1 is a subspace hence is the solution-set of $A_1 x = 0$ (for some A_1). Similarly, S_2 is a subspace and hence is the solution-set of $A_2 x = 0$ (for some A_2). The intersection set S is nothing but the solution set of $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x = 0$, which is itself a homogeneous set

of equalities and hence S must be a subspace.

  - Easy counter-examples show union of two subspaces is not a subspace.

- We outlined the key intuitions/definitions behind another special class of sets known as affine sets, which we will study in next the lecture.

- Mandatory book readings:

  - Appendix sections A.1, A.2, A.4 and A.7 from Nemirovski [2005].

Lecture 4

[A short quiz was conducted at the beginning of the lecture. A general note about the formalism in the proofs u write is made. Proofs written in exams will be evaluated if they are formal.]

We began with the discussion on Affine sets[1]

- Based on the intuition that affine sets are shifted versions of linear sets, a set A is defined as an affine set iff it can be expressed as $\{a_0\} + L$, where $a_0 \in V$ and L is some linear set.

- We showed that $L = A - A$ and hence is determined uniquely (given A). However a_0 (above) could be replaced with any $a \in A$.

- From this characterization of L, it is clear that any vector $x \in A$ can be *uniquely* written as $a_0 + \lambda_1 v_1 + \ldots + \lambda_n v_n$ where $\{v_1, \ldots, v_n\}$ is a basis for L and $\lambda_i \in \mathbb{R} \forall i$. The emphasis is on the point that given x, λs are fixed.

- Once the v_is are replaced with vectors in A, we will get an inner/primal description. By definition of A, $\exists a_i \in A \ni a_i = a_0 + v_i (\forall\ i)$. Hence, $x = (1 - \sum_{i=1}^{n} \lambda_i) a_0 + \lambda_1 a_1 + \ldots + \lambda_n a_n$. Re-writing, $\rho_0 = (1 - \sum_{i=1}^{n} \lambda_i)$, $\rho_1 = \lambda_1, \ldots, \rho_n = \lambda_n$, we have that $x = \sum_{i=0}^{n} \rho_i a_i$, $\sum_{i=0}^{n} \rho_i = 1$ and importantly, ρs are fixed given x. Since x was arbitrary we can say affine sets are closed under affine comb.[2] i.e., linear comb. with coeff. summing to unity. Conversely, set closed under affine combinations, i.e., set which is equal to its affine-hull, which is the set of all affine comb. with vectors in it, is also an affine set. Hence affine sets exactly those which are closed under affine comb.

[1] Usually $\mathcal{V} = (V, +, \cdot, \langle\rangle)$ represents our inner-product space.

[2] We could have also started with this as our definition for affine sets or a definition like affine sets are those closed under affine comb. with any two pair of vectors i.e., sets having lines. Realize that everything is the same.

- Any subset of an affine set whose affine hull is equal to the affine set itself is called an affinely spanning set. Sets in which two different affine combinations do not give the same vector are called as affinely-independent sets (representation under them is unique, mathematically the condition turns out to be non-trivial lin. comb. of vectors with sums of coeff. in lin.comb. being zero cannot be a trivial vector). Above discussion also shows that for any affine set there exists an affinely spanning, affinely-independent set, called the affine-basis, with which every vector in an affine set can be represented uniquely. Infact we showed how to build it.

- By above, the vector $x = \sum_{i=0}^{n} \rho_i a_i,\ \sum_{i=0}^{n} \rho_i = 1$ can be represented as an n+1 dim vector $\begin{bmatrix} \rho_0 \\ \vdots \\ \rho_n \end{bmatrix}$ with restriction that $\sum_{i=0}^{n} \rho_i = 1$. This representation is called barycentric co-ordinate rep. With this, it is easy to see that any affine set in n-dim space is equivalent to a hyperplane in $n+1$ dimensions (and the equation of the hyperplane is $\mathbf{1}^\top \mathbf{x} = 1$).

- The dual characterization was also simple and like primal case, it was got by looking at dual description of L instead: consider a matrix $M^\top$ whose columns form a basis for the orthogonal complement of L. Then the solution set of $M(x - a_0) = 0$ is A and moreover solution-set of any non-homogeneous finite set of consistent equalities is also an affine set. In case M has single row, i.e., L is of dim. $n-1$, then we get a hyperplane: $m^\top x = b$.

- We gave examples of affine sets with matrices and identified $\langle M, X \rangle_F = b$ as the hyperplane expression.

- Arbitrary intersection of affine sets is affine; whereas union of affine sets may not be affine.

- One can also talk about half-spaces associated with a hyperplane: $\mathcal{H}^+ = \{x \mid m^\top x \geq b\}$ is called the positive halfspace and $\mathcal{H}^- = \{x \mid m^\top x \leq b\}$ is called the negative halfspace. Note that half-spaces are neither linear sets nor affine sets. We next study subset which are formed by intersections of half-spaces formed by hyperplanes through origin – called as cones (or conic sets).

We gave examples of cones (in 2d, to begin with) which motivated its defn.:

- Given a set of vectors $W = \{v_1, \ldots, v_n\}$, their conic comb. is defined as $\lambda_1 v_1 + \ldots + \lambda_n v_n$ for some $\lambda_i \geq 0$.

- CONIC-HULL(W) is defined as the set $\{\lambda_1 v_1 + \ldots + \lambda_n v_n \mid \lambda_i \geq 0 \ \forall \ i\}$.
- A set C is called a cone or a conic set iff C=CONIC-HULL(C).
- We then looked at more eg. of cones including the ones in 3d like the ice-cream cone.
- We identified two kinds of cones: i) polyhedral cones, where there is a finite subset whose conic comb. gives the cone ii) non-polyhedral cones. So by defn., inner description is simple for polyhedral cones.
- We gave outer/dual description for polyhedral cones: solutions to $Ax \leq 0$ (i.e., intersection of half-spaces).
- We noted that for some cones, a simple norm-based locus description is most efficient. For eg., a (symmetric) ice-cream cone (in 3d) is described by $\sqrt{x^2 + y^2} \leq z$. Ice-cream cones with diamond, square and elliptical cross-sections can be described[3] by $\|v\|_1 \leq z$, $\|v\|_\infty \leq z$ and $\|v\|_M \leq z$ respectively; here $v = \begin{bmatrix} x \\ y \end{bmatrix}$.
- Examples also showed that with every cone, C, there is an associated cone, called the dual cone, C^*, which is defined as $C^* = \{x \mid \langle x, v \rangle \geq 0 \ \forall \ v \in C\}$. It is easy to show that it is a cone.
- We visualized dual cones for some examples. In the next class we will continue with cones.
- Mandatory Reading: Section A.3 from Nemirovski [2005]

[3] For definitions of 1,∞ norms and others refer http://en.wikipedia.org/wiki/Norm_%28mathematics%29. Also, $\|v\|_M = \sqrt{v^\top M v}$ where M is a pd matrix (also known as kernel).

Lecture 5

- We briefly reviewed conic sets:
  - The defn. and primal/inner description — this lead to a natural classification of cones. Polyhedral cones are the ones which can be obtained by conic combinations of finite vectors.
  - We informally argued that outer/dual description of cone is (arbitrary) intersection of halfspaces: $a_i^\top x \leq b_i, i \in I$ (I could be an uncountable index set). Also for polyhedral cones, I can be chosen finite. These results regarding duality were not provided; however we gave an intuition how a fundamental theorem known as *separation theorem* can help in proving this[1].
  - We gave examples of cones where primal description is more efficient than dual and vice-versa. Basically, whenever the cone is *flat*, then primal must be efficient.
  - We also briefly discussed open cones: intersection of open halfspaces or equivalently positive linear combinations of vectors. In this context we defined closure of a set M is the set M together with all limits of convergent sequences or equivalently, the smallest closed set containing M.
  - We gave a (incomplete, one-way) proof of the following: If C is a cone such that $V = \{v_1, \ldots, v_n\}$ gives its primal description (i.e., conic hull of V is C) and the set $A = \{a_1, \ldots, a_m\}$ gives its dual description (i.e., C is intersection of halfspaces $a_1^\top x \leq 0$), then the primal and dual descriptions of the dual cone are given by $-A$ and $-V$ respectively[2]. Hence if primal view is efficient for the primal cone then the dual view

[1] We will prove separation theorem after introducing convex sets, as it holds for the more generic case of convex sets.

[2] Here $-A$ is the set which has elements of A with negative sign. The minus sign is just appearing because of the sign conventions we chose for dual description and dual cone which are opposing.

will be efficient for the dual cone and vice-versa. Proof for $-V$ being the dual description of the dual cone was easy and the other part of the proof was left to the students to think over.

- We gave more examples of cones: ones containing lines[3], cones which are flat, dual cones of various cones, cones in matrix spaces like the polyhedral cone of all non-negative diagonal matrices, cone of all psd matrices[4] of a given size etc. We gave primal and dual descriptions for (most of) these examples. Infact, we realized that the defn. of psd is nothing but the dual description[5].
- Cones which are equal to their dual cone are called as self-dual cones. For eg. the cones $||x||_2 \leq y$, the psd cone etc.
- As with other sets we studied, (arbitrary) intersection of cones is a cone; whereas union of cones may not be a cone.

- All the sets studied till now are necessarily unbounded. We will now study sets which can be bounded. Like how affine sets must have lines, cones must have rays, convex sets are those which must have line segments.

- A set C is convex iff $x, y \in C \Rightarrow [x, y] \in C$. Here $[x, y]$ denotes the line segment between x, y i.e., $[x, y] = \{\lambda x + (1 - \lambda)y \mid 0 \leq \lambda \leq 1\}$.

- By induction, it is easy to show that convex sets are closed under convex combinations: linear combinations with weights non-negative and summing to unity. In other words, the convex-hull[6] of a convex set is itself.

- After giving some examples, we asked the question what can be the inner/primal description. This again lead to a natural classification of convex sets: those which are convex hulls of finite sets — called as Polytopes. Eg. simplex, square, tetrahedron, all polygons. Also, circle is an eg. of a convex set which is not a polytope.

- We realized that intersections of halfspaces of hyperplanes, not necessarily passing through the origin, are convex sets. Infact, intuitively we argued any convex set can be realized as (arbitrary) intersection of halfspaces — which gives the dual description. We also argued that for polytopes the dual description is finite. Again proving these will require separation theorem.

[3] Cones which do not contain any line are called as Pointed cones.

[4] A matrix M is psd iff i) M is symmetric ii) $x^\top Mx \geq 0, \forall\ x$. The set of all pd matrices form an open cone. The closure of set of all pd matrices is nothing but the set of all psd matrices.

[5] What will be the primal description of the set of all psd matrices?

[6] Given a set S, the convex-hull of it (denoted by conv(S)) is defined as the set of all possible convex combinations of the vectors in S.

- We realized there are convex sets like wedges (formed by intersection of two lines/planes in 2/3-d spaces) which are not polytopes, but dual description is finite. Convex sets where dual description is finite are called as polyhedral sets[7].
- Extending the defn. of dual cones which are cones induced from sets, we defined polar sets which give convex sets starting from a set: Given a set C, the polar set of C, denoted by C^*, is $\{x \mid < x, c > \leq 1 \ \forall \ c \in C\}$. We gave an example which lead to the famous Rangoli star pattern :)
- We will continue our study of convex sets in the coming lecture.
- Mandatory reading: appendix sections B.1.1-B.1.4 in Nemirovski [2005], sections 2.1,2.2 and relevant parts of 2.6 in Boyd and Vandenberghe [2004].

[7] By defn., all polytopes are polyhedra. Pritish Uday realized this classification purely based on primal view: polytopes have finite, polyhedra have countably infinite primal descriptions. I think this is true. Thanks to him for pointing this out.

Lecture 6

- We gave many examples of polytopes, polyhedra and generic convex sets in Euclidean as well as matrix spaces. eg. Simplex[1], Birkhoff polytope; set of stochastic matrices; normed balls and conic sections.

- We focussed on normed-balls (centered at origin i.e., $\{x \mid \|x\| \leq 1\}$, where $\|\cdot\|$ is some norm and need NOT be the inner-product induced one) and looked at their polar sets. We observed that the definition of polar set in this case can be equivalently written as: $\{x \mid f(x) \leq 1\}$ where $f(x) = \sup_{\|y\| \leq 1} \langle x, y \rangle$ (again, here $\|\cdot\|$ need NOT be the norm induced by $\langle \cdot, \cdot \rangle$). It is easy to show that f(x) defines a norm over x and is called the dual norm (denoted by $\|\cdot\|_*$) of the norm we began with.

- We gave examples of some dual norms. Infact we commented that in case of norms with Euclidean vectors for a p-norm (i.e., $\|x\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$; here $p \geq 0$ and with $p = \infty$ we have $\|x\|_\infty = \max_i |x_i|$), the dual norm is a q-norm, where $1/p + 1/q = 1$. This comes from Holder's inequality[2]. From this we have that 2-norm (the Euclidean norm) is self-dual.

- We also argued that $\|\cdot\|_{M^{-1}}$ is the dual norm of $\|\cdot\|_M$[3]. Hence we can call inverse of a matrix as the dual matrix. Again we got a fundamental Rangoli pattern with this :)

- We defined dimension of a convex/conic set as that of the affine hull of it. We noted that in many cases it might help to view the set restricted to its affine-hull.

[1] Simplex in n-dim space is the convex hull of $n + 1$ affinely independent points.

[2] Refer `http://en.wikipedia.org/wiki/H%C3%B6lder%27s_inequality`. Note that this inequality is very generic and hence can characterize dual norms in many Hilbert spaces apart from Euclidean, which we mentioned above.

[3] Bonus marks (max.5) will be awarded to the first student who communicates a proof of this to me.

- After some motivation, we defined terms like interior point, interior, boundary (refer section B.1.6.B in Nemirovski [2005]); relatively interior point, relative-interior and relative-boundary (refer section B.1.6.C in Nemirovski [2005]). A useful observation is that a (non-empty) convex set always has a non-empty rel.int.[4]. This is essentially because all (non-empty) convex sets must have appropriate dimensional simplices in them and thats what gives them interior (volume). Infact, if we change our model for object having volume from a sphere to a simplex, then it is trivial to prove this.

- We noted statements for Caratheodary, Radon and Helly theorems (sections B.2.1–B.2.3 in Nemirovski [2005]).

- We then introduced the notion of (linear) separability: Two (non-empty) sets S_1 and S_2 are separable iff there exists a vector a such that $\sup_{x \in S_1} \langle a, x \rangle \leq \inf_{y \in S_2} \langle a, y \rangle$ and $\inf_{x \in S_1} \langle a, x \rangle < \sup_{y \in S_2} \langle a, y \rangle$. The vector a is said to separate the two sets. In this case one can always compute an appropriate number b such that $S_1 (S_2)$ lies in the negative(positive) half-space of the hyperplane $a^\top x - b = 0$, which is called the separating hyperplane.

- Our idea was to prove the separation theorem from the Projection theorem, which we will detail in the coming lecture.

- Mandatory reading: appendix sections B.1, B.2.1–B.2.3, B.2.6 and B.2.8 in Nemirovski [2005], entire chapter 2 (except for generalized inequalities) in Boyd and Vandenberghe [2004]

- Optionally read sections 1, 2, 3, 14, 17 and 19 in Rockafellar [1996]

[4] Refer theorem B.1.1 in Nemirovski [2005] for a proof.

Lecture 7

- We showed that a non-empty closed convex set C and a point $v \notin C$ are separable using the projection theorem[1]:

  Theorem 7.0.1. *If C is a non-empty closed convex set and $v \notin C$, then:*

  1. *Projection of v onto C i.e., $P_C(v) = \text{argmin}_{y \in C} \|v - y\|$ exists and is unique.*
  2. *$\langle v - P_C(v), x - P_C(v) \rangle \leq 0 \ \forall \ x \in C$.*
  3. *$P_C(v)$ must be a point on relative boundary of C.*

- Separation of v and C is clear from result 2 in theorem above. Infact, we showed that a separating hyperplane passing through $P_C(v)$, a point on the rel.bound., can be drawn. Note that this hyperplane also separates $P_C(v)$ from C. Such a hyperplane, which separates a point on the rel. bound. and the set itself, is called as a supporting hyperplane of the set at that point.

