Convex Optimization Problems.

- Optimization problem in standard form

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad h_i(x) = 0, \quad i = 1, \ldots, p
\end{align*}
\]

- \( x \in \mathbb{R}^n \) is the optimization variable
- \( f_0 : \mathbb{R}^n \to \mathbb{R} \) is the objective or cost function
- \( f_i : \mathbb{R}^n \to \mathbb{R}, \quad i = 1, \ldots, m \), are the inequality constraints
- \( h_i : \mathbb{R}^n \to \mathbb{R} \) are the equality constraints functions

- Optimal Value:

\[
P^* = \inf \{ f_0(x) | f_i(x) \leq 0, \quad i = 1, \ldots, m, \quad h_i(x) = 0, \quad i = 1, \ldots, p \}
\]

- \( P^* = \infty \) if the problem is infeasible
- \( P^* = -\infty \) if the problem is unbounded below

Note: The above has nothing to do w/ convexity. Applies for all optimization problems.
- **Feasible point**: $x$ is feasible if $x \in \text{dom } f_0$ and it satisfies all the constraints.

- A feasible point $x$ is optimal if $f_0(x) = P^*$; $X_{opt}$ is the set of optimal points.

- **Local optimality**: $x$ is locally optimal if there is an $R > 0$ such that $x$ is optimal for

\[
\begin{align*}
\text{minimize} & \quad f_0(z) \\
\text{subject to} & \quad f_i(z) \leq 0, \quad i=1,\ldots,m, \\
& \quad g_i(z) = 0, \quad i=1,\ldots,p \\
& \quad \|z - x\|_2 \leq R
\end{align*}
\]
Feasibility Problem

\[
\text{find} \quad x \\
\text{Subject to} \quad f_i(x) \leq 0, \quad i=1, \ldots, m \\
\quad h_i(x) = 0, \quad i=1, \ldots, p
\]

Can be considered as a special case of the general form \( P^* \) with \( f_0(x) = 0 \).

\[
\begin{align*}
\text{Minimize} & \quad f_0(x) \\
\text{Subject to} & \quad f_i(x) \leq 0, \quad i=1, \ldots, m \\
& \quad h_i(x) = 0, \quad i=1, \ldots, p
\end{align*}
\]

\( P^* = 0 \) if \( x \) is feasible; any feasible \( x \) is optimal. 
\( P^* = \infty \) if constraints are violated.

Note: Solving a feasibility problem can be very hard.

Convex Optimization Problems.

\[
\begin{align*}
& \text{minimize} \quad f_0(x) \\
& \text{subject to} \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad a_i^T x = b_i, \quad i = 1, \ldots, p
\end{align*}
\]

- \( f_0(x), f_1(x), \ldots, f_m(x) \) are convex; equality constraints are affine.

- Often written as

\[
\begin{align*}
& \text{minimize} \quad f_0(x) \\
& \text{subject to} \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad A x = b
\end{align*}
\]

- By the "intersection property" of convex sets, the feasible set of a convex optimization problem is convex.

- Loosely written as

\[
\begin{align*}
& \text{minimize} \quad f_0(x) \\
& \text{subject to} \quad x \in X
\end{align*}
\]

where \( f_0 \) is convex, and \( X \) is convex.

It is okay ... but the above is not necessarily easy to solve.
The way of writing down the problem matters:

Example: minimize \( \min_x f_0(x) = x_1^2 + x_2^2 \)

subject to \( \begin{align*}
    & f_1(x) = x_1 / (1 + x_2^2) \leq 0 \quad (1a) \\
    & h_1(x) = (x_1 + x_2)^2 = 0. \quad (1b)
\end{align*} \)

- \( f_0 \) is convex, the feasible set \( \mathcal{X} = \{ (x_1, x_2) | x_1 + x_2 \leq 0 \} \) is also convex.
- The problem is NOT convex.

