Semi-Defined Programming
半正定规划

Outline

- 正定与半正定
- Review: 线性规划
- 半正定规划
- 一个case

Positive Defined 正定
Positive Semi-Defined 半正定

➢ 正定矩阵 vs. 半正定矩阵

【定义1】给定一个大小为 $n \times n$ 的实对称矩阵 A ，若对于任意长度为 n 的非零向量 $\boldsymbol{x}$ ，有 $\boldsymbol{x}^T A\boldsymbol{x} > 0$ 恒成立，则矩阵 A 是一个正定矩阵。

【定义2】给定一个大小为 $n \times n$ 的实对称矩阵 A ，若对于任意长度为 n 的向量 $\boldsymbol{x}$ ，有 $\boldsymbol{x}^T A\boldsymbol{x} \geq 0$ 恒成立，则矩阵 A 是一个半正定矩阵。

仔细看一下上面的定义可以看到两种矩阵的唯一区别就是正定要求是大于0，而半正定要求大于等于0。这个是不是很像二次函数 $y = ax^2$ ：

- 当 $a > 0$ 时，$y > 0$ ；

- 当 $a \geq 0$ 时， $y \geq 0$

Positive Defined 正定
Positive Semi-Defined 半正定

实对称矩阵 $A=\begin{bmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{bmatrix}\in\mathbb{R}^{3\times 3}$ 是否是正定矩阵?

解：设向量 $\boldsymbol{x}=\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\in\mathbb{R}^3$ 为非零向量，则

$$\boldsymbol{x}^T A\boldsymbol{x}=\begin{bmatrix}(2x_1-x_2) & (-x_1+2x_2-x_3) & -x_2+2x_3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$$

$$=x_1^2+(x_1-x_2)^2+(x_2-x_3)^2+x_3^2>0$$

因此，矩阵 A 是正定矩阵。

Positive Defined 正定
Positive Semi-Defined 半正定

【定义2】给定一个大小为 $n \times n$ 的实对称矩阵 A ，若对于任意长度为 n 的向量 $\boldsymbol{x}$ ，有 $\boldsymbol{x}^T A\boldsymbol{x} \geq 0$ 恒成立，则矩阵 A 是一个半正定矩阵。

根据正定矩阵和半正定矩阵的定义，我们也会发现：半正定矩阵包括了正定矩阵，与非负实数 (non-negative real number)和正实数 (positive real number)之间的关系很像。

几何意义：

正定矩阵和半正定矩阵的直观解释

若给定任意一个正定矩阵 $A \in \mathbb{R}^{n\times n}$ 和一个非零向量 $\boldsymbol{x} \in \mathbb{R}^n$ ，则两者相乘得到的向量 $\boldsymbol{y} = A\boldsymbol{x} \in \mathbb{R}^n$ 与向量 $\boldsymbol{x}$ 的夹角恒$\boxed{\text{小于 } \frac{\pi}{2}}$.(等价于： $\boldsymbol{x}^T A\boldsymbol{x} > 0$.)

若给定任意一个半正定矩阵 $A \in \mathbb{R}^{n\times n}$ 和一个向量 $\boldsymbol{x} \in \mathbb{R}^n$ ，则两者相乘得到的向量 $\boldsymbol{y} = A\boldsymbol{x} \in \mathbb{R}^n$ 与向量 $\boldsymbol{x}$ 的夹角恒$\boxed{\text{小于或等于 } \frac{\pi}{2}}$.(等价于： $\boldsymbol{x}^T A\boldsymbol{x} \geq 0$.)