- It was then easy to argue that every closed convex set is an intersection of closed halfspaces (those corresponding to supporting hyperplanes obtained by starting with $v \notin C$ and considering all $v \notin C$).

- An obvious question was do all points on rel.bound. have a supporting hyperplane ? One way to answer is to look at the Tangent cone and the Normal cone at that point. Tangent cone of a convex set C at a point $x \in C$ is defined as $T_C(x) = \{h \mid \exists t > 0 \ni x + th \in C\}$. Normal cone is the dual cone of the tangent cone. Take any direction which is negative of a vector in the Normal cone, that must define a supporting hyperplane.

[1] The outline of the proof we gave in lecture is at: `http://en.wikipedia.org/wiki/Hilbert_projection_theorem` and `http://www.convexoptimization.com/wikimization/index.php/Moreau%27s_decomposition_theorem`

- With this it is easy to conclude that for a closed convex set, take all supporting hyperplanes at all rel.bound. points and the intersection of their negative halfspaces is the convex set and hence gives a dual description. (Normal cone at an rel.int. point will be 0 as the Tangent cone is the entire affine hull of the set. And hence we cannot extend the argument to show existence of a separating hyperplane at an int. point).

- Now came the question will the above give the "most efficient" dual description? Looking at polytopes we observed that not all rel.bound. points may be needed only some which we defined as extreme points[2] are enough. And if the Normal cone at an extreme point has more than one vector then it is enough to take the fewest vectors which provide a primal description of the Normal cone and the negatives of them are the only supporting hyperplanes which need to be considered at any extreme point. This we argued is the most efficient dual description.

- Interestingly, it turns out that for any compact convex set C, we have $C = conv(ext(C))$, where $ext(C)$ is the set of all extreme points of C and gives the most efficient primal description[3]. Hence extreme points are useful for efficient representations in both the primal/dual views. Easy counter examples show that this theorem fails to apply on unbounded closed convex sets even if they are polyhedral.

- We then noted that the above separation result can be generalized to the separation theorem (refer sec B.2.5.B in Nemirovski [2005] for details). We then wrote down a lemma which follows from the separation theorem, known as the Farkas lemma (refer sec.B.2.4), and saw that duality sometimes helps us answer difficult questions by posing the difficult question as an easy question on a dual. Here is one way of writing Farkas lemma:

  Lemma 7.0.2. *Consider two sets of linear inequalities* (S_1) *given by:* $Ax = b, x \geq 0$ *(here,* x *is the dummy variable) and* (S_2) *given by* $A^\top y \geq 0, b^\top y < 0$ *(here,* y *is the dummy variable). Separation theorem gives that* (S_1) *is solvable/consistent/feasible if and only if* (S_2) *is not-solvable/in-consistent/in-feasible.*

  There are many ways of writing down such results and in general are called as "Theorems on Alternative". Some of them appear in theorem 1.2.1 and exercises 1.2-1.4 in Nemirovski [2005]. We will see later that such theorems form a basis for duality theory in optimization problems.

[2] Refer sec. B.2.7.A in Nemirovski [2005] for a definition of extreme point.

[3] This is known as the Krein-Milman theorem. Refer sec.B.2.7.B in Nemirovski [2005] for details.

- We then went on to prove some results whose proves we skipped earlier: i) polyhedral cones have finite dual description ii) For a closed convex set C which has 0, we have that $polar(polar(C)) = C$. In particular, for any closed cone C we will have $dualcone(dualcone(C)) = C$.

- Here is a sketch of proof for i): We already know that the dual cone of a polyhedral cone will have finite dual description i.e., intersection of finite number of closed half-spaces. Using a result u will prove in problem sets we showed that intersection of two polyhedral cones is a polyhedral cone and hence by induction we get that the dual cone of a polyhedral cone is a polyhedral cone. Using ii) we get that the dual of dual, which is primal, has a finite dual description (we infact complete a cycle and all required proofs).

- For proof of ii) refer Proposition B.2.2 in Nemirovski [2005].

- We intuitively argued results like exercise B.14 and B.15 in Nemirovski [2005]. We noted that such characterizations are sometimes important.

- We concluded the lecture by summarizing key aspects of the subsets we learnt about. We will begin discussing real-valued functions on finite dimensional Hilbert spaces from the next lecture.

- Mandatory reading: B.2.5, B.2.7, B.1.5, B.1.6 in Nemirovski [2005]. Optionally also read chp. 11 and 18 in Rockafellar [1996]

Lecture 8

- We started discussing the third ingredient of optimization problems which is the objective function — which is a real-valued function over a Vec.Spa/Inn.Prod.Spa. i.e., $f : V \mapsto \mathbb{R}$. We already know many examples for such functions: eg. norms, etc.
- We then began discussing special classes of functions, starting with linear functions — Given a vector space $\mathcal{V} = (V, +, \cdot)$, a function $f : V \mapsto \mathbb{R}$ is linear iff $f(\sum_{i=1}^{n} \lambda_i x_i) = \sum_{i=1}^{n} \lambda_i f(x_i)\ \forall\ (x_i \in V, \lambda_i \in \mathbb{R}, n \in \mathbb{N})$ i.e., Image of a linear combination of some points under the function is the same linear combination of images of those points. Basically, functions where linear intra-extrapolation is accurate. Note that since the domain is V itself, all linear combinations of vectors must be in V.
- If the vec.spa. $\mathcal{V}$ is also equipped with an inner-product $\langle\rangle$, then we showed that f is linear iff $f(x) = \langle a, x\rangle\ \forall\ x \in V$, for some $a \in V$ (the trick was to employ our usual saviour – the orthonormal basis). It also easy to show that no two linear functions can have the same a and when as are different then the linear functions are different. Hence there is a bijection between the set V and the set of all linear functions on V (denoted by L say). Moreover, if a define $+'$ and $\cdot'$ over L, which is the usual point-wise $+$ and point-wise $\cdot$, then a linear of two functions can be got by the same linear combination of the corresponding a and hence the two vectors spaces: $\mathcal{V}$ and $\mathcal{L} = (L, +', \cdot')$, themselves are equivalent![1]
- We discovered that linear functions and linear sets have a relation: graph[2] of a linear function is a hyperplane through the origin in the vector space which is direct sum of $\mathcal{V}$ and $\mathbb{R}$. We stressed on the point that the graph is a set in $n + 1$ dim. if the function is in n dim.

[1] A name of the vector space of all linear functions over a set of vectors is conjugate-space.

[2] given a function $f : S \mapsto \mathbb{R}$, where $S \subset V$ and $\mathcal{V} = (V, +, \cdot)$ is a vector space, the graph of f is defined as $graph(f) = \{(x, y) \in \mathcal{V} \oplus \mathbb{R} \mid f(x) = y\}$

- Basing on this discussion we defined affine functions — a function $f : A \mapsto \mathbb{R}$, where $A \subset V$ is an affine set in V, is affine iff $f(\sum_{i=1}^{n} \lambda_i x_i) = \sum_{i=1}^{n} \lambda_i f(x_i) \; \forall \; (x_i \in A, \lambda_i \in \mathbb{R}, \sum_{i=1}^{n} \lambda_i = 1, n \in \mathbb{N})$ i.e., Image of an affine combination of some points under the function is the same as the affine combination of images of those points. By definition all linear functions are affine functions (obviously the converse is not true).

- We took all affine functions defined on V itself[3] and showed that $f : V \mapsto \mathbb{R}$ is affine iff $f(x) = \langle a, x \rangle - b \; \forall \; x \in V$ for some $a \in V, b \in \mathbb{R}$. Now it is easy to give numerous examples of affine functions.

- With this characterization of affine functions, it is easy to show that the graph of an affine function $f : A \mapsto \mathbb{R}$ ($A \subset V$ is an affine set)is always a hyperplane in the vector space which is the direct sum of the subspace associate with the affine set A and $\mathbb{R}$.

- We then gave the obvious definition for conic functions (which would connect them to conic sets): A function $f : C \mapsto \mathbb{R}$, where $C \subset V$ is a conic set, is a conic function iff $f(\sum_{i=1}^{n} \lambda_i x_i) \leq \sum_{i=1}^{n} \lambda_i f(x_i) \; \forall \; (x_i \in C, \lambda_i \geq 0, n \in \mathbb{N})$ i.e., Image of a conic combination of some points under the function is over-estimated by the same conic combination of images of those points. By definition all linear functions are conic functions (obviously the converse is not true).

- It is easy to show that $f : C \mapsto \mathbb{R}$, where $C \subset V$ is a conic set, is a conic function if and only if $epi(f)$ is a conic set. Here $epi(f)$ is the epigraph[4] of f.

- We showed that $f(x) = \|x\|$ where $\|\cdot\|$ is any norm, is a conic function. Hence $f(x) = \|x\|_p (p \geq 1), f(x) = \|x\|_Q (Q \succ 0), f(X) = \|X\|_F$ are all conic functions. Moreover, all semi-norms[5] are conic functions for eg. $f(x) = \|x\|_Q (Q \succeq 0)$ is a conic function.

- We took the function $f(A) = \sup_{\|x\| \leq 1} x^\top A x$, where A is a square matrix. It was easy to check it was a conic function. If A is psd, then it indeed defines a norm[6] of A — the max. eigen value of a psd matrix.

---

[3] In probset u will show a similar thing will happen for all affine functions on A which is an affine subset of V.

[4] given a function $f : S \mapsto \mathbb{R}$, where $S \subset V$ and $\mathcal{V} = (V, +, \cdot)$ is a vector space, the epigraph of f is defined as $epi(f) = \{(x, y) \in \mathcal{V} \oplus \mathbb{R} \mid f(x) \leq y\}$.

[5] Refer http://en.wikipedia.org/wiki/Norm_%28mathematics%29 for a defn.

[6] Refer http://en.wikipedia.org/wiki/Matrix_norm for norms over matrices. Note that there are atleast 4 ways of defining norms over matrices.

- We then defined the next natural class of functions those which are convex: A function $f : C \mapsto \mathbb{R}$, where $C \subset V$ is a convex set, is a convex function iff $f(\sum_{i=1}^{n} \lambda_i x_i) \leq \sum_{i=1}^{n} \lambda_i f(x_i) \; \forall \; (x_i \in C, \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1, n \in \mathbb{N})$ i.e., Image of a convex combination of some points under the function is over-estimated by the same convex combination of images of those points. By definition all linear functions, affine function and conic functions are convex functions (obviously the converse is not true).

- It is easy to show that $f : C \mapsto \mathbb{R}$, where $C \subset V$ is a convex set, is a convex function if and only if $epi(f)$ is a convex set.

- We showed that $f(x) = x^2$ (parabola) is a convex function. With this it was clear that squared norms $f(x) = ||x||^2$ are all convex[7].

- We showed that if f and g are two convex functions (defined over the same vector space), then $h = max(f, g)$, which is the point-wise maximum i.e., $h(x) = max(f(x), g(x)) \; \forall \; x$, is also convex. By induction, the point-wise maximum over any finite number of convex functions is a convex function. We gave examples of such convex functions.

- This extends to the case of point-wise maximum of an arbitrary (possibly uncountable) set of convex functions (the proof is easiest if done from the characterization of convex functions as those with epigraphs as convex sets). Hence we have the function h defined[8] as $h(y) = \sup_{x \in X} f(x, y)$, where each $f(x, \cdot)$ is a convex function (with fixed x, we have that $f(x, y)$ is convex in y), is itself is a convex function.

- In particular, we have that given any arbitrary set S of vectors, the function $s(y) = \sup_{x \in S} < x, y >$ is a convex function. Please visualize such functions starting from 2-d examples. Infact u will observe (u can prove) that this function is a conic function. Hence this gives a way of building conic functions from arbitrary sets. The function s defined above is called the support function of the set S. The name comes from the fact that this gives supporting hyperplanes to the epigraph cone ... we will study more of this in the next lecture.

- Before concluding, we mentioned the Jensen's inequality, which is a direct consequence from the definition of a convex function, and its use. We noted that it is a very fundamental and generic inequality from which many familiar inequalities like AM-GM, Holder's inequality etc. can be derived.

[7] This is happening because squared norm is a composition of the functions x^2 and the norm. This motivates us to study compositions (and in general operations) of functions which preserve convexity.

[8] X represents the arbitrary index set for the convex functions.

- Mandatory reading: sec. 3.1,3.2 in Boyd and Vandenberghe [2004]. Sec. C.1 in Nemirovski [2005]. Optionally also read chp. 4 in Rockafellar [1996].

Lecture 9

- After a quick recap, we focused on the support function definition. We took examples support functions for some sets like i) $\{-1, 1\}$ whose support function was $f(x) = |x|$ and the set ii) unit circle centered at origin whose support function was $f(x) = \|x\|_2$ (ice-cream cone). Encouraged by the examples we asked whether support function of any set is a conic function (ofcourse we proved in last lecture it is a convex function)? The answer is yes and it is easy to show it using the very definition of conic functions.

- The converse question was given any closed conic function[1], can it be always written as a support function of some set? The answer is yes and the proof follows from the fact that epigraph of closed conic function is a closed conic set and hence a supporting hyperplane (through origin) exists at any point on the graph. Re-writing such a dual description of the epigraph using the *sup* notation gives us the required form. Hence support functions give a new definition/characterization of (closed) conic functions — from now we refer to this as the dual description of the conic function in question.

- Now encouraged by the dual norm definition we gave earlier which arose out of the support function of a normed-ball, we asked the question can we in general define "duals" of (closed) conic functions (denoted by f^*) such that: $epi(f^*)$ is the dual cone of $epi(f)$? If we do such a thing, then we will achieve our goal of extending the dual-norm idea to all conic functions.... this is because, then $epi(f^{**})$ will be the dual cone of $epi(f^*)$ which must be $epi(f)$ and hence $epi(f^{**}) = epi(f)$ and $f^{**} = f$. Looking at the dual-norm definition we conjunctured that an appropriate definition might be: $f^*(y) = \sup_{x \in \{z \mid f(z) \leq 1\}} \langle x, y \rangle$. We took an example of a conic function which is a semi-norm and show that all the nice results we desired are holding for this example. Students were asked to prove or disprove this is general. The answer will be revealed in next lecture.

[1] A closed function is a function whose epigraph is closed

- These results (as expected) might not work with conic functions which are not closed. We gave some examples of convex/conic functions which are not closed and realized that they are pathetic cases and not of interest to us anyway.

- The obvious question now was if we can repeat this business of duality with convex functions? In particular, is there a dual description of a convex function (like support function for conic functions)? Ofcourse the answer is again given by the dual description of epigraphs of convex functions — which are now supporting hyperplanes that need not pass through origin. This gave us that a closed convex function f can always be expressed as $f(x) = \sup_{y \in Y} \langle y, x \rangle - b_y$. It is also easy to show that, starting from arbitrary sets Y and b_y we can form convex functions by using the form above and hence this gives a characterization for (closed) convex functions. This can be called as the dual description of (closed) convex functions.

- Changing notation slightly and calling b_y as some $f(y)$ (i.e., some function which takes y and gives a number. Now Y is domain of f) we re-wrote above dual form as: $f^*(y) = \sup_{x \in dom(f)} \langle y, x \rangle - f(x)$. This is valid definition for all y such that $f^*(y) < \infty$ i.e., y in the domain of f^*. We called f^* as the conjugate of f. Sometimes it is also called as Legendre transformation or Fenchel's dual of f. Obviously the conjugate of any function is a closed convex function.