Equivalent Problem:

\[
\begin{align*}
    \min_x & \quad x_1^2 + x_2^2 \\
    \text{subject to} & \quad x_1 \leq 0 \quad (2a) \\
    & \quad x_1 + x_2 = 0 \quad (2b)
\end{align*}
\]

is a convex optimization problem.

Remark: Even (1a)-(1b) and (2a)-(2b) represent the same convex set \( \mathcal{X} \), it does not mean the corresponding optimization problems are both convex.
• Local and global minima

Claim: Any locally optimal point of a convex problem is (globally) optimal.

Proof: Suppose \( x \) is locally optimal, but there exists a \( y \) with \( f_0(y) < f_0(x) \).

\( x \) is locally optimal means there is a \( R > 0 \) such that \( z \) feasible, \( \| z - x \|_2 \leq R \Rightarrow f_0(z) \geq f_0(x) \).

Consider \( z = \theta y + (1 - \theta) x \) with \( 0 < \theta < \frac{1}{2} \).

- One can see that \( \| y - x \|_2 > R \), so \( 0 < \theta < \frac{1}{2} \).
- In addition, \( \| z - x \|_2 = \| \theta y + (1 - \theta) x - x \|_2 = \| \theta y - \theta x \|_2 = \theta \| y - x \|_2 = \frac{\theta R}{\| y - x \|_2} \).

Therefore, \( f_0(z) = f_0(\theta y + (1 - \theta) x) \leq \theta f_0(y) + (1 - \theta) f_0(x) \leq f_0(x) \) by convexity (by \( \theta \in (0, \frac{1}{2}) \)).

Which contradicts the assumption that \( x \) is locally optimal.
- Optimal criteria for differentiable $f_0$.

Concavity: $f_0(y) \geq f_0(x) + \nabla f(x)^T (y-x)$
for all feasible $y$.

Then $x$ is optimal iff $\nabla f(x)^T (y-x) \geq 0$ for all feasible $y$.

- Geometrically:

- Unconstrained problems: minimize $f_0(x)$.

$x$ is optimal if and only if $x \in \text{dom } f_0$ and $\nabla f(x) = 0$.

E.g. minimize $\|Y - Ax\|_2^2 \implies x^* = (A^TA)^{-1}A^Ty$.

Q: What is the $f(x)$ here?
Proof of optimality.  (iff)

First, suppose \( x \in X \) and satisfies  \( f_0(y) \geq f_0(x) + \nabla f_0(x)^T (y - x) \).

by convexity.

Then \( f_0(y) > f_0(x) \) since the second term is nonnegative.

Which means that \( f_0(x) \) is the optimal value for \( x \in X \).

Conversely, suppose \( x \) is optimal, but the condition (4.21) does not hold, i.e., for some \( y \in X \) we have
\[ \nabla f_0(x)^T (y - x) < 0. \]

Consider the point \( z(t) = ty + (-t)x \), where \( t \in [0, 1) \) is a parameter. \( z(t) \) is feasible if \( y \) and \( x \) are feasible.

\[ \frac{d}{dt} f_0(z(t)) \bigg|_{t=0} = \nabla f_0(x)^T (y - x) < 0. \]

it means when \( t \) is very small the direction from \( x \) to \( z(t) \) is a descending direction, which means \( f_0(z(t)) < f_0(x) \),
and thus \( x \) is not optimal. This is a contradiction.
### Equality Constrained Problems

Minimize $f_0(x)$

Subject to $Ax = b$.

$x$ is optimal if and only if there exists a $v$ such that $x \in \text{dom } f_0$, $Ax = b$, $\nabla f_0(x) + A^T v = 0$. 

[This is what we call the KKT condition.]

**Proof:** For optimal $x$, we have

$\nabla f_0(x)^T (y - x) \geq 0$ for all $Ay = b$. (*)

Since $x$ is feasible, $Ax = b$ and thus $A(y - x) = 0$. This means that $y - x = z \in N(A)$.

