CSE 203B: Convex Optimization
Discussion Week 3

Isabella Liu
1/21/2022
• Qualification & Enumeration
• Dual Cone
• Support Vector Machine
Content

- Qualification & Enumeration
- Dual Cone
- Support Vector Machine
Qualification & Enumeration

Implicit/qualification expression: \[ \{ x | Ax \leq b, x \in \mathbb{R}^n \} \]

Explicit/enumeration expression: \[ \{ U\theta | 1^T \theta = 1, \theta \in \mathbb{R}_+^m \} \]

Example:

\[
\begin{bmatrix}
1 & 1 \\
1 & 0 \\
-1 & 0 \\
0 & -1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\leq
\begin{bmatrix}
2 \\
1 \\
1 \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1 & -1 & -1 \\
1 & -1 & -1 & 3
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}
\]

Convert

(HW1.1: find extreme points)
Implicit/qualification expression:  
\[ \{ x | Ax \leq b, x \in \mathbb{R}^n \} \]

Explicit/enumeration expression:  
\[ \{ U\theta | 1^T \theta = 1, \theta \in \mathbb{R}_+^m \} \]

Question:
The dimension \( m \) of \( \theta \in \mathbb{R}_+^m \) is determined by?

The number of extreme points in \( \{ x | Ax \leq b, x \in \mathbb{R}^n \} \)
• HW 1.1 – how to find extreme points in a linear system of inequalities

• **Extreme points**: Let $C \subseteq \mathbb{R}^n$ be a convex set. For a $z \in C$, it is an extreme point of $C$ if there is no distinct $x, y \in C$ and $\lambda \in (0,1)$ such that $z = (1 - \lambda)x + \lambda y$. (always be the endpoint if in a line segment inside $C$)
HW 1.1 – how to find extreme points in a linear system of inequalities

- Set $n$ of the $m + n$ inequalities to equality (by turns) and solve the corresponding $n \times n$ system
- Check whether the solution satisfies the remaining inequalities, discard if doesn’t
- Note $x \in R^6_+$, so $x_i \geq 0$ are also inequality conditions
Content

• Qualification & Enumeration

• Dual Cone

• Support Vector Machine
Dual Cone

- Let $K$ be a cone, the set

$$K^* = \{ y \mid x^T y \geq 0, \forall x \in K \}$$

is called dual cone of $K$, $K^*$ is always convex, even when the original $K$ is not.
Example: **Self-dual**, the cone $\mathbb{R}_+^n$ is its own dual

\[ x^T y \geq 0, \forall x \geq 0 \iff y \geq 0 \]
Example: Dual cone of a non-convex set is convex.

\[ C^* = \{ y \mid \langle y, x \rangle \geq 0, \forall x \in C \} \]

https://www.youtube.com/watch?v=CTNFqM8nRS0
HW 1.3

Derive the dual cone of the set \( \{ x | Ax \leq 0, x \in \mathbb{R}^6 \} \)

for any \( y \) in the dual cone, there should be \( y^T x \geq 0, \forall x \in \{ x | Ax \leq 0, x \in \mathbb{R}^6 \} \)
Content

- Qualification & Enumeration
- Dual Cone
- Support Vector Machine (SVM)
Given a dataset of $n$ points in form $(x_1, y_1), ..., (x_n, y_n)$ where class labels $y_i$ are either 1 or -1. The SVM tries to find the “maximum-margin hyperplane” that divides the points into correct groups.

$$\min ||w||^2_2$$
$$\text{s.t. } \begin{cases} w^T x_i - b \leq -1, & \text{if } y_i = -1 \\ w^T x_i - b \geq 1, & \text{if } y_i = 1 \end{cases}$$

Geometrically, the distance between hyperplane $w^T x - b = 1$ and $w^T x - b = -1$ is $\frac{2}{||w||}$, so maximize it is equivalent to minimize $||w||$
Support Vector Machine (SVM)

- In hard-margin SVM, the decision boundary is only affected by the support vectors

Check out this live demo
Support Vector Machine (SVM)

- Sometimes data are not linearly separable 😞
  - Soft-margin SVM: Add a loss function to penalize points on the wrong side

\[
\max(0, 1 - y_i (w^T x_i - b))
\]

if a point is on the wrong side, the loss is proportional to its distance to the margin.

Minimize:

\[
\lambda \|w\|^2 + \left[ \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i (w^T x_i - b)) \right]
\]
Support Vector Machine (SVM)

- Sometimes data are not linearly separable 😞
  - Kernel Trick
    - The idea of mapping the data to a higher dimensional space
    - Kernel functions is computational efficient
  - Common kernel functions:
    - Polynomial kernel
      \[ k(x, y) = (x^T y + 1)^d \]
    - Gaussian kernel
      \[ k(x, y) = e^{-\gamma \|x-y\|^2}, \gamma > 0 \]
Support Vector Machine (SVM)

• Multiclass classification using SVM
  • One-to-one: use a binary SVM for each pair of classes
  • One-to-rest: separate a class and the rest of points

https://www.baeldung.com/cs/svm-multiclass-classification
Thank you