- We also gave a sketch of proof for the statement: If f is closed convex, then $f^{**} = f$. We concluded the lecture with some examples of conjugate functions.

- Mandatory reading: sec. 3.3 in Boyd and Vandenberghe [2004]. Sec. C.6.3 in Nemirovski [2005]. Optionally also read chp. 12,13,26 in Rockafellar [1996]. The duality correspondences for functions are exhaustively dealt with in the Rockafellar [1996] book and hence is best option for this topic.

Lecture 10

- We answered the question about the right definition of dual of a (closed) conic function. We began by taking an example[1] of a conic function which is not a semi-norm: $f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$. On this example our previous (wrong) definition of dual conic function did not give the desired result. However it showed a quick and easy fix which lead us to the correct definition: $f^*(y) = \sup_{x \in \{z \mid f(z) \leq 1\}} \langle -y, x \rangle$. We call f^* as the dual (conic) function of f. With this definition it is easy to show that $epi(f^*)$ is dual cone of $epi(f)$ and hence $f^{**} = f$ for any f which is a closed conic function.

- Some important uses of the conjugate/dual functions are notable:

  1. it is easy to see that the value of conjugate function at the origin $f^*(0)$ is the infimum of the function values of f. This simple connection is very useful: we can get a global property of f by looking at a local property of f^* and by duality we can get more such relations (we may see some of them later).
  2. it is easy to see that $f(x) + f^*(y) \geq \langle x, y \rangle$ for all $x \in dom(f), y \in dom(f^*)$. This inequality is called the Conjugate/Fenchel's inequality. This is again a very useful inequality (like the Jensen's) from which main important inequalities like AM-GM and Holder's inequality follow.

- We know that the dual description of epigraph lead to discovery of dual and conjugate functions. We will now re-write the same in another useful form: consider any convex function f and a point in the rel.int. of its domain, say x_0. Note that $(x_0, f(x_0))$ is a point on the relative boundary of the epigraph of f and existence of a supporting hyperplane at that point is assured. Let the equation of the supporting hyperplane at $(x_0, f(x_0))$ be $\langle (a, a_0), (x - x_0, z -$

[1] Uma and Gopi gave simple linear functions which are examples of conic functions that are not a semi-norm.

$f(x_0))\rangle = 0$ for some $a \in dom(f), a_0 \in \mathbb{R}$. We also know if (x', z') is a point in the epigraph of f, then $\langle (a, a_0), (x' - x_0, z' - f(x_0))\rangle \leq 0$. In particular looking at the inequality for a point $(x_0, z) \in epi(f)$ gives that $a_0 \leq 0$. Now $a_0 = 0$ implies a "vertical" supporting hyperplane — which cannot occur at a rel.int. point (we can prove this formally too after knowing about Lipschitz conts. of f). Therefore we can divide the whole inequality by $-a_0$ (which is > 0). Rearranging terms and using $z = f(x)$ gives for any $x_0 \in relint(dom(f))$ that: $f(x) \geq f(x_0) + \langle \nabla f(x_0), (x - x_0)\rangle \ \forall x \in dom(f)$, where $\nabla f(x_0) = -a/a_0$. Infact we know that there might be multiple supporting hyperplanes i.e., multiple (a, a_0). So we define sub-gradient of f at x_0, denoted by $\nabla f(x_0)$, as any vector (in dom(f)) which satisfies $f(x) \geq f(x_0) + \langle \nabla f(x_0), (x - x_0)\rangle \ \forall x \in dom(f)$. This later inequality will henceforth be referred to as the sub-gradient inequality centered at x_0. We already proved that for any convex function a sub-gradient exists at any relint point of the domain.

- Conversely, one can easily show that[2] if there is a function which satisfies the sub-gradient inequality centered around any domain point then it must be a convex function. One can limit the condition of sub-gradient inequality satisfaction to all points in rel.int. of the domain, but assume the function is continuous and prove the same result that the function must be convex. This gives to the following characterizations for a closed convex and continuous convex function:

  Theorem 10.0.3. *Let $\mathcal{V} = (V, +, \cdot, \langle\rangle)$ be a inner-product space and $C \subset V$ be a convex set. Let $f : C \mapsto \mathbb{R}$ be a function. Then the following statements are true:*

  1. *Assume that f is closed and $x_0 \in C$, then f is convex if and only if $f(x) \geq f(x_0) + \langle \nabla f(x_0), (x - x_0)\rangle \ \forall x \in dom(f)$. i.e., convexity is characterized by sub-gradient inequality satisfaction centered at any domain point.*
  2. *Assume that f is continuous and $x_0 \in relint(C)$, then f is convex if and only if $f(x) \geq f(x_0) + \langle \nabla f(x_0), (x - x_0)\rangle \ \forall x \in dom(f)$. i.e., convexity is characterized by sub-gradient inequality satisfaction centered at any relint point of the domain.*

- The set of all sub-gradients at a point x_0 is called the sub-differential at that point and is denoted by $\partial f(x_0)$.

- The sub-gradient inequality infact gives a dual description of the sub-differential set and proves that it is always a closed convex set. We briefly mentioned the possible use of such a realization.

[2] Refer to the problem set.

- Most important benefit of this characterization is an immediate answer to the question of minimizers for a convex function defined on the entire vector space (or infact any open set; we fill focus on this point later): i.e., $argmin_{x \in V} f(x)$. The statement is: $x_0 \in \text{argmin}_{x \in V} f(x)$ if and only if $0 \in \partial f(x_0)$. It is also was easy to see that the set of all minimizers i.e. $argmin_{x \in V} f(x)$ is itself a convex set – which infact is a special level-set: we define level-set at $\alpha \in \mathbb{R}$ as $L_\alpha = \{x \in dom(f) \mid f(x) \leq \alpha\}$. For any α it is easy to see that this set is a convex set.

- We motivate that when normal cone at a point is a singleton then the sub-differential is also a singleton and in this case it corresponds to the (may be familiar) notion of gradient. To see this we started looking at some convex functional analysis.

- We began by asking if convex functions are bounded (i.e., are the function values over the domain bounded). Easy counter-examples show the converse; but what is true is that they are locally bounded and infact locally Lipschitz continuous. Refer to Proposition C.4.1 in Nemirovski [2005] for related definitions and proofs. Important take-home is every convex function is locally Lipschitz continuous at any rel.int point in the domain.

- From this it followed that all convex functions are continuous at any relint. point in the domain. In the subsequent lecture we will see the connection between gradients and sub-gradients etc.

- Mandatory reading: Sec. C.6.2, C.4 and C.5 in Nemirovski [2005]. Optionally also read chp. 10, sections on sub-gradient in chp. 23 in Rockafellar [1996].

Lecture 11

- We revised the notion of derivative of real-valued function defined over reals at a point: Let $C \subset \mathbb{R}$. A function $f : C \mapsto \mathbb{R}$ is said to be differentiable at $x \in C$ iff there exists a number $f'(x) \in \mathbb{R}$ such that $\lim_{y \to x} \frac{f(y)-f(x)-f'(x)(y-x)}{|y-x|} = 0$. In this case one can show that the number $f'(x)$ is unique and is called as the derivative or gradient of the function f at x. We wrote it this defintion in some equivalent forms and noted the intuition about derivative representing instantaneous slope.

- Looking at this definition, we defined the concept of differentiability for real-values functions defined over arbitrary Hilbert spaces: Let $\mathcal{V} = (V, +, \cdot, \langle\rangle)$ be a Hilbert space, $||||$ is the inner-product induced norm and $C \subset V$. A function $f : C \mapsto \mathbb{R}$ is said to be differentiable at $x \in C$ iff there exists a vector[1] $\nabla f(x) \in V$ such that $\lim_{y \to x} \frac{f(y)-f(x)-\langle \nabla f(x), y-x \rangle}{||y-x||} = 0$. In this case one can show that the vector $\nabla f(x)$ is unique and is called as the derivative or gradient of the function f at x.

- We then wanted to see if our slope intuition is still valid for the generic definition of the gradient. Since now there could be many directions we chose to move along a vector, say $u \in V$, and write y as $x + hu$ where $h \in \mathbb{R}$. Re-writing the gradient definition we obtained: $\langle \nabla f(x), u \rangle = \lim_{h \to 0} \frac{f(x+hu)-f(x)}{h}$. In the case $||u|| = 1$, this quantity is called the directional derivative. Ofcourse this gives back our instantaneous slop interpretation in direction of u.

- If $\{e_1, e_2, \ldots, e_n\}$ forms an orthonormal basis for the finite dimensional Hilbert space $\mathcal{V}$ in question, then it is easy to see that $\nabla f(x)$ is completely determined by n numbers: $\langle \nabla f(x), e_1 \rangle, \ldots, \langle \nabla f(x), e_n \rangle$. For the specific case of Euclidean space with standard basis, these n numbers are simply the partial

[1] We are here using the same symbol for both sub-gradient and gradient. This is fine as we shall soon show that gradient is indeed a sub-gradient.

derivatives, and hence in this case the gradient vector is the vector of partial derivatives. Similarly for Hilbert spaces of matrices with the standard basis, the gradient (which is a matrix) is the matrix of all partial derivatives.

- From the directional derivative definition it is clear that the instantaneous direction of maximum increase of the function is $\nabla f(x)$ and the instantaneous direction of maximum decrease is $-\nabla f(x)$. This observation actually motivates algorithms like gradient-descent, which we will study later.

- We then proved[2] the following theorem, which gives yet another characterization for convex functions (under differentiability assumptions):

  Theorem 11.0.4. *Let $\mathcal{V} = (V, +, \cdot, \langle\rangle)$ be an inner-product space and $C \subset V$ be a convex set. Let $f : C \mapsto \mathbb{R}$ be a continuous function. Assume that the function f is differentiable at any rel.int. of C, say $x_0 \in relint(C)$ and the gradient is $\nabla f(x_0)$. Then f is convex if and only if $f(x) \geq f(x_0) + \langle \nabla f(x_0), (x - x_0) \rangle \ \forall x \in dom(f)$. i.e., convexity is characterized by sub-gradient inequality satisfaction centered at any relint point of the domain.*

- One immediate conclusion from the above theorem is that (for convex functions) the gradient is indeed a sub-gradient. This is the reason we chose to use the same symbol for both.

- We then computed gradients for various convex functions: i) $f(x) = \langle a, x \rangle - b$ (affine function in any finite-dim. Hilbert space). $\nabla f(x) = a$ ii) $f(x) = \|x\|$ (conic function which is the inner-product induced norm in any fin.dim. Hilbert space; i.e., ice-cream cones). $\nabla f(x) = \frac{x}{\|x\|}$ if $x \neq 0$ and when $x = 0$ the sub-differential set[3] is $\partial f(0) = \{y \mid \|y\| \leq 1\}$. We also noted that if $f(x) = \|x\|$ is any norm (need not be the inner-product induced one), then $\partial f(0) = \{y \mid \|y\|_* \leq 1\}$. iii) $f(x) = \|x\|^2$ (norm-squared function with norm as the induced one; i.e., paraboloids). $\nabla f(x) = 2x$. iv) generalization of norm-squared which is homogeneous quadratic functions[4] $f(x) = x^\top A x$ (this eg. is in Euclidean space and here A is symmetric). $\nabla f(x) = 2Ax$. v) (non-homogeneous) quadratic function $f(x) = x^\top A x + b^\top x + c$. $\nabla f(x) = 2Ax + b$.

[2] Refer sec. C.3 in Nemirovski [2005] or sec. 3.1.3 in Boyd and Vandenberghe [2004] for a proof.

[3] This comes from defn. of dual norm and the fact that the inner-product induced norms are always self-dual

[4] We also gave intuition for homogeneous quadratic functions in other spaces provided we know how to built linear functions which take vector in that space and give another one in the same. If we denote such a function by $\mathcal{A}$, then $f(x) = \langle x, \mathcal{A}(x) \rangle$ will be the homogeneous quadratic functions. In case of Euclidean vectors, $\mathcal{A}(x) = Ax$ and in case of matrices they are given by tensor-matrix multiplication. Insterested students can read up material on self-adjoint operators.

The usual trick for computing ∇f was to guess the gradient and verify it either in the limit definition or the sub-gradient inequality.

- We then noted (but did not prove) that the usual rules for computing derivative of a linear combination of functions do hold for gradients and subgradients i.e., if $h = \lambda f + \rho g$, then $\nabla h(x) = \lambda \nabla f(x) + \rho \nabla g(x)$. Infact many other such rules hold but are beyond scope of our course.

- We then went on to second order derivatives (we restrict ourselves to Euclidean spaces as the definition otherwise would require homogeneous quadratics in those spaces, which we skipped studying and mentioned in footnote above): Let $C \subset \mathbb{R}^n$. A function $f : C \mapsto \mathbb{R}$ is said to be double differentiable at $x \in C$ iff there exists a symmetric matrix $H(x)$ such that $\lim_{y \to x} \frac{f(y) - f(x) - \langle \nabla f(x), y - x \rangle - \frac{1}{2}(y-x)^\top H(x)(y-x)}{\|y-x\|^2} = 0$. In this case one can show that the symmetric matrix $H(x)$ is unique and is called as the Hessian (or double derivative) of the function f at x.

- Again by fixing $y = x + hu$, we get $\frac{1}{2} u^\top H(x) u = \lim_{h \to 0} \frac{f(x+hu) - f(x) - h\langle \nabla f(x), u \rangle}{h^2}$. From this two facts are evident: i) the Hessian matrix is simply the matrix of all possible second-order partial differentials. ii) since term inside the limit in RHS of above equation is non-negative (by the sub-gradient inequality condition), we have that $\frac{1}{2} u^\top H(x) u \geq 0$ for all u in the tangent-cone of the domain of the function at x. And for an interior point in domain, the tangent-cone is the entire space and this amounts to the Hessian being psd. Infact we can prove the Theorem C.2.1 in Nemirovski [2005], which gives the second-order differential characterization of convex functions.

- Using this useful characterization we proved convexity of many functions: i) $f(x) = x^p$ where p is even ii) $f(x) = x^p$ where $x > 0$ and $p \leq 0$ *or* $p \geq 1$ iii) $f(x) = e^{ax}$ iv) $f(x) = -log_a(x)$ (here, $x > 0$ and $a > 1$) v) $f(x) = \sum_{i=1}^n x_i log(x_i)$ (negative entropy; here $x > 0$) vi) $f(x, y) = \frac{x^2}{y}, y > 0$ vii) $f(x, M) = x^\top M^{-1} x, M \succ 0$ (we wrote down one meaningful optimization problem involving such a function).

- Mandatory reading: Sec. C.2.2 and C.3 in Nemirovski [2005]; Sec. 3.1.3 in Boyd and Vandenberghe [2004]. Optionally also read chp. 23, 24 and 25 in Rockafellar [1996].

Lecture 12

[A bonus quiz asking to list down 5 possible definitions of convex functions (with various assumptions) was floated.]

- We begin studying optimization problems (Mathematical programs): problems of the form

$$\begin{aligned} \min_{x \in X} \quad & f(x) \\ \text{s.t.} \quad & x \in C, \end{aligned} \tag{12.1}$$

  where X is called the domain (space in which variable moves or equivalently the function is defined); f – the function being minimized is called the objective function; and C is called the constraint set of the optimization problem in (12.1).

- We then defined various other terms which are relevant for an optimization problem:
  - Feasible solution: x is called a feasible solution iff $x \in X \cap C$.
  - Feasibility set: set of all feasible solutions i.e., $\mathcal{F} = X \cap C$.
  - A program is said to be feasible iff the feasibility set is non-empty. If feasibility set is empty then we say the program is infeasible.
  - Optimal value: is the infimum of objective function values over the feasibility set i.e., $inf(f(x) \mid x \in X \cap C)$ provided the feasibility set is non-empty and in case feasibility set is empty we define it as '∞'.
  - If the optimal value is $-\infty$, then we call such a program an unbounded one. If the optimal value is $> -\infty$, then we call it as bounded program.
  - We say two optimization problems are equal iff their optimal values are the same. Needless to say, with this the program is itself equal to its optimal value.