(*) can then be re-written as

$\nabla f_0(x)^T z \geq 0$ for all $z \in N(A)$

$\Rightarrow \nabla f_0(x) \perp N(A)$

Using the fact that $N(A) = R(A^T)^\perp$, the optimality condition can be expressed as.

$\nabla f_0(x) \perp N(A) \Leftrightarrow \nabla f_0(x) \in R(A^T)$

$\Rightarrow \nabla f_0(x) + A^T v = 0$.

Q.E.D.

**Note:**

$N(A) = \{ x | Ax = 0 \}$.

$R(A^T) = \{ x | x = A^T \theta \}$.

Let $x \in N(A)$ and $y \in R(A^T)$

$x^T y = x^T A^T \theta = (x A^T) \theta = 0$.

$
\Rightarrow N(A) \perp R(A^T)$
• Equivalent Convex Problems.

(Has a lot to do with finding a solver...)

- Two problems are equivalent if the solution of one is readily obtained from the solution of another.

- Eliminating equality constraints

\[
\begin{align*}
\text{minimize } & f_0(x) \\
\text{subject to } & f_i(x) \leq 0 \\
& Ax = b.
\end{align*}
\]

is equivalent to

\[
\begin{align*}
\text{minimize } & f_0(Fz + x_0) \\
\text{subject to } & f_i(Fz + x_0) \leq 0.
\end{align*}
\]

where \( R(F) = N(A) \)

\[ Ax = b \iff x = Fz + x_0. \]

- Adding Slack variables.

\[
\begin{align*}
\text{minimize } & f_0(x, s) \\
\text{subject to } & f_i(x) + s_i = 0 \\
& Ax = b \\
& s_i \geq 0.
\end{align*}
\]

[To prove equivalence, show two directions]

(Quite useful for designing ADMM algo.)

- Minimizing over some variables.

\[
\begin{align*}
\text{minimize } & f_1(x_1, x_2) & \text{and } & \text{minimize } f_2(x_1) \\
\text{over } & x_1, x_2 & & \text{over } x_1
\end{align*}
\]

\[
\text{e.g. Minimize } \|y - Ax\|_2^2 \\
\text{subject to } & y \geq 0
\]

where \( f_2(x_1) = \min_{x_2} f(x_1, x_2) \)
Examples of Convex Optimization

- Linear Program (LP)

\[ \text{minimize } \mathbf{c}^T \mathbf{x} + d \]
\[ \text{subject to } \mathbf{G} \mathbf{x} \leq \mathbf{h}, \quad \mathbf{A} \mathbf{x} = \mathbf{b}. \]

- Feasible set is a polyhedron

Recall the diet problem from the last lecture.
- Choose quantity of \( x_1, \ldots, x_n \) of \( n \) foods
- One unit of food \( j \) cost \( c_j \), contains amount \( a_{ij} \) of nutrient \( i \).
- Diet requires nutrient \( i \) in quantity at least \( b_i \).

\[ \text{minimize } \mathbf{c}^T \mathbf{x} \]
\[ \text{subject to } \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq 0. \]
\[
( \sum a_{ij} x_j \geq b_i, \quad x_j \geq 0 )
\]
• Problems that can be converted to LP.

• Piecewise-linear minimization

\[
\text{minimize } \max_{i=1, \ldots, n} (a_i^T x + b_i).
\]

Equivalent to an LP

\[
\text{minimize } t
\]

\[
x, t
\]

Subject to \(a_i^T x + b_i \leq t, \ i = 1, \ldots, m\).

• Compressive Sensing and Sparse Recovery.

Assume that we are given a large vector \(x \in \mathbb{R}^n\) and a sensing matrix \(A \in \mathbb{R}^{m \times n}\), where \(m \leq n\).

\[
y = Ax\] is a "compressing" process.

\[
\begin{bmatrix}
1 \\
y \\
\end{bmatrix} = \begin{bmatrix}
A \\
x \\
\end{bmatrix}
\]

Q: Is \(y = Ax\) enough to determine \(x\)?