Outline

- 正定与半正定
- Review: 线性规划
- 半正定规划
- 一个case

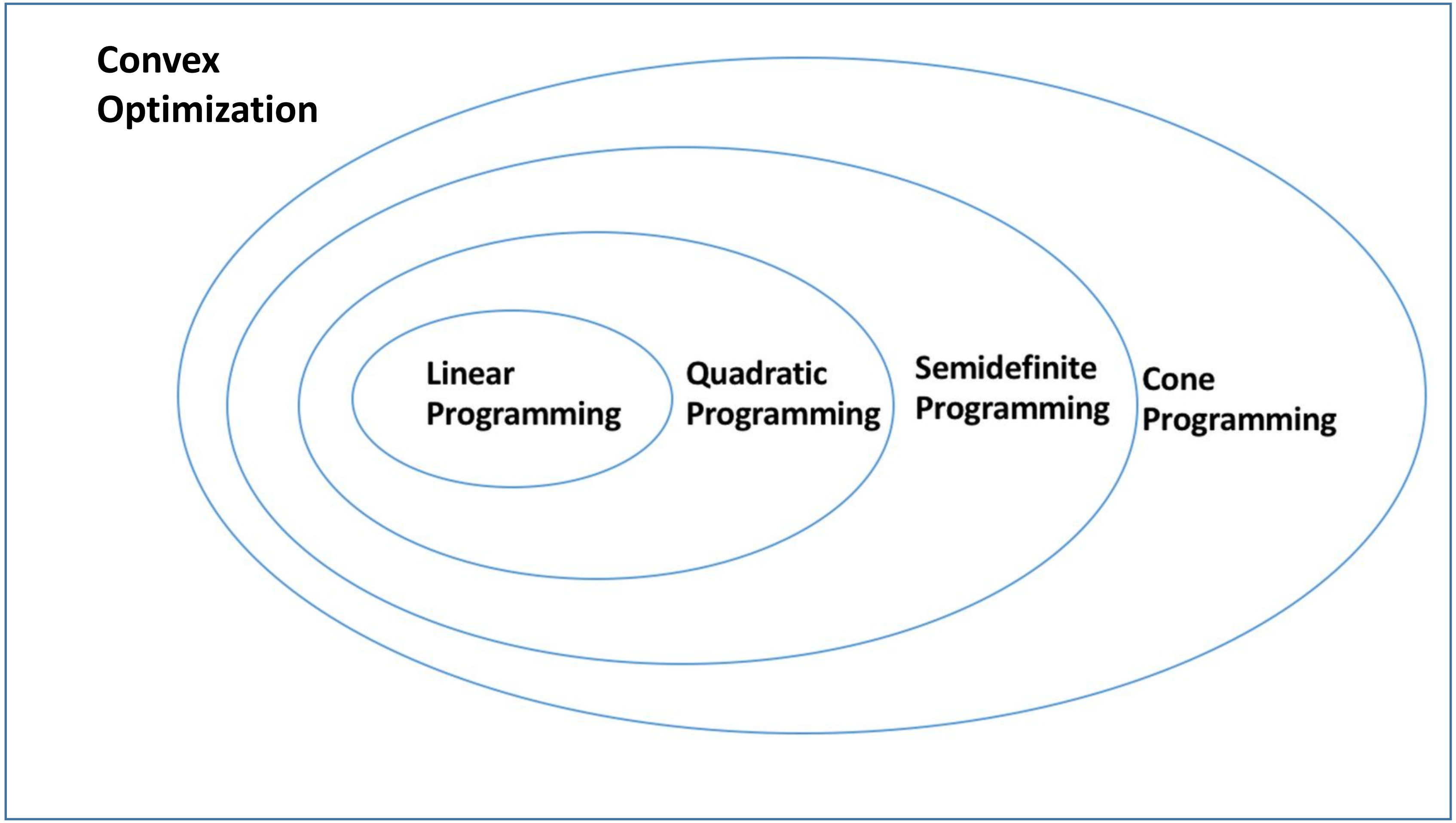
Convex Optimization: 凸优化问题

Cone Programming: 锥规划

Semidefine-Programming: 半正定规划

Quadratic Programming: 二次规划

Linear Programming: 线性规划



Key terms to Linear programming

$$
\begin{aligned}
\min_{x \in \mathbf{R}^n} \quad & f(x) \\
\text{s.t.} \quad & c_i(x) \leqslant 0, \quad i = 1, 2, \cdots, k \\
& h_j(x) = 0, \quad j = 1, 2, \cdots, l
\end{aligned}
$$

- For a (convex) optimization problem, when the objective and constraint functions are all affine, the problem is called a linear program (LP).
- 线性规划问题：在线性约束条件下，线性目标函数求极值的问题
- 可行解 (feasible solution)：满足线性约束条件下的解
- 可行域 (feasible set)：由所有可行解组成的集合
- 最优解：使目标函数取得极值的可行解