- Optimal solution: x is an optimal solution iff $x \in \mathcal{F}$ and $f(x) \leq f(y) \; \forall \; y \in \mathcal{F}$.
- Optimality set: the set of all optimal solutions is called the optimality set and is denoted by the symbol:

$$\begin{aligned} \arg\min_{x \in X} \quad & f(x) \\ \text{s.t.} \quad & x \in C, \end{aligned}$$

- A program is said to be solvable iff optimality set is non-empty.

- We also listed down four fundamental questions we would like to answer: questions regarding i) boundedness ii) solvability iii) uniqueness of optimal solution iv) optimality conditions. The hope is that we can give necessary and/or sufficient conditions for each of these.

- We said that the answers to these are simple and elegant for the case of a special class of programs called Convex programs: programs with convex objective function and convex constraint set. Other names for convex programs are convex optimization problems, or simply convex problems.

- We gave two examples of programs and commented on their convexity:

  1. We said the program $\min_{x \in \mathbb{R}^n} x^\top A x + b^\top x + c$ is convex if and only if A is psd. (this followed from the Hessian of objective, which in this case is $2A$).
  2. We proved (using various results about convex sets and convex functions) that the following nice geometrically meaningful problem, which is that of finding the smallest ellipsoid enclosing a given set of m points (say, x_i), is convex[1]:

$$\begin{aligned} \min_{c \in \mathbb{R}^n, M \succ 0, R \in \mathbb{R}} \quad & R \\ \text{s.t.} \quad & ||x_i - c||^2_{M^{-1}} \leq R \; \forall \; i = 1, \ldots, m \end{aligned} \tag{12.2}$$

- We conjunctured that convex programs with bounded feasibility sets are themselves bounded (providing a sufficient condition for our first question). We hinted that it follows from Theorem C.4.1 in Nemirovski [2005]. In the next lecture we will detail the proof and carry-on with the other questions.

- Mandatory reading: Sec. D.1 in Nemirovski [2005]; Sec. 4.1, 4.2 in Boyd and Vandenberghe [2004].

[1] The following were the key steps in the proof: i) ascertaining the domain, objective function and constraint set for the problem ii) showing that the domain, objective and constraint set are all convex

Lecture 13

- We began by proving a sufficiency condition for the first fundamental question of boundedness. We always assume that (P) represents the following convex program:

$$\begin{aligned} &\min_{x \in X} && f(x) \\ &\text{s.t.} && x \in C \end{aligned} \tag{13.1}$$

Theorem 13.0.5. *The convex program (P) is bounded if the feasibility set $\mathcal{F} = X \cap C$ is bounded.*

- The proof depends on theorem C.4.1 in Nemirovski [2005][1] and the Hiene-Borel theorem[2] and the following lemma:

Lemma 13.0.6. *Given a convex function f and convex set C in the domain of f, we have that $\min_{x \in C} f(x) = \min_{x \in cl(C)} f(x) = \min_{x \in relint(C)}$.*

- Lets first prove the lemma: Just looking at the size of constraint sets we know $\min_{x \in cl(C)} f(x) \leq \min_{x \in C} f(x) \leq \min_{x \in relint(C)} f(x)$. But we also know that $\min_{x \in cl(C)} f(x) = \min\left(\min_{x \in relint(C)} f(x), \min_{x \in relbnd(C)} f(x)\right)$. Starting from sub-gradient inequality and then taking limit as $x_n \in relint(C)$ approaches $x \in relbnd(C)$, it followed that $\min_{x \in relint(C)} f(x) \leq \min_{x \in relbnd(C)} f(x)$. From this the result of the lemma follows.

- Here is a sketch of the proof for theorem 13.0.5: from theorem C.4.1 in Nemirovski [2005] we get that f is lower bounded on K. Now if only this

[1] We also gave examples of functions like $\|x\|$ and $\|x\|^2$ and talked about their Lip. cont. We noted that if a function is Lip. conts. and is diff. then the supremum of derivative over the set considered must be less than less than or equal to the Lip.const.

[2] http://en.wikipedia.org/wiki/Heine%E2%80%93Borel_theorem

theorem C.4.1 were true[3] for any compact set (not necessarily in the relint), then by looking at $\min_{x \in cl(\mathcal{F})} f(x)$ it is trivial to prove theorem 13.0.5. Now unfortunately K is restricted to be in relint and one compact set in relint cannot cover[4] the relint, so the idea is to cover the relint with open balls: one for each relint point such that the open ball is itself inside the relint (this is possible by defn. of relint). Now over each of these balls by theorem C.4.1, we have a lower bound for f (infact for the corresponding closed balls itself we have a lower bound by the theorem). Now still there could be infinite balls and infinite lower bounds and the infimum can still go to infinity. So if we can somehow get a finite cover then we will be done. The idea is to use the Hiene-Borel Theorem (HBT), which actually talks about finite sub-cover. But again the HBT can be applied to only compact sets, so we extend our existing open ball cover of relint to the relbnd of closure of $\mathcal{F}$ and if we can show still at each ball there is a lower bound then we will get a finite sub-cover and then we will have to do a minimum over finite number of lower bounds which will give a lower bound $> -\infty$. For this extension of cover simply take every relbnd point in the closure and put a open ball of (say) unit radius. For the sake of lower bound we already know that none of these balls matter (for e.g. we can simply choose the lower bound for these balls as ∞). Using the HBT on this open cover for closure of $\mathcal{F}$ gives the result in the theorem and completes the proof.

- We emphasized on that point that the sufficiency condition in theorem 13.0.5 may no longer be a sufficient condition if (P) is not convex. We gave as the example problem of minimizing $\log(x)$ over $(0, 1]$. Here feasibility set is bounded but the problem is unbounded.

- We moved on to a sufficiency condition for the second fundamental question (which was trivial to prove at this stage):

  Theorem 13.0.7. *The convex program (P) is solvable if the feasibility set $\mathcal{F} = X \cap C$ is compact and the objective function f is continuous in it.*

- Then we ventured into determining when can the optimal solution be unique this will be the case where near the optimal solution (atleast locally) the function strictly increases on all sides (in tgt. cone). While this is a condition on the optimal solution point, we said nevertheless we can get a sufficiency

[3] We know that the theorem is not true for any compact set and the restriction of being in the relint is important. But just to ease the argument say we assume this (later we will correct ourselves).

[4] http://en.wikipedia.org/wiki/Cover_%28topology%29

condition by insisting on this "strict convexity" everywhere. This led to defining strictly convex functions: A function $f : C \mapsto \mathbb{R}$, where $C \subset V$ is a convex set, is strictly convex iff $f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y) \; \forall \; \lambda \in (0,1), \; x, y \in C$. It was immediate that all strictly convex functions are convex. We also defined strictly convex program as a convex program with strictly convex objective function. By a simple proof by contradiction, the following theorem was immediate:

Theorem 13.0.8. *A strictly convex program, if solvable has a unique optimal solution.*

- We also proved the first order and second order condition for strict convexity. These appear in the problem sets. We gave examples of strictly convex and non-strictly convex functions.

- We went on to characterize optimal solutions and the following was immediate: an $x^* \in \mathcal{F}$ will be optimal if and only if $f(x^*) \leq f(x) \; \forall \; x \in \mathcal{F}$ and this is if and only if[5] $f(x^*) \leq f(x^* + hu) \; \forall \; u \in T_{\mathcal{F}}(x^*)$ and all appropriate $h > 0$ such that $x + hu \in \mathcal{F}$. Now two special cases are easy and interesting:

  1. Suppose $x^* \in int(\mathcal{F})$. In this case the tangent cone $T_{\mathcal{F}}(x^*)$ has ALL directions. By theorem C.5.1 in Nemirovski [2005], the above nec. and suff. condition simply becomes $f(x^*) \leq f(x) \; \forall \; x \in X$ i.e., in this case the optimal solution is actually the optimal solution with feasibility set as entire domain. By definition of sub-gradient, this is if and only if $0 \in \partial f(x^*)$. This proves the following theorem:

     Theorem 13.0.9. *Let $x^* \in int(\mathcal{F})$. x^* is an optimal solution of (P) if and only if $0 \in \partial f(x^*)$.*

     In particular, the qualification that $x^* \in int(\mathcal{F})$ is satisfied by all points in $\mathcal{F}$ if it is open or the entire vector space. Hence in this case all solutions can be characterized by $0 \in \partial f(x^*)$.

  2. Suppose $x^* \in \mathcal{F}$ and f is differentiable at x^*. By looking at the definition of directional gradients along $u \in T_{\mathcal{F}}(x^*)$, the following theorem is obvious:

     Theorem 13.0.10. *Let $x^* \in \mathcal{F}$ and f be differentiable at x^*. x^* is an optimal solution of (P) if and only if $\langle \nabla f(x^*), u \rangle \geq 0 \; \forall \; u \in T_{\mathcal{F}}(x^*)$. In other words the gradient of f at x^* belongs to the normal cone of $\mathcal{F}$ at x^* i.e., $\nabla f(x^*) \in N_{\mathcal{F}}(x^*)$.*

[5] $T_S(x)$ is the tangent cone of set S at point x.

Lecture 14

- We continued our discussion on optimality conditions. We commented that theorem 13.0.10 is very generic and applicable to any differentiable convex program[1]. It talks about optimality characterization for any feasible solution, including the cases where the optimality happens at the boundary of the feasibility set. Though the nec.&suff. condition in the theorem is generic, it involves computation of tangent/normal cones. Instead of repeating this exercise for every problem we thought of identifying some specific CPs where we will pre-compute these cones and re-write the nec.&suff. in a more usable way. The idea is that whenever we encounter an optimization problem in practice/research, we will try to put it in one of these standard forms and then characterizing optimality conditions will be simple[2].

- Before proceeding to the various special cases where we re-write theorem 13.0.10, for the sake of completeness, we also conjunctured how the conditions will look like for a convex program which is not differentiable (needlessly to say, sub-differentiability at all feasible points is assumed): given a feasible solution x^* of (P), x^* is an optimal solution of (P) if and only if for every $u \in T_{\mathcal{F}}(x^*)$ we can identify a $\nabla f(x^*) \in \partial f(x^*)$ such that $\langle \nabla f(x^*), u \rangle \geq 0$. It was easy to prove the sufficiency part, which follows from the sub-gradient inequality. We said that the proof for the necessity part follows from a characterization of support function of the sub-differential set in terms of one-sided directional derivative. Interested students are requested to read section 23 from Rockafellar [1996] and specifically theorem 23.4 in it.

- Here are the special cases:

[1] A convex program (13.1) is said to be differentiable iff its objective function is differentiable everywhere in the feasibility set.

[2] We stressed on the importance of the fact that the results will apply only to the standard forms we assume. So if we make a mistake in posing the problem at hand into these standard forms, then we are doomed.

1. Suppose we know $x^* \in int(\mathcal{F})$, then the conditions are simply $\nabla f(x^*) = 0$ (because the normal cone has the zero vector alone). This is a special case of theorem 13.0.9. So optimality characterization is simple. Interestingly, in this case we also know that $f(x^*) = -f^*(0)$. So the optimal value of (P) can be simply got by looking at f^* (and seemingly x^* needed not be computed for obtaining the optimal value).

   In the following special cases we assume that the domain is an open set. In particular, it can be the entire vector space[3].

2. Suppose we know that (P)'s objective is a linear function and the constraint set is a polyhedron. Such a convex program is known as a Linear Program (LP):

$$\begin{aligned} &\min_{x \in X} && \langle c, x \rangle, \\ &\text{s.t.} && \langle a_i, x \rangle \leq b_i, \ \forall \ i = 1, \ldots, m. \end{aligned} \tag{14.1}$$

   We then have the following theorem (derived in MidSem and follows from theorem 13.0.10:

   Theorem 14.0.11. *Suppose x^* is a feasible solution of the LP (14.1) i.e., $x^* \in X, \langle a_i, x^* \rangle \leq b_i, \ \forall \ i = 1, \ldots, m$ and the constraints indexed by the set $I \subset \{1, \ldots, m\}$ are active at x^*, i.e., $\langle a_i, x^* \rangle = b_i, \ \forall \ i \in I$. Then we have x^* is an optimal solution if and only if there exist $\lambda_i \geq 0 \ \forall \ i \in I$ such that $c + \sum_{i \in I} \lambda_i a_i = 0$. Also, the optimal value $f(x^*) = -\sum_{i \in I} \lambda_i b_i$.*

3. Suppose we have a convex program (P) with the constraint set as a polyhedron, then we call it a Polyhedrally-Constrained Convex Program (PCCP):

$$\begin{aligned} &\min_{x \in X} && f(x), \\ &\text{s.t.} && \langle a_i, x \rangle \leq b_i, \ \forall \ i = 1, \ldots, m. \end{aligned} \tag{14.2}$$

   It is then easy to write down the following theorem (from above case):

   Theorem 14.0.12. *Suppose x^* is a feasible solution of (14.2) i.e., $x^* \in X, \langle a_i, x^* \rangle \leq b_i, \ \forall \ i = 1, \ldots, m$ and the constraints indexed by the set $I \subset \{1, \ldots, m\}$ are active at x^*, i.e., $\langle a_i, x^* \rangle = b_i, \ \forall \ i \in I$. Then we have x^* is an optimal solution if and only if there exist $\lambda_i \geq 0 \ \forall \ i \in I$ such that $\nabla f(x^*) + \sum_{i \in I} \lambda_i a_i = 0$. Also, the optimal value $f(x^*) = -f^*(-\sum_{i \in I} \lambda_i a_i) - \sum_{i \in I} \lambda_i b_i$.*

[3] Alternatively, we could have assumed domain is any convex set, however all the theorems 14.0.11-14.0.13 need to have an assumption that $x^* \in int(X)$

If the objective in a PCCP is a quadratic function, then it is called as a Quadratic Program.

4. Suppose we have a convex program (P) which can be written in the following form:

$$\begin{aligned} &\min_{x \in X} && f(x), \\ &\text{s.t} && g_i(x) \leq 0 \ \forall \ i = 1, \ldots, m. \end{aligned} \tag{14.3}$$

Such a convex program is called an Ordinary Convex Program (OCP). An OCP is said to be differentiable iff all $f, g_1, \ldots, g_m$ are differentiable in the feasibility set, which is $\{x \mid x \in X, g_i(x) \leq 0 \ \forall \ i = 1, \ldots, m\}$. In this case we have the following theorem:

Theorem 14.0.13. *Let (P) be a differentiable OCP (refer equation 14.3) such that atleast one of the following is true:*

(a) There exists $\bar{x} \in X$ such that $g_i(\bar{x}) < 0$ for all i where g_i is not an affine function.

(b) Let $x^* \in \mathcal{F}$ and let I be the index set of non-affine constraints which are active at x^*. Assume that the set $\{\nabla g_i(x^*) \mid i \in I\}$ is a linearly independent set.

The first condition is called the Slater's condition. An OCP satisfying this condition is known as a regular CP. Note that the second condition is not a condition on the OCP, but rather a condition on the feasible point x^*. A feasible point satisfying this constraint is called a regular point.

We have that: x^* is an optimal solution if and only if there exists a $\lambda^* = [\lambda_1^* \ldots \lambda_m^*]^\top$ such that (x^*, λ^*) satisfy the Karush-Kuhn-Tucker (KKT) conditions. A point (x^*, λ^*) is said to satisfy KKT conditions for a differentiable OCP (14.3) iff:

(a) $x^* \in X$ and $\lambda^* \geq 0$ (domain conditions[4]).

(b) $g_i(x^*) \leq 0 \ \forall \ i = 1, \ldots, m$ (feasibility conditions).