− No, any \(x + v\) for \(v \in N(A)\) satisfy \(y = A(x + v)\).

This is the reason why the above is called under-determined.
to determine/recover $x$ from $y$, normally we will need $m \geq n$, i.e., $A$ being a tall full-column rank matrix, which has no null space.

* Compressive Sensing: Consider $x$ being sparse; i.e., $x$ only has $s$ non-zero elements and $s \ll m$.

  - This means we "essentially" have an over-determined system.

* The RecoveryCriterion:

  $\min_x \|x\|_0$

  $\text{s.t. } y = Ax.$

(provably guaranteed to recover $x$

under some conditions)

Difficulty: $\|x\|_0 = \# \text{ of nonzero elements of } x.$

Combinatorial, nonconvex.

Idea: Use a convex norm to approximate $\ell_0$-norm

\[ \ell_1\text{-norm} \]

\[ (\ell_2\text{-norm})^2 \]

\[ \ell_p\text{-quasi-norm} \]

* $\ell_1$-norm is the best matching convex norm!
\[ \ell_1 \text{-minimization based Compressive Sensing} \]

\[
\text{minimize } \| x \|_1.
\]

Subject to \( y = Ax \).

Equivalent to \[
\text{minimize } \sum_{i=1}^n S_i
\]

subject to
\[
\begin{align*}
S_i & \leq x_i & \leq S_i \\
S_i & \geq 0
\end{align*}
\]

Which is a linear program (LP)

- **Worst Case Regression.**

In regression, we wish to attempt match

\[ y \approx Ax \]

\[
\text{training: } \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \approx \begin{bmatrix} \text{features} \\ \text{available} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}
\]

\[ \text{feature to be predict.} \]

\[ \text{testing: } \hat{y} = a^T x \]
Training: \( \min_x \|y - Ax\|_2^2 \) (Least squares)

But sometimes the data are not "equally good".

or, the linear model is not "equally good" to everyone.

We wish to minimize the worst matching error among all people:

\[
\min_x \max_{i=1,\ldots,m} |y_i - a_i^T x|
\]

\( \iff \min_x \|y - Ax\|_\infty \)

- Can be solved by LP

\[
\min_{x, t} t
\]

\[ -t \leq y_i - a_i^T x \leq t, \quad i=1, \ldots, m \]

\[ t \geq 0. \]
Quadratic Program (QP)

\[
\begin{align*}
\text{minimize} & \quad (y_z)x^TPx + q^Tx + r \\
\text{subject to} & \quad Gx \leq h \\
& \quad Ax = b
\end{align*}
\]

\(P \in S^n\), so the objective is convex quadratic.

- Minimize a convex quadratic function over a polyhedron.

- Important Applications.
  - Least-Squares
    \[
    \text{minimize} \quad \| Y - Ax \|_2^2
    \]
    Convex; also has an analytical solution \(x^* = A^+ b\).
  - Constrained least squares.
Application #1: Simplex-constrained least-squares for topic assignment.

Consider a topic model. 

\[ \text{Topics} \quad \text{Weighting vector} \]

\[ \text{Dictionary} \sim \left[ \begin{array}{c} y_1 \\ a_1 \\ a_2 \\ \vdots \\ a_K \\ x \end{array} \right] \]

\[ y_e \approx A x_e \]

- Intuition: A document is a weighted combination of several topics.
- Given some learned topics \( A = [a_1, \ldots, a_K] \) from the training set, what topics does \( y_{\text{new}} \) talk about?

(\text{Prediction/assignment})

- Formulation:

\[ \text{minimize } \| y_{\text{new}} - A x_{\text{new}} \|_2 \]

Subject to \( 1^T x_{\text{new}} = 1 \).

\( x_{\text{new}} \geq 0 \).

- This is used in many "assignment" problems.