Key terms to Linear programming

- 定义模型步骤：

1. 确定决策变量；2. 确定线性目标函数，求max或min；
3. 确定线性约束条件；4. 写出数学模型。

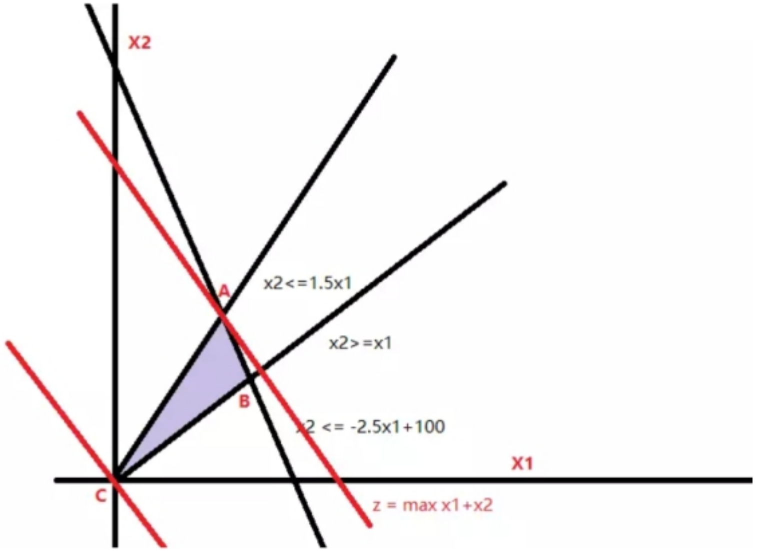
例：假设你是一个单位的采购经理，你有2000元经费，需要采购单价为50元的若干桌子和单价为20元的若干椅子，你希望桌椅的总数尽可能的多，但要求椅子数量不少于桌子数量，且不多于桌子数量的1.5倍，那你需要怎样的一个采购方案呢？

x_1张桌子，x_2把椅子

目标：max $x_1 + x_2$

约束：

$$x_1 \le x_2$$
$$1.5x_1 \ge x_2$$
$$50x_1 + 20x_2 \le 2000$$
$$x_1, x_2 \ge 0$$



Various forms of linear programming

- Form 1. General form: mixture of linear inequalities and equalities

$$
\begin{array}{lcccccccccc}
\min & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & & & \\
s.t. & a_{i1}x_1 & + & a_{i2}x_2 & + & \dots & + & a_{in}x_n & \geq & b_i & i \in M \\
 & a_{j1}x_1 & + & a_{j2}x_2 & + & \dots & + & a_{jn}x_n & = & b_j & j \in \overline{M} \\
 & & & & & & & x_i & \geq & 0 & i \in N
\end{array}
$$

- Form 2. Standard form: linear inequalities

$$
\begin{array}{lcccccccccc}
\min & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & & & \\
s.t. & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq & b_1 & \\
 & a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq & b_2 & \\
 & \dots & & \dots & & \dots & & \dots & & & \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq & b_m & \\
 & & & & & & & x_i & \geq & 0 & \text{for } \forall i
\end{array}
$$

  - Standard form in matrix language:

$$
\begin{array}{lr}
\min & \mathbf{c}^{\mathbf{T}}\mathbf{x} \\
s.t. & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}
\end{array}
$$

Outline

- 正定与半正定
- Review: 线性规划
- 半正定规划
- 一个case

半正定规划

$$\min_{x \in D} \quad c^T x$$
$$\textbf{subject to}: x_1 F_1 + x_2 F_2 + \cdots + x_n F_n \le F_0$$
$$Ax = b$$

其中 $x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ ， $F_i \in \mathbb{R}^{d \times d}$ 是对称矩阵， $A \in \mathbb{R}^{m \times n}$ 。以上形式可以换写成**SDP的standard form**：

$$\min_{X \in D} \quad C \bullet X$$
$$\textbf{subject to}: A_i \bullet X = b_i, i = 1, 2, \cdots, n$$
$$X \succeq 0$$

其中 $C \bullet X = \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij}$ (Frobenius inner product)， C， $X \in \mathbb{R}^{n \times n}$ 是对称矩阵，$b_i \in \mathbb{R}$ 。

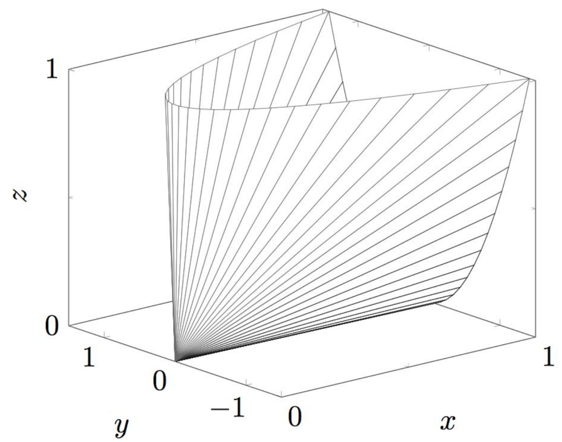
半正定规划

$$\min_{x \in D} \quad c^T x$$
$$\textbf{subject to:}\ x_1F_1 + x_2F_2 + \cdots + x_nF_n \le F_0$$
$$Ax = b$$

$$\min_{X \in D} \quad C \bullet X$$
$$\textbf{subject to:}\ A_i \bullet X = b_i, i = 1, 2, \cdots, n$$
$$X \succeq 0$$