(c) $\lambda_i^* g_i(x^*) = 0 \ \forall \ i = 1, \ldots, m$ (complementary-slackness conditions).

(d) $\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0$ (gradient condition).

Such a point is called as a KKT point for the given problem.

In other words, for differentiable OCPs, the KKT conditions are necessary and sufficient for optimality of a regular point. Moreover, for

[4] Again, if we dont assume X is open, then we must have $x^* \in int(X)$

regular differentiable OCPs, the KKT conditions are necessary and sufficient for optimality of *any* point.

Firstly, note that this theorem includes the previous theorems on LPs and CPCs as special cases. We gave a simple sketch of proof of this theorem: we proved this theorem by starting from theorem 13.0.10 and re-writing the Normal-cone in terms of gradients of g_i. We will complete the proof in the next lecture and discuss some examples where we actually obtain closed form solutions for some non-trivial real-world problems using the above KKT characterization of optimal solutions.

Lecture 15

- Using KKT conditions we showed that $\|\cdot\|_{M^{-1}}$ is the dual norm of $\|\cdot\|_M$, where $M \succ 0$.
- Using KKT conditions we showed that the optimal solution of the following problem:

$$\begin{aligned} \min_{p \in \mathbb{R}^{n++}} \quad & \textstyle\sum_{i=1}^n p_i \log(p_i) - p^\top x, \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^n p_i = 1, \end{aligned}$$

  which we said is frequently encountered in probability theory, is $p_i = \frac{e^{x_i}}{\sum_{i=1}^n e^{x_i}}$.
- The above examples illustrated the utility of KKT conditions. We then went on to complete the proof of theorem 14.0.13. This proof is given at the end of this notes in hand-written appendix-1.
- Most of the optimization algorithms try to use the optimality characterizations we learnt till now to arrive at the optimal solution. So the conditions not only help us to arrive at analytical solutions when they exist (like in above problems), but also help in devising numerical algorithms for (approximately) solving them. So these conditions form the core theory behind optimization.
- It is very useful to know that KKT characterization is useful for non-convex problems too. Infact, KKT conditions are useful in characterizing local optimality (which we didnt formally define, but must be familiar to alteast some of you). For more details please non-linear optimization books by Fletcher or Bertsekas etc. Hopefully while derving KKT conditions again from Lagrangian duality, we might realize some of these results.

Lecture 16

- With the example of finding minimum path between cities we concluded that three main properties for a "dual" optimization problem are desirable. It is desirable that the dual is such that:

  1. Its objective value at any feasible point is a lower bound for the given (primal) problem — Principle of Weak Duality.
  2. Its optimal value must be that of the primal — Principle of Strong Duality.
  3. Its should be a computationally feasible problem; ideally a convex program.

- We listed some schemes for obtaining duals for which all 3 desirable properties hold:

  1. LP duality scheme — works on LPs only, but straight-forward to write (even programmable).
  2. Conic duality scheme — works for Conic programs (yet to be defined formally), also straight-forward to write (programmable).
  3. Lagrangian duality scheme — works for (regular) OCPs and hence very generic, may not always be easy to write.
  4. Fenchel/Conjugate duality scheme — very generic scheme, probably covers all above and may not always be easy to write.

- We noted two generic methodologies/strategies, which all of these schemes implement: one based on lower bounding the primal objective and the other starting with a function of both primal and dual variables. We said that LP and Conic duality schemes implement the former whereas the Lagrangian and Conjugate duality schemes implement the latter strategy.

- We then began with LP duality. Looking at the structure of optimal value given by KKT conditions for LP (14.1), we focussed our attention on the following function of λ: $g(\lambda) = -\sum_{i=1}^{m} b_i\lambda_i$ when $\lambda \geq 0$, $-\sum_{i=1}^{m} \lambda_i a_i = c$. We know that if it so happens that LP is solvable with x^* as solution, then there exists a λ^* satisfying the conditions above and additionally the complementary-slackness conditions such that $g(\lambda^*)$ is the optimal value of the original LP (14.1).

- We then analyzed the behaviour of g wrt LP's (14.1) objective: if x is any feasible point of LP (14.1) and λ is such that $\lambda \geq 0$, $-\sum_{i=1}^{m} \lambda_i a_i = c$, then the following was immediate: $\langle c, x\rangle = \langle -\sum_{i=1}^{m} \lambda_i a_i, x\rangle = -\sum_{i=1}^{m} \lambda_i \langle a_i, x\rangle \geq -\sum_{i=1}^{m} \lambda_i b_i = g(\lambda)$. Hence optimal value of LP (14.1) will always be greater than or equal to the optimal value of this problem:

$$\begin{aligned} &\max_{y\in\mathbb{R}^m} && -\textstyle\sum_{i=1}^{m} b_i y_i, \\ &\text{s.t.} && y \geq 0,\ -\textstyle\sum_{i=1}^{m} y_i a_i = c. \end{aligned} \tag{16.1}$$

  In other words, weak duality holds between the primal LP (14.1) and the dual problem (16.1) i.e., (16.1) is a weak dual of (14.1).

- More importantly, as noted above, if the primal LP is solvable, then by KKT conditions we know that there exists a λ^* belonging to feasibility set of the dual (16.1) such that $g(\lambda^*)$ is the optimal value of the original LP. In other words, strong duality holds i.e., (16.1) is a strong dual[1] of (14.1).

- Interestingly, the dual (16.1) can also be expressed as a LP and hence is a convex program.

- Infact we showed[2] some more interesting results about LP duality which are summarized in theorem 1.2.2 in Nemirovski [2005].

- We noted as with duality is the case always, there are trade-offs in using the primal or dual form. Many times it is used to an advantage in designing algorithms. Infact we said that many of state-of-the-art algorithms employ both forms.

- We extended same methodology and showed that the following is a (weak,

[1] Usually when we say D is a dual of P, we actually mean that D is a weak as well as strong dual of P.

[2] Only the part that dual of dual is primal (symmetry) we did'nt explicitly show in lecture; nevertheless can be taken by reader as an exercise.

strong) dual of PCCP (14.2):

$$\begin{array}{rll} \max\limits_{y \in \mathbb{R}^m} & -f^*(-\sum_{i=1}^m y_i a_i) - \sum_{i=1}^m b_i y_i, \\ \text{s.t.} & y \geq 0. \end{array} \tag{16.2}$$

- Again the dual (16.2) is convex and it reduces to the LP dual if f is linear.
- Mandatory reading: Please read entire section 1.2 of Nemirovski [2005] in detail. This gives an alternate way of deriving LP-duality using theorems on alternative.

Lecture 17

- We noted that the dual might also be used in for arriving at optimality conditions (basically gives a way of re-writing the optimality conditions).
  - In the context of LP or PCCP duality, given a pair $(\bar{x}, \bar{y})$ such that $\bar{x}$ is primal feasible and $\bar{y}$ is dual feasible, the KKT conditions give that: $(\bar{x}, \bar{y})$ form an optimal primal-dual pair if and only if the duality gap i.e., the difference between primal and dual objective values at $(\bar{x}, \bar{y})$ is zero. In case of LPs, the duality gap $\langle c, x \rangle + b^\top y$ must be zero for optimality. Similarly, we can talk about the KKT-gap, which is the "gap" in satisfying the constraints remaining to be satisfied in the KKT set beyond those implied by the primal-dual feasibility. For instance, in case of LPs, the KKT gap $\bar{y}^\top (b - A^\top \bar{x})$ (where, A is a matrix with columns as a_i and b is column vector with entries as b_i) must be zero for optimality. Corresponding expressions for PCCPs can be worked out. Though the expressions are different, the idea behind duality gap and KKT-gap remains the same and can be written down for all the duality schemes which are going to encounter in this course.
  - We noted that these duality gap and KKT gap conditions are not only helpful in case of algorithms which maintain both the primal and dual iterates, but also may be helpful in case of algorithms solving either of them: for e.g., if we have an iterative algorithm maintaining only the primal iterates, and further we assume that the dual variables can be computed using the KKT conditions, then the duality/KKT gap can indeed be computed and the optimality of the iterate can be verified.
- We then focussed on a particular optimization problem, which was fundamental to the notion of duality: the problem of separating two closed convex sets. We recalled we encountered this problem first in separation theorem, which is the key theorem behind all duality. We also recalled the strategy we used to arrive at a separating hyperplane: by finding points closest in

the two sets and taking the perpendicular bisector of the line joining them, which infact gives the maximal separator between the two sets.

- We wanted to verify if the duality notion we developed till now in optimization problems atleast gives back this fundamental duality: i.e., maximally separating two (closed, convex) sets is same as minimizing distance between them. In view of this, we began writing the maximal separation problem as a mathematical program, infact as a PCCP:

$$\begin{aligned} \max_{a \in \mathbb{R}^n} \quad & S_C(-a) - S_D(a), \\ \text{s.t.} \quad & \|a\|_\infty \leq 1. \end{aligned}$$

  Here, C, D are the two closed convex sets which need to be maximally separated by the unknown a such that their separation, which is equal to $S_C(-a) - S_D(a)$, is maximized. Here, S_A denotes the support function of A. The bound $\|a\|_\infty \leq 1$ is simply put to get a non-trivial solution.

- Using the PCCP duality result we showed in last class, we derived the following dual:

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^n, \mu \in \mathbb{R}^n} \quad & \textstyle\sum_{i=1}^n \lambda_i + \mu_i, \\ \text{s.t.} \quad & \lambda \geq 0,\ \mu \geq 0,\ \lambda - \mu \in C - D \end{aligned}$$

  which is equal to the following problem:

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & \|z\|_1, \\ \text{s.t.} \quad & z \in C - D, \end{aligned}$$

  which is indeed the problem of minimizing distance between the two given sets. Infact, we noted that the last equality provides a generic way of representing $\|z\|_1$ in terms of linear function and constraints.

- We later commented that the 1-norm in dual is appearing as we started with its dual, the ∞ norm, in the primal. With this we guessed the general form of the duality, which we may prove at a later stage (with current duality schemes we must be limited to polyhedral constraint sets).

- In the next lecture, we will look at a big class of CPs, called as conic programs. Interestingly, the duality in their case turns out to be as simple and elegant as in LPs.

Lecture 18

- We began by motivating the definition of conic programs in Euclidean spaces:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x, \\ \text{s.t.} \quad & b - A^\top x \in K, \end{aligned} \tag{18.1}$$

  where, K is a cone and A is an $n \times m$ matrix with columns as a_i and b is a $m \times 1$ vector. It was evident that every conic program is a convex program.

- We noted few examples of programs we encountered in this course which actually were conic programs. Infact, if K is chosen to be the first quadrant of $\mathbb{R}^n$, then, the above conic program is exactly the LP (in the standard primal form). We commented that we will later on see that many convex programs, including many we already encountered, are actually conic programs.

- Nevertheless we began by deriving a dual for a conic program and ended up with this conic duality theorem[1] (please refer to theorem 1.7.1 in Nemirovski [2005] and the corresponding proof):

  Theorem 18.0.14. *Consider the problem:*

$$\begin{aligned} \max_{y \in \mathbb{R}^m} \quad & -b^\top y, \\ \text{s.t.} \quad & y \in K^*, -Ay = c. \end{aligned} \tag{18.2}$$

  We have the following results:

  1. *(18.2) is a weak dual of (18.1).*
  2. *If primal (18.1) is bounded, then (18.2) is bounded (and if (18.2) feasible, then (18.2) is also solvable).*

[1] We proved everything in lecture except the symmetry, which can be taken as an exercise.

3. *If (18.2) is bounded, then primal (18.1) is bounded (and if (18.1) feasible, then (18.1) is also solvable provided K is a closed cone).*
4. *If K has non-empty interior and the primal is bounded and the conic program satisfies a mild regularity condition (like Slater's condition) that there exists an $\bar{x} \ni b - A^\top \bar{x} \in int(K)$, then infact (18.2) is bounded, feasible, solvable and it is a strong dual of the primal.*
5. *If K is pointed (then we know that K^* has non-empty interior) and if the dual is bounded and a mild regularity condition that there exists a $\bar{y} \in int(K^*) \ni -A\bar{y} = c$, then infact the primal (18.1) is bounded, feasible, solvable and the optimal value of the primal is same as that of the dual.*
6. *If K is a closed cone, then the dual of the dual is the primal (symmetry).*

To summarize, if we start with a primal conic program with a good cone K, i.e., K is closed, pointed and has non-empty interior, then whenever either of the problems are bounded, then the other is solvable and strong duality holds. Also, symmetry holds i.e., the dual of the dual is exactly the primal.

- In particular, this theorem is the same as the LP-duality theorem if K were the first quadrant cone. However this theorem is more general and still as easy to write as in the LP case (provided K^* is known. In special case, K is self-dual, then things are easier: $K^* = K$).
- We then defined conic-quadratic program (CQ) or second-order cone program (SOCP) as a conic program with K as cross-product of finite number of ice-cream cones (note that this K is a good cone and duality theorem applies; moreover, it is self-dual! So writing the dual is a trivial job). We commented that many convex programs in Euclidean spaces can be written as CQs/SOCPs and hence are very useful. We will study SOCPs in the next lecture. Then we will study conic programs in generic (finite-dim) Hilbert spaces — a particular case leads to Semi-definite programs, which are very generic and yet easyto handle.
- Mandatory reading: Entire Lecture-1 in Nemirovski [2005].

Lecture 19

- We defined Second Order Cone Programs (SOCPs) or Conic Quadratic Programs (CQs) as those conic programs with the cone K as a Cartesian-product of finite number of ice-cream cones i.e., $K = L^{p_1} \times L^{p_2} \times \ldots \times L^{p_q}$, where $L^p = \{x \in \mathbb{R}^p \mid \|[x_1 \ldots x_{p-1}]^\top\|_2 \leq x_p\}$. Assuming $b = \begin{bmatrix} d_1 \\ \nu_1 \\ \vdots \\ d_q \\ \nu_q \end{bmatrix}$ and $A^\top = \begin{bmatrix} -D_1^\top \\ -f_1^\top \\ \vdots \\ -D_q^\top \\ -f_q^\top \end{bmatrix}$, where each $D_i \in \mathbb{R}^{n \times p_i - 1}, d_i \in \mathbb{R}^{p_i - 1}, f_i \in \mathbb{R}^n, \nu_i \in \mathbb{R}$, we can re-write any SOCP as:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x, \\ \text{s.t.} \quad & \|D_i^\top x + d_i\|_2 \leq f_i^\top x + \nu_i, \ \forall\ i = 1, \ldots, q. \end{aligned} \tag{19.1}$$

  Infact, many textbooks choose to define SOCPs as OCPs of the above (19.1) form. The constraints of the form those in an SOCP are called as Second Order Cone Constraints (SOCs) or Conic Quadratic Constraints (CQCs).

- Since K here is a good cone (and infact self-dual), the conic-duality theorem 18.0.14 can be applied and we showed that the dual turns out to be:

$$\begin{aligned} \max_{y_i \in \mathbb{R}^{p_i - 1}, \lambda_i \in \mathbb{R},\ \forall\ i = 1, \ldots, q} \quad & \textstyle\sum_{i=1}^q -d_i^\top y_i - \nu_i \lambda_i, \\ \text{s.t.} \quad & \|y_i\|_2 \leq \lambda_i, \ \forall\ i = 1, \ldots, q, \ \textstyle\sum_{i=1}^q D_i y_i + \lambda_i f_i = c. \end{aligned} \tag{19.2}$$

- Notice that the dual is again easy to write down (can write a computer program to do it). Also, it is again an SOCP. So it is self-dual (like a LP).