(\text{total budget & nonnegative})
Second-order cone programming (SOCP)

\[
\begin{align*}
\text{minimize} \quad & f^T x \\
\text{subject to} \quad & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i=1, \ldots, m \\
& F x = g.
\end{align*}
\]

\[(A_i \in \mathbb{R}^{n \times n}, \quad F \in \mathbb{R}^{p \times n})
\]

\item \(\|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i=1, \ldots, m\) are called second-order cone (SOC) constraints.

\[(A_i x + b_i, c_i^T x + d_i) \in \text{Second-order cone in } \mathbb{R}^{n+1_2}
\]

\[\text{for } n_i = 0 \Rightarrow \text{LP}
\]

\[\text{for } c_i = 0 \Rightarrow (QC) \text{QP, i.e.,}
\]

\[
\begin{align*}
\text{minimize} \quad & f^T x \\
\text{subject to} \quad & \|A_i x + b_i\|_2 \leq d_i^2 \\
& F x = g.
\end{align*}
\]

Note: Proof of convexity of \(\|A_i x + b_i\|_2 \leq c_i^T x + d_i\)

Consider \(\theta x_1 + (1-\theta) x_2\) for \(\theta \in [0,1]\)
\[
\|A_1 \theta x_1 + (1-\theta) x_2 + b_i\|_2
\]

\[
= \|A_1 \theta x_1 + \theta b_i + A_i (1-\theta) x_2 + (1-\theta) b_i\|_2
\]

\[
\leq \theta \|A_i x_1 + b_i\|_2 + (1-\theta) \|A_i x_2 + b_i\|_2
\]

\[
\leq \theta (c_i^T x_1 + d_i) + (1-\theta) (c_i^T x_2 + d_i)
\]

\[
\leq c_i^T (\theta x_1 + (1-\theta) x_2) + d_i \quad \text{Q.E.D.}
\]
Application of SOCP:

- Robust Linear Programming.

\[
\begin{align*}
& \text{minimize } \mathbf{c}^T \mathbf{x} \\
& \text{subject to } \mathbf{a}_i^T \mathbf{x} \leq \mathbf{b}_i, ~ i = 1, \ldots, m.
\end{align*}
\]

There might be uncertainty in \( \mathbf{c}, \mathbf{a}_i, \mathbf{b}_i \).

Two ways of modeling uncertainty:

• Deterministic model:

\[
\begin{align*}
& \text{minimize } \mathbf{c}^T \mathbf{x} \\
& \text{subject to } \mathbf{a}_i^T \mathbf{x} \leq \mathbf{b}_i ~ \text{for all } \mathbf{a}_i \in \mathcal{E}_i, ~ i = 1, \ldots, m.
\end{align*}
\]

Note: in the diet problem, it could mean that the amount of nutrient in food \( i \) is not certain.

• Stochastic model: \( \mathbf{a}_i \) is random variable; constraints must hold with probability \( \gamma \).

\[
\begin{align*}
& \text{minimize } \mathbf{c}^T \mathbf{x} \\
& \text{subject to } \text{Prob} (\mathbf{a}_i^T \mathbf{x} \leq \mathbf{b}_i) \geq \gamma, ~ i = 1, \ldots, m.
\end{align*}
\]
A deterministic approach via SOCP.

Choose an ellipsoid as $E_i$:

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

($\bar{a}_i \in \mathbb{R}^n$, $P_i \in \mathbb{R}^{n \times n}$)

$\bar{a}_i$ is the "rough idea" that you know about $a_i$.

$E_i$ is the confidence region you have about $a_i$.