首先在SDP的standard form中，要寻找对称矩阵 X ，本质上其实还是寻找矩阵 X 里的 n^2 个元素 x_{ij} ，由于 X 是对称的，所以其实是寻找X里面上三角的 $\frac{n(n+1)}{2}$ 个元素。而 $C \bullet X = \sum_{j=1}^{n}\sum_{i=1}^{n} c_{ij}x_{ij}$ 就是这 $\frac{n(n+1)}{2}$ 个元素 x_{ij} 的线性方程，所以目标函数是convex的，standard form跟第一个SDP的formulation是等价的。

那可行域也是convex的吗？首先同理 $C \bullet X$ 的展开， $A_i \bullet X = b_i$ 是 x_{ij} 的affine函数。而且， $X \succeq 0$ 代表的区域也是convex的，为什么？举个例子，我们不妨看看 $X = \begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix}$ 为2x2对称矩阵时 $X \succeq 0$ 所代表的区域，也就是 $x_{11} \ge 0, x_{22} \ge 0, x_{11}x_{22} - x_{12}^2 \ge 0$ 的区域，如下图中曲面所包裹整个内部区域（图中 x_{11} 是 x 轴， x_{12} 为 y 轴， x_{22} 为 z 轴），即 $y^2 < xz$ 区域：



线性规划 vs. 半正定规划

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \leqslant 0, \quad i = 1, 2, \cdots, k \\ & h_j(x) = 0, \quad j = 1, 2, \cdots, l \end{aligned}$$

Standard form in matrix language:

$$\begin{aligned} \min \quad & \mathbf{c}^{\mathbf{T}}\mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \min_{x \in D} \quad & c^T x \\ \textbf{subject to}: \quad & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n \leq F_0 \\ & Ax = b \end{aligned}$$

$$\begin{aligned} \min_{X \in D} \quad & C \bullet X \\ \textbf{subject to}: \quad & A_i \bullet X = b_i, i = 1, 2, \cdots, n \\ & X \succeq 0 \end{aligned}$$

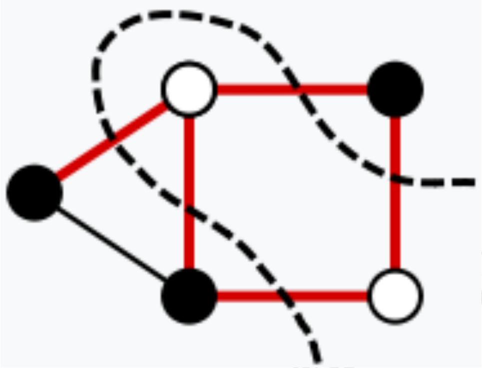
LP问题是SDP的特殊情况，相当于以上SDP的standard form中的C矩阵和 X矩阵 不仅仅是对称的，还是对角的。

$$C = \begin{pmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & c_n \end{pmatrix}, X = \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & x_n \end{pmatrix}$$

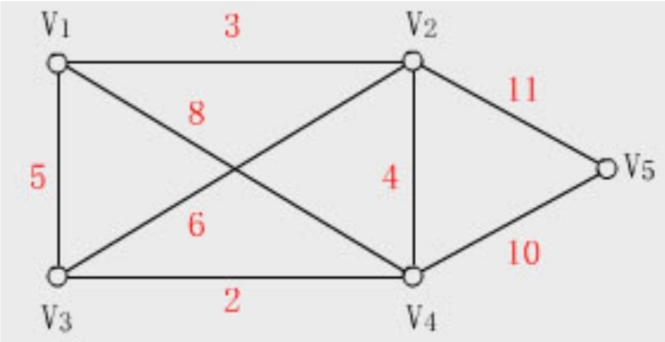
一个case: 最大割问题 max-cut problem

G: Graph, V: Vertex, E: Edge

问题描述： 给定一个无向图 G = (V, E), 每条顶点之间的连线E都又权重w, 现在使用一条割线，将图上的所有顶点分成两个子集，要求割线所经过的edge的权重之和最大。