- We took examples of three programs we encountered in this course and actually realized that they can be written as SOCPs (two of them appear in the problem sets). One of them was the dual norm problem: $\min_{x \in \mathbb{R}^n} x^\top y$, s.t.$\|x\|_M \leq 1$, where $M \succ 0$. We first wrote down this program as a SOCP (using the EVD of M) and then the dual of this SOCP. Interestingly, the dual of this SOCP was trivial to solve and essentially boiled down to the constant number $\|y\|_{M^{-1}}$, which is indeed the optimal value of the original problem! We concluded that sometimes the dual may turn out to be a simple problem, which can be analytically solved; whereas this solution may not be obvious for the primal.

- We later on pointed out that there are many sets, and functions whose epigraphs or level-sets may be represented using some finite number of SOCs. We stressed on the importance of familiarizing ourselves with all such "good" functions/sets. For e.g., $f(x) = \|x\|_M$ is such a function. A huge list of such good functions/sets appear in lecture-2 in Nemirovski [2005] and in Lobo et al. [1998]. Such a knowledge will help in identifying SOC-nature of the programs you might encounter in research.

- A problem-set problem also shows that any Quadratically Constrained Quadratic Program (QCQP) can be written as an SOCP.

- We mentioned that `cvx`[1] is a software, which identifies many "good" functions/sets and allows to use them in describing optimization problems. It internally converts them to an SOCP and calls a suitable SOCP solver. Infact it is capable of doing this with all named CPs we encounter in this course. Examples of some SOCP solvers are `Mosek`[2] and `SeDuMi`[3]. Unlike `cvx`, these require the user to describe programs only in their standard form i.e., either (19.1) or (19.2). Infact all of these toolboxes can also handle SDPs, which form a big class of convex programs, which we will define later.

- We then went on to see how conic programs in arbitrary (finite-dim) Hilbert spaces look like. We wanted to define them generically where the cone K may not lie on the space of the variables. In such a case we wanted to look at how one can linearly transform vectors from one space to another.

[1] `http://cvxr.com/cvx/`
[2] `http://www.mosek.com/`
[3] `http://sedumi.ie.lehigh.edu/`

- We showed that every linear function $f : V \mapsto W$ can be represented using a $m \times n$ matrix, say M_f,where m is dim. of W and n is dim. of V. Infact we showed that if $\hat{v}$ is the representation of $v \in V$ using a basis of V and $\hat{w}_v$ is the representation of $w_v = f(v) \in W$ using a basis of W, then $\hat{w}_v = M_f \hat{v}$ (note that this is in sync with our notion of linear functions from $\mathbb{R}^n$ to $\mathbb{R}^m$). Moreover, if the basis of V and W were orthonormal, then we also showed that for any $w \in W$ and $v \in V$, we have $\langle w, f(v) \rangle_W = \hat{w}^\top M_f \hat{v} = \hat{v}^\top M_f^\top \hat{w} = \langle v, f^\top(w) \rangle_V$, where $f^\top$ is that linear function induced by the matrix $M_f^\top$ and is called as adjoint of f. In case $f = f^\top$ (analogous to $M = M^\top$), we say that f is self-adjoint[4].

- In the next lecture we will use our knowledge about linear functions/transformations and define/study Semi-Definite Programs (SDPs).

- Mandatory reading: Entire Lecture-2 in Nemirovski [2005]. Read Lobo et al. [1998], which is a seminal work by Boyd and his team. This can be downloaded from: `http://stanford.edu/~boyd/papers/socp.html`.

[4]Because symmetric matrices have EVD, and every linear function has this associated matrix, we can now talk about EVD of any self-adjoint linear function f. This is the natural generalization of EVD.

Lecture 20

- We started with the definition of a conic program in generic (finite-dim) Hilbert spaces:

$$\begin{aligned} &\min_{x \in V} && \langle c, x \rangle_V, \\ &\text{s.t} && b -_W \mathcal{A}^\top(x) \in K \subset W, \end{aligned} \tag{20.1}$$

  where $\mathcal{V} = (V, +_V, \cdot_V, \langle\rangle_V)$ is the Hilbert space in which the variable lives, $\mathcal{W} = (W, +_W, \cdot_W, \langle\rangle_W)$ is the Hilbert space in which K, a cone, lies; $c \in V$, $b \in W$ and $\mathcal{A}^\top : V \mapsto W$ is a linear function, whose adjoint is $\mathcal{A} : W \mapsto V$.

- It was left as an exercise to verify that a completely analogous conic-duality theorem 18.0.14 can be written in this case and the (strong) dual is:

$$\begin{aligned} &\max_{y \in W} && -\langle b, y \rangle_W, \\ &\text{s.t.} && y \in K^*, \mathcal{A}(y) +_V c = 0_V, \end{aligned} \tag{20.2}$$

  where 0_V is the identity element for $\mathcal{V}$.

- We looked at a special case of the conic program with $\mathcal{V}$ as the usual n-dimensional Euclidean space, $\mathcal{W}$ as the usual space of symmetric matrices of size m, K as the psd cone and $\mathcal{A}^\top(x) = \sum_{i=1}^n x_i A_i$, where each $A_i \in S^m$, the set of all symmetric matrices of size m. This is defined as an Semi-Definite Program (SDP):

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} && c^\top x, \\ &\text{s.t} && B - \textstyle\sum_{i=1}^n x_i A_i \succeq 0, \end{aligned} \tag{20.3}$$

  here $B \in S^m$.

- The constraint of the form of that in an SDP (20.3) is called a Linear Matrix Inequality (LMI).

- The dual of SDP is immediate to write provided we determine the adjoint $\mathcal{A}$. We showed that $\mathcal{A}(Y) = [\langle A_1, Y\rangle_F \dots \langle A_n, Y\rangle_F]^\top$. Moreover the psd cone is self-dual in space of symmetric matrices. Hence the dual of an SDP (20.3) is:

$$\begin{aligned} \max_{Y \in S^m} \quad & -\langle B, Y\rangle_F, \\ \text{s.t.} \quad & Y \succeq 0, -\langle A_i, Y\rangle_F = c_i, \ \forall\ i = 1, \dots, n \end{aligned} \tag{20.4}$$

- We took two examples of problems i) finding min. radius ellipse enclosing given set of points ii) finding a "simple" elliptic separator between two sets of points. We gave a rough sketch of posing these problems as SDPs. The key tricks employed were the clever use of Schur-complement lemma and that every finite number of LMIs can be written as a single LMI.

- We also showed that[1] every SOC can be expressed as an LMI: $\|Dx + d\|_2 \leq f^\top x + \nu \Leftrightarrow (Dx+d)^\top(Dx+d) \leq (f^\top x + \mu)^2, f^\top x + \mu \geq 0 \Leftrightarrow \begin{bmatrix} (f^\top x + \mu)I & (Dx+d) \\ (Dx+d)^\top & f^\top x + \mu \end{bmatrix} \succeq 0$, which can be expressed as an LMI. Hence SOCPs can be written as SDPs and in some sense form the "biggest" well studied set of convex programs.

- As mentioned earlier, `Mosek`, `SeDuMi` and `cvx` all handle SDPs.

- We then focussed our attention to OCPs and their duality. We recalled the connection of dual and optimality conditions and said that the KKT conditions directly if employed in obtaining dual, then dual will involve the optimal solution of x^*, which then makes the dual unusable. Instead, we wanted to re-write the problem in such form, where optimality conditions (which ofcourse will be equivalent to KKT) may be more elegant. This is what motivated us to write OCP (14.3) as an un-constrained problem: $\min_{x\in X} f(x) + I_C(x)$, where C is the constraint set of the OCP i.e., $C = \{x \mid g_i(x) \leq 0\}$.

- Moving towards duality, we re-wrote the indicator function in its "dual form"[2]: $I_C(x) = \max_{\lambda \geq 0} \sum_i \lambda_i g_i(x)$. Thus the OCP 14.3 is equal to:

$$\min_{x\in X} \max_{\lambda \geq 0} L(x, \lambda),$$

  where $L(x,\lambda) = f(x) + \sum_{i=1}^n \lambda_i g_i(x)$. This is called the Lagrangian function of the OCP (14.3). The domain of the Lagrangian function is $X \times \mathbb{R}^m_+$.

[1] This proof is what I think Sami was suggesting. Somehow I failed to realize :(

[2] Rather than its conjugate form, which is the support function of C.

- It was then easy to see that OCP (14.3) is greater than or equal to:

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^m} \quad & \underline{L}(\lambda), \\ \text{s.t.} \quad & \lambda \geq 0, \end{aligned} \tag{20.5}$$

  where $\underline{L}(\lambda) = \min_{x \in X} L(x, \lambda)$ (usually called as the Langrangian dual function). Hence (20.5) is a weak dual of (14.3).

- Interestingly, the weak duality holds even if f, g_i in OCP (14.3) are not convex! Moreover, even in this case, $-\underline{L}$, which is the point-wise maximum of affine functions, is a convex function! In other words, the weak dual (20.5) is ALWAYS a convex program; irrespective of convexity of the original ordinary program. This result is very helpful in obtaining lower bounds (sometimes tight) on the primal problem's optimal value.

- In case we assume (14.3) is indeed OCP, then we additionally get that $L(x, \lambda)$ is convex in x. In this case it the optimization problem involved in obtaining $\underline{L}(\lambda) = \min_{x \in X} L(x, \lambda)$ is convex and hence "easy" to solve. Also, we have that: $\overline{L}(x) = \max_{\lambda \geq 0} L(x, \lambda)$ is convex (note that the primal problem is same as $\min_{x \in X} \overline{L}(x)$).

- We visualized how the Lagrangian function for an OCP looks like and guessed how the Lagrangian function of a perhaps non-convex ordinary program may look like.

- In the next lecture, under some mild regularity conditions we will show that the problem in (20.5) is infact a (strong) dual. This duality scheme is called as Lagrangian duality.

- Mandatory reading: Entire Lecture-3 in Nemirovski [2005]. Read L. Vandenberghe and S. Boyd [1999], which is a seminal work by Boyd and his team. This can be downloaded from: `http://www.stanford.edu/~boyd/papers/pdf/sdp-apps.pdf`. Vandenberghe and Boyd [1996] is also worth reading.

Lecture 21

- We set out for proving that (20.5) [denoted by (D) henceforth in this lecture] is a (strong) dual of the primal problem (14.3) [denoted by (P) henceforth in this lecture]; perhaps under some regularity conditions.

- Looking at the similarity in the expression of the Lagrangian function and the gradient condition in the KKT conditions, we started by assuming the primal (P) is a differentiable regular ocp. In such a case, we know that x^* is an optimal solution of (P) if and only if there exists a λ^* such that (x^*, λ^*) is a KKT point. The gradient condition in the KKT conditions gives us that $\underline{L}(\lambda^*) = L(x^*, \lambda^*) = f(x^*)$. Together with the weak duality proved in previous lecture, this gives that λ^* is infact an optimal solution for (D) and moreover, strong duality holds!

- For us this is the easiest way of proving Lagrangian duality. It however turns out that the statement of Lagrange duality does not even require assumptions of solvability of the primal (boundedness is enough) and (P) need not be differentiable. Here is the generic statement which can be made:

  Theorem 21.0.15. *Let (P) given by (14.3) be an ordinary program (not necessarily convex). Then:*

  1. *(D) is a weak dual of (P). i.e., (P) $\geq$ (D).*
  2. *if (P) is regular bounded ocp, then (D) is a (strong) dual of (P) i.e., (P)=(D). Further, (D) is solvable*[1].

[1] Note that (P) may be unsolvable. Infact, Theorem D.2.3 in Nemirovski [2005] shows that x^*, λ^* are optimal solutions of (P) and (D) respectively if and only if (x^*, λ^*) is a saddle point of the Lagrangian function. However this characterization of optimality may not be very useful to us and hence we did not cover this theorem in lecture. Many books, including Nemirovski [2005], infact prove KKT conditions from this saddle point characterization of optimality.

3. *if (P) is regular bounded and differentiable ocp, then in addition to the above results, we have that x^*, λ^* are primal, dual optimal solutions (respectively) if and only if (x^*, λ^*) is a KKT point.*

- We outlined the proof of this theorem and not surprisingly, the key idea is to employ the separation theorem. The detailed proof is in Theorem D.2.2 in Nemirovski [2005].

- We took the example of a strictly convex QP and computed its Lagrange dual. The dual was another strictly convex QP! In process, we noted that if the QP was not strictly convex i.e., Hessian of the objective is psd, then there is no "easy" way of eliminating the primal variables to write the dual. We later on said the the most cumbersome part of writing the Lagrange dual is infact computing $\underline{L}$. And ofcourse, if the problem is not differentiable, then computing $\underline{L}$ is mostly likely cumbersome.

- We then generalized the example by taking a PCCP and wrote down its Lagrangian dual and the dual was same as that given by PCCP duality we derived earlier. This was not surprising as we know that all the different duality schemes, though outwardly look different, indeed use the separation theorem and hence cannot be fundamentally different.

- We then took example of an SOCP and tried to write its dual. Again, computing $\underline{L}$ was the problem and there seemed to be no way of writing the SOCP as a differential OCP (in which case computing $\underline{L}$ perhaps is easy). This example motivated us to combine the "nice" aspects of Lagrange and conic duality, and perhaps write down the "Lagrange-conic duality scheme"[2]. We were able to easily establish weak duality:

$$\begin{array}{ll} \max\limits_{y \in \mathbb{R}^m} & -f^*(-\sum_{i=1}^m y_i a_i) - \sum_{i=1}^m b_i y_i, \\ \text{s.t.} & y \in K^*. \end{array} \tag{21.1}$$

is the weak dual of

$$\begin{array}{ll} \min\limits_{x \in X} & f(x), \\ \text{s.t.} & b - \mathcal{A}(x) \in K, \end{array} \tag{21.2}$$

- Note the analogy with the PCCP duality. We further said that we can talk about Lagrange-conic duality in problem of following form[3]:

$$\begin{array}{ll} \min\limits_{x \in X} & f(x), \\ \text{s.t.} & [-g_1(x) \ldots - g_m(x)]^\top \in K, \end{array} \tag{21.3}$$

[2] This is studied under generalized inequalities in chp5. of Boyd and Vandenberghe [2004].

[3] Bonus marks to students who write down the duality theorem in this case and prove it.

- Now the idea is to "push" the non-differentiabilities in the problem at hand into the cone and have f, g_i differentiable; so that computing $\underline{L}$ is easy. Further if K is self-dual, then $K^* = K$. Thus we expect that once the problem is posed in (21.2/21.3) form (with differentiable f, g_i), writing down the dual will be easy.

- We concluded the lecture with a short discussion of non-convex problems, in whose case it is desirable to obtain a lower bound on the optimal value of the (non-convex) minimization problem at hand. One way is to write down a dual of it. We noted that both conic (with K being arbitrary set) and Lagrangian duality schemes (or the Lagrange-conic duality) can be applied to obtain weak duals. The nice thing is that both ways we get a convex problem as the dual, which can be solved "easily" to obtain a lower bound on primal's optimal value. Another way is[4], to write the given problem in a form with convex objective (this can always be done) and then relax the non-convex constraints. i.e., take a convex set which covers the non-convex feasibility set. The resulting problem is convex and solving it will again give a lower bound on primal's objective. Now, in general, in both these methods the tightness of the bound is not known. However there are a very interesting class of problems for which bounds are known. Please read sections 3.4 and 3.5 in Nemirovski [2005] to know more about them. Most interesting is the case of a generic QCQP with a single constraint (it so happens that the eigen-value-problem can be posed as one in this form). In this case both the methods (Lagrangian dual and Shor's relaxation scheme) give a problem whose optimal value is equal to the original problem! This also gives an example of a non-convex program whose dual is strong!

- Mandatory reading: Entire Appendix section D in Nemirovski [2005]. Entire chp.5 in Boyd and Vandenberghe [2004] (except may be sensitivity analysis).