- **Robust LP.**

  minimize $c^T x$

  subject to $a_i^T x \leq b_i$, $\forall a_i \in E_i$, $i = 1, \ldots, m$

  is equivalent to the SOCP

  minimize $c^T x$

  subject to $\bar{a}_i^T x + \|P_i x\|_2 \leq b_i$, $i = 1, \ldots, m$

**Proof:** $a_i^T x \leq b_i$, $\forall a_i \in \bar{a}_i + P_i u \|u\|_2 \leq 1$.

means that

$$\sup_{\|u\|_2 \leq 1} (\bar{a}_i + P_i u)^T x \leq b_i$$

LHS: $= a_i^T x + \sup_{\|u\|_2 \leq 1} (\bar{a}_i + P_i u)^T x$

= $a_i^T x + \|P_i x\|_2$

(How to prove this?)
- Stochastic Approach via SOCP

- Assume \( a_i \) is Gaussian with mean \( \bar{a}_i \), covariance \( \Sigma_i \):
  \[ a_i \sim N(\bar{a}_i, \Sigma_i) \]

- \( a_i^T x \) is Gaussian r.v. with mean \( \bar{a}_i^T x \), variance \( x^T \Sigma_i x \):
  hence
  \[ \text{Prob}(a_i^T x \leq b_i) = \Phi\left( \frac{b_i - \bar{a}_i^T x}{\|\Sigma_i^{1/2} x\|} \right) \]
  where \( \Phi(x) = \left(\frac{1}{\sqrt{2\pi}}\right) \int_x^\infty e^{-t^2/2} \, dt \) is CDF of \( N(0,1) \)

- Robust LP

  \[
  \begin{align*}
  &\text{minimize} & & c^T x \\
  \text{subject to} & & & \text{Prob}(a_i^T x \leq b_i) \geq y, \quad i=1, \ldots, m \\
  \text{with} & & & y \geq \frac{1}{2}, \text{is equivalent to} \\

  &\text{minimize} & & c^T x \\
  \text{subject to} & & & \bar{a}_i^T x + \Phi^{-1}(y) \|\Sigma_i^{1/2} x\| \leq b_i, \quad i=1, \ldots, m
  \end{align*}
  \]
Support Vector Machines.

- SVM is a perfect example of applying what we have learned from this course to machine learning.

- Classification: Assume that we have two classes of data, namely, 'cat' and 'dolphin'. Our goal is to train a classifier that can identify the class label of an unseen data.

- Setup: We are given training data

\[ \{ \mathbf{x}_i, y_i \}_{i=1}^N \]

where, \( \mathbf{x}_i \in \mathbb{R}^n \) and \( y_i \in \{+1, -1\} \).

- \( y_i = +1 \) means \( \mathbf{x}_i \) is a training sample from class 1.
- \( y_i = -1 \) means \( \mathbf{x}_i \) is a training sample from class 2.

- Goal of the training stage: find a classifier that can separate/identify all the training samples according to their labels.

(Remark: there are many other considerations ... but the above is roughly the goal)
Which separation hyperplane is the best?

Intuitively, we wish the classifier to be "as far away as possible" to the convex hulls spanned by the two test classes.

How to formulate the problem?
Intuition of finding a "good" separation plane:

1) Find convex hulls of class A and B.
2) Find points from both convex hulls that are closest to each other.
3) Construct a line segment between the two points.
4) Find a hypersphere that is orthogonal to the line segment and goes through the middle point of the line segment.
• **Convex hull of class A.**

\[
\text{ConvA} = \{ x | x = u_i A_i + \ldots + u_m A_m \}.
\]

\[\sum u_i = 1, \; u_i \geq 0\]

Where \( A_i \in \mathbb{R}^n \) is the \( i \)-th training sample from class A.

• **Convex hull of class B**

\[
\text{ConvB} = \{ x | x = u_i B_i + \ldots + u_k B_k \}.
\]

\[\sum u_i = 1, \; u_i \geq 0\]

• The problem of finding two closest points in the convex hulls, is

\[
\text{minimize } \frac{1}{2} \| A^T u - B^T v \|^2_2
\]

\[(C\text{-Hull}) \quad u, \; v \in \mathbb{R}^m, \quad A^T u = 1, \quad B^T v = 1.
\]

\[u \geq 0, \quad v \geq 0.\]

Where

\[
A = \begin{bmatrix} A_1^T \\ \vdots \\ A_m^T \end{bmatrix}
\]

\[
B = \begin{bmatrix} B_1^T \\ \vdots \\ B_m^T \end{bmatrix}
\]

**Remark:** C-Hull is a QP, which is convex and efficiently solvable.

• Let \( \tilde{u}, \tilde{v} \) be an optimal solution to \((C\text{-Hull})\).