[4]Suggested by Sami

Lecture 22

- All along the course we mentioned that convex programs are "easy" to solve. However we never clearly mentioned what easy was and what results about solving them exist. To this end, we presented Theorem 4.1.2 in Nemirovski [2005][1]. The book presents a constructive proof of this theorem by presenting the ellipsoid method which can solve a generic convex program (with not "too difficult" objective and feasibility set). In loose words, the statement is that a generic convex program with reasonable objective and feasibility set can be solved in polynomial time.

- Though this result gives immense relief, the ellipsoidal method is rarely employed in practice. This is because it is too generic to achieve "fast" convergence on the specific problem or class of problems at hand.

- We then began a theoretical study of optimization algorithms. The algorithms we study are numerical procedures, and are iterative in nature. Hence one usually is interested in knowing how many iterations does the algorithm need to reach an ϵ-accurate solution: $\hat{x}$ is ϵ-accurate if either i) $||\hat{x} - x^*|| \leq \epsilon$ or ii) $|f(\hat{x}) - f^*| \leq \epsilon$, where f^*, x^* are optimal value and solution respectively.

- Since it does not make sense to compare algorithms (for number of iterations for convergence) for a single instance of a mathematical program, the idea is to talk about bounds on the number of iterations for convergence on a pre-specified class of programs.

- When it comes to comparing algorithms (in terms of no. iterations), it makes sense to do so across algorithms with same per-iteration "information-cost". For example, the ellipsoid method uses a gradient and feasibility-set separator oracles at every iteration. So it makes sense to compare it with other methods which also have this "information-cost" (i.e., same oracles).

[1] Please study the proof of this theorem from the book.

Finally, an algorithm is said to be optimal if it has better convergence rate (i.e., no.iterations for ϵ-accuracy) than any other method in its class (i.e., any other method with same information-cost). We mentioned that such analysis of algorithms is common in optimization theory and is called as Information-based complexity[2] (rather than computational complexity).

- Now that the setting is clear we began analyzing the performance of a well-known method gradient-method, given by (1.2.9) in Nesterov [2004], on the class of Smooth un-constrained convex programs, which are convex programs with feasibility set as the entire Hilbert space and the objective function is a differentiable everywhere convex function with the gradient function (the function which takes a point and returns the gradient of the objective function at that point) being Lipschitz continuous i.e., $||\nabla f(x) - \nabla f(y)|| \leq L||x-y||, \forall\ x, y \in V$ for some $L \in \mathbb{R},\ L > 0$.

- The key step in gradient method is its updated formula given by $x_{k+1} = x_k + s_k \nabla f(x_k)$. We discussed two motivations for this update rule i) Any direction which makes angle greater than 90^o with the gradient is a direction of descent i.e., the function locally decreases along that direction. Hence the idea is to take appropriate steps of size s_k along the negative gradient direction, which gives the "steepest descent" ii) consider the problem of minimizing the first order approximation of the function around x_k i.e., $f(x_k) + \langle \nabla f(x_k), (x - x_k) \rangle$ rather than the function itself. Now, since this approximation is valid locally, lets try to minimize this approximation while also insisting on not moving far off from the current iterate x_k. Thus one way of obtaining x_{k+1} is:

$$x_{k+1} = \text{argmin}_{x \in V} f(x_k) + \langle \nabla f(x_k), (x - x_k) \rangle + \frac{1}{2s_k}||x - x_k||^2 = x_k - s_k \nabla f(x_k)$$

  Intutively, both explainations suggest that the choice of step-size might be crucial for convergence. This is indeed the case.

- We noted that many variants of gradient method exist with various schemes for choosing step sizes (where convergence can be gauranteed). The simplest of them is a constant step size of $\frac{1}{L}$.

- We repeated the proof of corollary 2.1.2 in Nesterov [2004]. In summary, the result is that with the gradient method ($s_k = \frac{1}{L}$), we have: $f(x_k) - f^* \leq \frac{2L||x_0 - x^*||^2}{k+4}$. This results shows many things: i) firstly it shows that, the method when started with *any* x_0, asymptotically converges to the optimal value f^*. i.e., as $k \to \infty$, $f(x_k) \to f^*$. If any algorithm satisfies such a

[2] Refer section 1.1 in Nesterov [2004].

property, then it is known as globally convergent. Hence, we have that the gradient method with constant step size $\frac{1}{L}$ is globally convergent on the class of smooth unconstrained convex programs with Lipschitz const. L ii) To reach ϵ-accuracy, we need atmost $k \propto O(\frac{1}{\epsilon})$ iterations. This is called the rate of convergence. Note the keyword atmost: it means that there might be specific problems where we get better convergence rate iii) higher the L, "complex" is the objective and the bound shows that the number iterations for ϵ-accuracy grows with it iv) similarly, as the initial guess x_0's distance from optimal solution increases, more iterations are needed.

- We then asked the question is the gradient method optimal? It turns out that in view of theorem 2.1.7 in Nesterov [2004], all methods with information-cost being first-order i.e., first-order (gradient returning) oracles are assumed to be available, and $x_k \in x_o + LIN(\{\nabla f(x_0, \ldots, \nabla f(x_{k-1}))\})$, will *atleast* need $O(\frac{1}{\sqrt{\epsilon}})$ iterations. Hence the gradient method, which does fall under this class of algorithms, may not be optimal.

- Infact, Nesterov presented a modified gradient method, which is as easy to code as the gradient method, but has convergence rate atmost $O(\frac{1}{\sqrt{\epsilon}})$, showing that it is an optimal method! Infact it turns out that it is hard to prove such a good bound (under global convergence) with other traditional methods like conjugate gradient or bundle methods or even second-order methods like the Newton method. We refer to this as the Nesterov method; refer eqn. (2.2.8) in Nesterov [2004].

- Mandatory reading: Sections 2.1 and 2.2 in Nesterov [2004].

Lecture 23

- We began by listing out practical difficulties in implementing the (simple looking) gradient-method with constant step-size, whose convergence was established in the previous lecture[1]. Firstly, the gradient might not have an analytical expression, in which case one has to resort to numerical techniques. An example is the Lagrangian dual function[2]. Now convergence with exact gradient is proven; what about such an approximate gradient? There are some results along this direction (but beyond scope of this course). In any case, if the gradient computation itself is computationally challenging, then we are doomed. Secondly, L (the Lipschitz const.) is not given in applications and more often than not, it is difficult to estimate it. Also, if a pessimistic estimate of L is used, then the convergence in practice might be terribly slow.

- Fortunately, the second problem has an easy way out: there are many other "easier" schemes for step-sizes for which convergence can be shown: i) Armijo's rule (refer eqn. 1.11 and corresponding section on pg.29 in Bertsekas [1999]) ii) diminishing step-sizes i.e., $s_k \to 0, \sum_{k=1}^{\infty} s_k \to \infty$ (for eg. $s_k = \frac{1}{k}$). Proposition 1.2.1 and 1.2.4 in Bertsekas [1999] show the convergence with these step-size schemes, infact on generic (non-convex) smooth programs[3]. In the convex smooth case, essentially both theorems prove global convergence under the condition that the sequence $\{x_k\}$ is bounded or an objective with bounded level sets. From results it clear that in the convex smooth case, the diminishing step sizes is an easy option[4]; while the Armijo rule is best to try in a generic setting. Note that, however, Armijo rule assumes a function value oracle (i.e., zero-order oracle in addition to the

[1] Similar comments hold for the Nesterov's method

[2] Interested students look up Danskin's theorem

[3] Infact, prop. 1.2.1 doesn't even insist on smoothness. Differentiability seems enough!

[4] This doesn't mean that always this gives fastest convergence. For different specific instances of programs, different step-size schemes may perform better.

first-order one) being present. Such variants in step-sizes are also studied for the Nesterov's method: Refer sec. 10.2.1 in Nemirovski [1995].

- We then studied variants of gradient method, where the direction chosen at each step is not negative of gradient, but other valid descent direction. One way of representing updates with such methods is: $x_{k+1} = x_k - s_k D_k \nabla f(x_k)$, where $D_k \succ 0$ is a pd matrix. Such methods are called as Descent methods[5]. We also showed the connection between descent methods and methods which locally approximate the function by second-order terms (instead of first-order terms in the gradient method):

$$x_{k+1} = \text{argmin}_{x \in \mathbb{R}^n} f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2s_k}(x - x_k)^\top D_k^{-1}(x - x_k) + \frac{\gamma_k}{2}\|x - x_k\|_2^2,$$

  which[6] is equal to $x_{k+1} = x_k - \left(s_k^{-1} D_k^{-1} + \gamma_k I\right)^{-1} \nabla f(x_k)$ and exactly gives back the Descent method when $\gamma_k = 0$. Gopi pointed out that though $\gamma_k = 0$ initially seems bad for convergence (as second-order approximation holds locally), the other quadratic term infact acts as the regularizer[7] instead of $\|x - x_k\|_2^2$.

- Two special cases of the Descent methods are particularly interesting: i) Newton method: here, $\gamma_k = 0, s_k = 1 \ \forall \ k$ and $D_k = H_f(x_k)^{-1}$, the inverse of the Hessian of the objective f at x_k. Hence this method applies to cases where Hessian is pd at all x_k rather than psd (psd alone is guaranteed for convex functions) ii) Levenberg-Marquardt (LM) method (refer sec.5.2 in Fletcher [2000]): here, $\gamma_k > 0, s_k = 1 \ \forall \ k$ and $D_k = H_f(x_k)^{-1}$. Note that this method applies to any convex unconstrained program as the Hessian is always psd and hence Hessian plus a positive diagonal matrix is always pd (and hence invertible).

- We mentioned that convergence with both variants can be proved (in case of Newton with suitable assumptions about invertibility of Hessian). However, what is more interesting is an extremely fast convergence of these methods, when x_0 is "close" to an optimal solution. This analysis is called local-convergence analysis (as opposed to the global one we saw earlier). To this end we presented theorem 1.2.5 in Nesterov [2004]. This shows that $k \propto \log(\frac{1}{\epsilon})$ is the (local) rate of convergence of the Newton's method — which is extremely fast when compared to the ones we saw till now. While this is encouraging, proving such rates for global convergence is difficult!

[5] Infact, the propositions in Bertsekas book noted book prove convergence with descent methods.

[6] Note that with $D_k = H_f(x_k)^{-1}$, the inverse of the Hessian of the objective f at x_k, this method uses the Taylor expansion for the second-order term.

[7] This is true, and hence the Bertsekas book manages to prove convergence of Descent methods.

- Namit and Sami then gave an interpretation to the LM method: with high γ_k it is like gradient method and with low γ_k it is more like Newton and hence it is kind of a hybrid method. For faster rate of convergence one might want lower γ_k, provided the Hessian $+ \gamma_k I$ stays pd (and not ill-conditioned); this explains the funny γ_k scheme in LM. To summarize, LM method tries to achieve a mix of the goodness in gradient method (that of global convergence) and the Newton method (extremely fast local convergence) — and hence is preferred in practice than the Newton's method, unless it is known that the Hessian is pd.

- Now comes the question whether to prefer LM-method or the Nesterov method. Ofcourse these methods are not comparable. Empirically also there is no clear winner. It is best to try both and then zero-in.

- In the next lecture we will present schemes which avoid the Hessian computation and inversion in Newton/LM methods and try to achieve comparable rates. These are known as Quasi-Newton methods.

- Mandatory reading: Entire chapter 1 in Nesterov [2004]. Recommended reading: sections 9.1-9.5 in Boyd and Vandenberghe [2004]

Lecture 24

- The key overhead in Newton/LM methods is the Hessian computation and its inversion at each iteration (which is $O(n^3)$). If we have a descent method with D_k as not the inverse of Hessian, but rather some-other pd matrix, then Hessian computation/inversion can be avoided. Ofcourse we have to still ensure we take a pd matrix and one which atleast matches gradients. The methods which achieve this goal are called as Quasi-Newton (or Variable Metric) methods, which not only update the iterate, but also the matrix D_k using a suitable "simple" update rule.

- We started by saying that we will have a simple 1-rank update $D_k = D_{k-1} + \alpha u u^\top$ (where $\alpha > 0$). This update rule ensures that all D_k are pd, provided D_0 is pd. Later on we can repeat the exercise with rank-2 update: $D_k = D_{k-1} + \alpha u u^\top + \beta v v^\top$ $(\alpha, \beta > 0)$.

- The key constraint imposed on D is that the corresponding objective function second-order approximation's gradient will match that of the objective function itself. i.e., gradient of $f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top D_k^{-1} (x - x_k)$ at x_k and x_{k-1} equals $\nabla f(x_k)$ and $\nabla f(x_{k-1})$ respectively. This leads to the Quasi-Newton rule: $D_k(\nabla f(x_k) - \nabla f(x_{k-1})) = x_k - x_{k-1}$. One way of choosing u and α with this rule being satisfied is given on pg. 41 in Nesterov [2004] (under the name rank-1 correction scheme).

- The rank-2 update schemes on D_k and D_k^{-1} respectively are also in pg. 41 under the names DFP and BFGS respectively. As mentioned there, the BFGS scheme is the most preferred.

- Quasi-newton methods have local convergence rates faster than gradient method (exact form again appears in pg. 41). However wrt. global convergence no bounds with rates better than gradient method are proven. These methods require $O(n^2)$ computations in each iteration.

- We then moved on to conjugate-gradient methods which are variants with $O(n)$ computations in each iteration and infact are directly comparable with Nesterov/gradient method (first-order black box). The nice thing about these methods is i) n-step convergence in case of quadratic functions ii) local convergence rate faster than gradient method.

- Since these methods were originally developed for quadratic (unconstrained) minimization, we outlined the method with suitable motivations for this case: $f(x) = \frac{1}{2}x^\top Ax + b^\top x + c$, where $A \succ 0$. From EVD knowledge we know that there exist a basis (infact, orthogonal) $\{d_1, \ldots, d_n\}$ such that they form a A-orthogonal (conjugate-direction) set i.e., $d_i^\top A d_j = 0$ whenever $i \neq j$ and not zero otherwise. It was easy to see that once these directions are known, then the optimal solution $x^* = \sum_{i=1}^n \lambda_i^* d_i$ can be computed using: $\lambda_i^* = \frac{b^\top d_i}{d_i^\top A d_i}$ (i.e., in $O(n^2)$ operations).

- Now the idea is to come up with these directions iteratively using $O(n)$ operations at each iteration such that movement along these directions at each step guarantees optimality at the end of n steps (this will again be $O(n^2)$ computations). One way to do this is by taking steps $x_k = x_{k-1} + s_{k-1}d_{k-1}$ and update $d_k = -\nabla f(x_{k-1}) + \beta_{k-1}d_{k-1}$. Now the idea is to choose s_k and β_k such that i) the directions d_k indeed form a A-orthogonal set ii) the directions d_k are descent directions iii) x_{n+1} is optimal solution i.e., $\nabla f(x_{n+1})^\top d_i = 0$ for all $i = 1, \ldots, n$.

- We will now show that it is enough to choose s_{k-1} appropriately to satisfy iii): $x_k = x_{k-1} + s_{k-1}d_{k-1}$

  $\Rightarrow \nabla f(x_k) = \nabla f(x_{k-1}) + s_{k-1}Ad_{k-1}(\because \text{ multiply by } A)$
  $\Rightarrow d_{k-1}^\top \nabla f(x_k) = d_{k-1}^\top \nabla f(x_{k-1}) + s_{k-1}d_{k-1}^\top Ad_{k-1}(\because \text{ dot product with } d_{k-1})$
  $\Rightarrow d_{k-1}^\top \nabla f(x_k) = 0(\text{If clever choice of } s_{k-1} = \frac{-d_{k-1}^\top \nabla f(x_{k-1})}{d_{k-1}^\top Ad_{k-1}})$

  It remains to show $d_i^\top f(x_k) = 0 \ \forall \ i = 1, \ldots, k-2$. This can be shown provided d_i indeed form a A-orthogonal set: taking dot product on both sides with d_{k-2} in above equations, we have $d_{k-2}^\top \nabla f(x_k) = d_{k-2}^\top \nabla f(x_{k-1}) + s_{k-1}d_{k-2}^\top Ad_{k-1} = 0$, because, the first term in the sum is zero by choice of s_{k-2} and the second term in sum is zero again because of A-orthogonality (or conjugacy). Now because of the additive nature of the update formula, we also have: $\nabla f(x_k) = \nabla f(x_{k-i}) + \sum_{j=1}^i s_{k-j}Ad_{k-j}$. Again repeating the above exercise we have: $d_i^\top \nabla f(x_k) = 0 \ \forall \ i = 1, \ldots, k-1$. In particular, this is true with $k = n+1$, and hence iii) is satisfied.

- We will now show that it is enough to choose β_{k-1} appropriately to satisfy i):

$$d_k = -\nabla f(x_{k-1}) + \beta_{k-1} d_{k-1} \Rightarrow d_{k-1}^\top A d_k = -d_{k-1}^\top A \nabla f(x_{k-1}) + \beta_{k-1} d_{k-1}^\top A d_{k-1}$$
$$\Rightarrow d_{k-1}^\top A d_k = 0 (\text{with clever choice of } \beta_{k-1} = \frac{d_{k-1}^\top A \nabla f(x_{k-1})}{d_{k-1}^\top A d_{k-1}}$$

  Analogous to the iterate update, since the direction update is also additive in nature, we obtain i). A slight re-writing of the formula for β gives the Fletcher-Rieves formula in pg. 45 in Nesterov [2004].

- In summary, the conjugate-gradient method with the above choice of s_k and Fletcher-Rieves direction update, converges in n steps to the optimal solution in case of quadratic (unconstrained) minimization problems.

- Now this method "as is" will not work for generic smooth functions. The conjugate-gradient method for smooth functions is presented in pg. 45 in Nesterov [2004]. Note the key changes: i) the choice of s_k is by exact line search ii) Polak-Ribbiere direction update is the most preferred iii) the interpretation for d_k is lost after n iterations (as there can be only n LI vectors in n-dim space). Hence, after every n conjugate-gradient steps, one sets $\beta = 0$ i.e., takes a gradient-descent step. And this procedure is repeated. With this one gets local convergence rate which is better than gradient method and is near to Newton method's rate for low-dimensional problems (exact expression again appears on pg. 45). Global convergence rate again is not proven to be better than the gradient method.

- This completed our discussion about algorithms for smooth-unconstrained problems. We gave some tips for choosing the appropriate algorithm:

  1. Ofcourse if nothing is known about a problem, except that it is unconstrained i.e., not even convexity is established, then gradient method is the choice.
  2. If additionally it is known that it is convex, then one can try `cvx`.
  3. Additionally if it is smooth, then Nesterov's method is the first choice and the best bet.
  4. If for some reason Nesterov's method is "slow" in your problem, and the problem at hand is high-dimensional (where $O(n^2)$ or beyond operations are infeasible), then one can try conjugate-gradient with Polak-Riebbiere direction update or gradient method with Armijo rule etc.
  5. If the problem is moderate dimension ($O(n^2)$ operations is ok), then one may try Quasi-Newton method with BFGS update.

6. Additionally if Hessian information is available and the problem dimension is small ($O(n^3)$ operations are fine), then LM-method can be a choice.

- Also, on the class of smooth unconstrained problems no known algorithm, including the ones using second-order information, beats the global convergence rate of the Nesterov's method! However on specific problems, or on special class of smooth problems more interesting results exist. For e.g.:

  1. For strictly convex quadratic ones (which are not ill-conditioned), conjugate-gradient is the best (converges in n steps).
  2. For strongly convex functions (refer section 2.1.3 in Nesterov [2004]), the convergence rate achieved by gradient method itself is comparable to local convergence rate of Newton's method (refer theorem 2.1.15 in Nesterov [2004]).
  3. For self-concordant convex functions, the Newton method (actually modified Newton method called damped Newton method, which has a step-size parameter) globally converges with an extremely fast rate (refer section 9.6 in Boyd and Vandenberghe [2004] and/or section 4.1 in Nesterov [2004])!

- We then began discussing algorithms for non-smooth unconstrained minimization i.e., we assume a sub-gradient oracle. We gave an intuition why this class is more difficult to optimize — because the negative of sub-gradient direction may actually be a direction of increase of the function (even locally)! Hence in a simple algorithm like sub-gradient method: $x_k = x_{k-1} + s_{k-1}\nabla f(x_{k-1})$, where ∇ represents sub-gradient, is NOT a descent method.

- We then defined the sub-gradient method (equation 3.2.8 in Nesterov [2004]) and repeated theorem 3.2.2. in the same book, which proves asympotic convergence of the method (in some sense).

- We noted that the sub-gradient descent is "optimal" on the class of non-smooth unconstrained problems (refer to theorem 3.2.1 in Nesterov [2004]).

- Mandatory reading: Sections 1.3.1, 1.3.2 and 3.2.3 in Nesterov [2004]. Recommended reading: sections 9.6-9.7 in Boyd and Vandenberghe [2004].

Lecture 25

- We briefly reviewed the sub-gradient method, which is in some sense optimal. We noted that, analogous to the case of smooth problems, there might be special class of non-smooth (uncons.) problems which might be easier to solve: for eg. objectives of the form $f(x) = \max_{i=1,\ldots,m} f_i(x)$ with each $f_i(x)$ smooth. Note that f may not be differentiable. However we can compute f's sub-gradient[1]. In this case a method which converges in $O(\frac{1}{\sqrt{\epsilon}})$ can be devised. The details are in section 2.3 in Nesterov [2004].

- We then moved on to solving smooth convex programs: $\min_{x \in C} f(x)$, where C is closed convex set and f is smooth convex function. In this case we wrote down the following simple update rule, which is analogous to that in the gradient method: $x_{k+1} = \text{argmin}_{x \in C} f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2s_k}\|x - x_k\|_2^2 = \text{argmin}_{x \in C} \|x - (x_k - s_k \nabla f(x_k))\|_2^2 = P_C(x_k - s_k \nabla f(x_k))$. In simple words, take the usual gradient step and projected it back to the constraint set. This is called the Projected Gradient Method (PGM)

- It was easy to show convergence once we proved $\|y - z\|_2^2 \geq \|y - P_C(y)\|_2^2 + \|P_C(y) - z\|_2^2$, for all $y \notin C$ and $z \in C$ (refer Lemma 3.1.5 in Nesterov [2004]). This gives that $\|P_C(x_k - s_k \nabla f(x_k)) - x^*\| \leq \|x_k - s_k \nabla f(x_k) - x^*\|$, where x^*, the optimal solution. After this step, once can repeat the proof[2] of convergence in the unconstrained case and obtain the same convergence rate. Though the convergence rate is the same, it is coming at a cost of an extra projection step at each iteration.

- It is easy to see that PGM is well-suited to problems where projection onto the constraint set is easy: for eg. a box, ball, simplex, ellipse etc.

- As in case of unconstrained problems, PGM is not optimal. There exist

[1] A problem set problem characterizes the sub-differential set.
[2] A lengthy proof (but more insightful one) of the same is in Nesterov [2004].

projection versions of Nesterov method which are indeed optimal (refer eqn. 2.2.19 in Nesterov [2004]).

- Now what if the constraint set is described by functional inequalities i.e., the case of an ordinary convex program? It so happens that in this case one can avoid the projections and do the following: define $\bar{g}(x) = \max_{i=1,\ldots,m} g_i(x)$ and if $\bar{g}(x_k) < 0$, then take a (sub)gradient step; else take a step which deceases $\bar{g}$ i.e., go in negative direction of $\nabla\bar{g}(x_k)$ (i.e., (sub)gradient step). A slightly modified version of this method with appropriate step-choice is in eqn. 3.2.13 in Nesterov [2004]. By theorem 3.2.3, it is guaranteed to converge and infact, the rate is same as that in the unconstrained problem case! Note that unlike the projected gradient method, this method has no "big" overhead compared to unconstrained case. This shows that in some sense constrained optimization is as easy/difficult as the unconstrained case.

- Some algorithms pose/approximate the constrained problem as an unconstrained one and then solve using unconstrained problem techniques. The interior-point algorithms (refer chapter 4 in Nesterov [2004] or chapter 11 in Boyd and Vandenberghe [2004]), which are at the heart of the standard toolboxes, employ this very idea. The alternate way is to look at the Lagrange dual and if it is smooth, then since the constrained set is first quadrant, one can employ projected gradient method. However with this one can recover optimal solution for dual and the optimal value. One may need to do extra work to arrive at optimal solution for the primal. We gave an example of a convex QP with equality constraints to illustrate various points. Please read chapter 10 in Boyd and Vandenberghe [2004] for such special cases.

Bibliography

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

R. Fletcher. *Practical Methods of Optimization*. Wiley, 2000.

L. Vandenberghe and S. Boyd. Applications of Semidefinite Programming. *Applied Numerical Mathematics*, 29:283–299, 1999.

M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of Second-Order Cone Programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

A. Nemirovski. Lectures On Modern Convex Optimization. Available freely at `www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf`, 2005.

A. Nemirovski. Efficient Methods in Convex Programming. Available at `http://www2.isye.gatech.edu/~nemirovs/Lect_EMCO.pdf`, 1995.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

Sheldon Axler. *Linear Algebra Done Right*. Springer-Verlag, 1997.

L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review*, 38(1): 49–95, 1996.

Appendix I

28/Sep/11

We'll prove theorem 14.0.13 with assumption (a) in the following and leave the assumption (b) case to the reader.

Given: (P) is a regular differentiable OCP (refer eqn. 14.3)

TST x^* is optimal soln. $\iff \exists \lambda^* = \begin{bmatrix} \lambda_1^* \\ \lambda_m^* \end{bmatrix}$ such that (x^*, λ^*) is a KKT point.

Proof: From theorem 13.0.10, we have that:

x^* is optimal soln. $\iff x^* \in X$, $g_i(x^*) \leq 0 \ \forall i$,
$\nabla f(x^*) \in N_{\mathcal{F}}(x^*)$

where $\mathcal{F} = \{x / g_i(x) \leq 0 \ \forall i, x \in X\}$

The idea is to characterize $N_{\mathcal{F}}(x^*)$ in terms of $\nabla g_i(x^*)$.
To this end consider the following claim:

claim ①: If I is the index set where $g_i(x^*) = 0 \ \forall i \in I$,

then $N_{\mathcal{F}}(x^*) = \text{conichull}(\{-\nabla g_i(x^*) \mid i \in I\})$

We will prove this claim lately, but before that assuming it to be true we have:

x^* is optimal soln. $\iff$ $x^* \in X$, $g_i(x^*) \le 0 \ \forall i$

& $\nabla f(x^*) \in \text{Conical}\left(\{-\nabla g_i(x^*) / i \in I\}\right)$

$\iff$ $x^* \in X$, $g_i(x^*) \le 0 \ \forall i$,

$\exists \lambda_i^* \ge 0 \ \forall i \in I \ni \nabla f(x^*) = -\sum_{i \in I} \lambda_i \nabla g_i(x^*)$

$\iff$ $x^* \in X$, $g_i(x^*) \le 0 \ \forall i$,

$\exists \lambda_i^* \ge 0 \ \forall i \in I \ni \nabla f(x^*) + \sum_{i \in I} \lambda_i \nabla g_i(x^*) = 0$

$\iff$ $x^* \in X$, $g_i(x^*) \le 0 \ \forall i$,

$\exists \lambda^* \ge 0 \ni \lambda_i^* g_i(x^*) = 0 \ \forall i$,

$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) = 0$

$\iff$ (x^*, λ^*) is a KKT point.

Now realize that I doesn't depend on I, hence the proof is valid for any I and hence theorem 14.0.13 is proved.

Proof of Claim ①: The idea is to look at $T_J(x^*)$

First we note that: $T_J(x^*) = T_{\{x / g_i(x) \le 0 \ \forall i \in J, \ x \in X\}}(x^*) = \bigcap_{i=1}^{m} T_{\{x / g_i(x) \le 0\}}(x^*) \cap T_X(x^*)$

The above follows from the following claim:

Claim ②: If A & B are two sets and x^* is a common point on them, then $T_{A\cap B}(x^*) = T_A(x^*) \cap T_B(x^*)$

(Take this as an exercise)

Since X is an open set $T_X(x^*) = V$, the entire space.

Hence for all $i \notin I$ we have $T_{\{x/g_i(x)\le 0\}}(x^*) = V$.

↓

This is because $x^* \in \text{int}(\{x/g_i(x)\le 0\})$ where $i \notin I$

Claim ③ i.e. if $g_i(x^*) < 0 \Rightarrow x^* \in \text{int}(\{x/g_i(x)\le 0\})$

We will prove this ~~later~~ in the problem set.

Now assuming this we have: $T_{\mathcal{F}}(x^*) = \bigcap_{i\in I} T_{\{x/g_i(x)\le 0\}}(x^*)$

Now lets write down a description for $T_{\{x/g_i(x)\le 0\}}(x^*)$ in terms of $\nabla g_i(x^*)$ for $i \in I$.

To this end consider the following claim:

Claim ④: ~~Let x^* be such that $g_i(x^*) = 0$~~ (Here $i \in I$)

(a) $u \in T_{\{x/g_i(x)\le 0\}}(x^*) \Rightarrow \langle \nabla g_i(x^*), u\rangle \le 0$

(b) $u \in T_{\{x/g_i(x)\le 0\}}(x^*) \Leftarrow \langle \nabla g_i(x^*), u\rangle < 0$

Claim ④ follows from the defn. of directional derivative and has been done in the lecture.

~~How (4)b gives that the~~

Now suppose g_i $(i \in I)$ is affine function in that case we know that $T_{\{x / g_i(x) \le 0\}}(x^*)$ is a half space & by claim (4)b

we ~~get~~ get that $T_{\{x / g_i(x) \le 0\}}(x^*)$ ~~due~~ is all u satisfying $\langle \nabla g_i(x^*), u \rangle \le 0$ — (R1)

~~$T_{\{x / g_i(x) \le 0\}}(x^*) = \{u / \langle \nabla g_i(x^*), u \rangle \le 0\}$.~~

Now if g_i $(i \in I)$ is non-affine, then we assured Slater's condition, so obviously $\nabla g_i(x^*) \neq 0$.

So $T_{\{x / g_i(x) \le 0\}}(x^*)$ will at least contain the open half-space described by all u such that: $\langle \nabla g_i(x^*), u \rangle < 0$ (by claim (4)b)

Also by claim (4)a we know $u \in T_{\{x / g_i(x) \le 0\}}(x^*) \Rightarrow \langle \nabla g_i(x^*), u \rangle \ge 0$

Hence we get that $cl\left(T_{\{x / g_i(x) \le 0\}}(x^*)\right)$ is all u satisfying $\langle \nabla g_i(x^*), u \rangle \le 0$ — (R2)

By (R1) & (R2) in any case of g_i we have ↗

Now in the Normal cone ~~is~~ of a cone & its double & its interior are all same. So ~~by~~ we ~~get~~ instead look at $cl(T_F(x^*))$.

To summarize, we have:

$$cl(T_F(x^*)) = \bigcap_{i \in I} cl\left(T^{(x^*)}_{\{x / g_i(x) \le 0\}}\right)$$

$$= \text{set of all } u \text{ satisfying} \quad (\because \text{by } (R2))$$

$$\langle \nabla g_i(x), u \rangle \le 0 \quad \forall \ i \in I$$

Since primal description of dual cone is -ve of dual description of primal we get that:

$$N_F(x^*) = \text{conichull}\left(\{-\nabla g_i(x) \mid i \in I\}\right)$$

Hence claim (1) proved.

—— Appendix (1) end ——