Then, \[W = c - d = A^T \tilde{u} - B^T \tilde{v}\]

\[r = \left( \frac{c + d}{2} \right)^T W\]

The hyperplane is \( W^T x = r \)
An alternative way of finding the "best" separation hyperplane.

Q: What is the distance between \( w^T x = \alpha \) and \( w^T x = \beta \)?

- Let \( x_1 = \frac{\alpha}{\|w\|_2} \) and \( x_2 = \frac{\beta}{\|w\|_2} \).

- Both \( x_1 \) and \( x_2 \) are on the vector \( w \) (same direction).

- W.l.o.g. assume \( \alpha < \beta \).

Then, the distance is \( \|x_2 - x_1\|_2 = \frac{\|w \frac{\beta - \alpha}{\|w\|_2}\|_2}{\|w\|_2} \).

- Come back to SVM

We wish to find a hyperplane, with \( w \) such that the distance \( \frac{\beta - \alpha}{\|w\|_2} \) is maximized.
- Hence, the C-Margin Problem

$$\text{minimize } \frac{1}{2} ||w||^2 - (x - \beta)$$

$$w, \alpha, \beta \quad A^w \geq \alpha 1, \quad B^w \leq \beta 1$$

Where $1$ is an all-one vector with proper length.

**Remark:** Still a QP - efficiently solvable.

How about the inseparable case?

**Class B**

**Class A**

The two convex hulls are not separable.

**Class B** **Class A**

But if we "shrink" both convex hulls a bit, the shrinked versions are separable.
Definition (Reduced Convex Hull). The set of all combinations with the form \( x = A^T u \), where \( A^T u = x \), \( u \geq 0 \), \( u_i \leq d \), \( d = 13 \).

Example

\[ \begin{align*}
&\text{Convex hull} \\
&\text{reduced convex hull.}
\end{align*} \]

The RC-hull formulation

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} ||A^T u - B^T v||^2 \\
\text{subject to} & \quad A^T u = x, \quad B^T v = y, \quad 0 \leq u_i \leq d, \quad 0 \leq v_i \leq d.
\end{align*}
\]

Remark: still a QP!

The RC-Margin formulation

\[
\begin{align*}
\text{minimize} & \quad D \left( \varepsilon^T \xi + \gamma^T y \right) + \frac{1}{2} ||W||^2 - (\alpha - \beta) \\
\text{s.t.} & \quad Aw - \alpha e + \xi \geq 0, \quad \xi \geq 0 \\
& \quad -Bw + \beta e + y \geq 0, \quad y \geq 0.
\end{align*}
\]

Remark: still a QP!
Semidefinite Programming (SDP)

- Inequality form
  \[
  \text{minimize } c^T x \\
  \text{subject to } F_i(x) \preceq 0 \quad (\text{LMI})
  \]
  \[
  F_i(x) = F_0 + x_1 F_1 + \cdots + x_n F_n, \quad x_i \in S^{m_i}
  \]

- Standard form
  \[
  \text{minimize } tr(CX) \\
  \text{subject to } X \succeq 0 \\
  tr(A_i X) = b_i, \quad i = 1, \ldots, m
  \]
  \[
  \text{where } A_i \in S^{n}, \quad C \in S^{n}
  \]

- The two forms can be shown to be equivalent.

Example: Max. eigenvalue minimization.

Recall that \( \lambda_{\text{max}}(\cdot) \) is convex.

\[
\lambda_{\text{max}}(A(x)) \leq c \Leftrightarrow A(x) - tI \preceq 0
\]

\[
\text{minimize } t \\
\text{subject to } A(x) - tI \preceq 0 \quad (\text{LMI})
\]

Example: Kernel SVM (come back after duality!)