典型的NP-hard 问题，没办法直接求解，需要寻求近似求解方法



给定 $G=(V,E)$，有 $V = \{1,2,\dots,n\}\ E \in V \times V$; w_{ij} 为顶点i和顶点j之间边的权重，

对于顶点集V的任意一个真子集S有：

$$\delta(S) = \{e_{ij} \in E, i \in S, j \in V - S\}$$

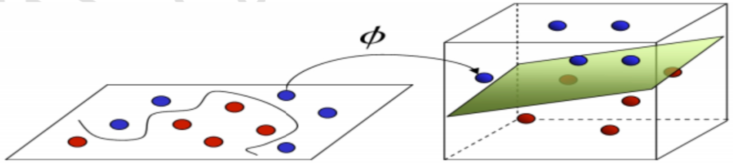
则由S确定的割的集合cut(S)为：

$$cut(S) = \sum_{e_{ij} \in \delta(S)} w_{ij}$$

目标：使得cut(S)最大

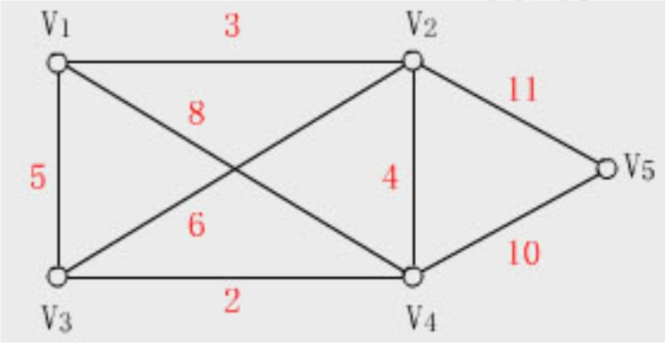
图的矩阵形式:

$$\begin{bmatrix} 0 & 3 & 5 & 8 & 0 \\ 3 & 0 & 6 & 4 & 11 \\ 5 & 6 & 0 & 2 & 0 \\ 8 & 4 & 2 & 0 & 10 \\ 0 & 11 & 0 & 10 & 0 \end{bmatrix}$$



SVM 线性不可分-> 高维线性可分

一个case: 最大割问题 max-cut problem



图的矩阵形式:

$$\begin{bmatrix} 0 & 3 & 5 & 8 & 0 \\ 3 & 0 & 6 & 4 & 11 \\ 5 & 6 & 0 & 2 & 0 \\ 8 & 4 & 2 & 0 & 10 \\ 0 & 11 & 0 & 10 & 0 \end{bmatrix}$$

给定 $G = (V, E)$，有 $V = \{1,2,\ldots,n\}$ $E \in V \times V$; w_{ij} 为顶点i和顶点j之间边的权重，

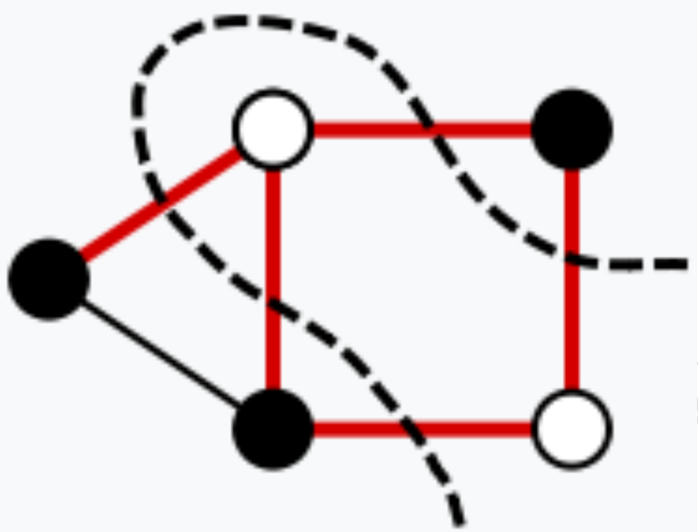
对于顶点集V的任意一个真子集S有：

$$\delta(S) = \{e_{ij} \in E, i \in S, j \in V - S\}$$

则由S确定的割的集合cut(S)为：

$$cut(S) = \sum_{e_{ij} \in \delta(S)} w_{ij}$$

目标：使得cut(S)最大



使用**线性规划**进行松弛(近似):

$\{x_v\}_{v \in V} \in \{0,1\}$ $\{z_e\}_{e \in E} \in \{0,1\}$

x encodes which partition the vertex is in

z encodes whether the edge is cut

$$\max \sum_{uv \in E} w_{uv} z_{uv}$$

$$\text{s.t.} \quad z_{uv} \le x_u + x_v$$

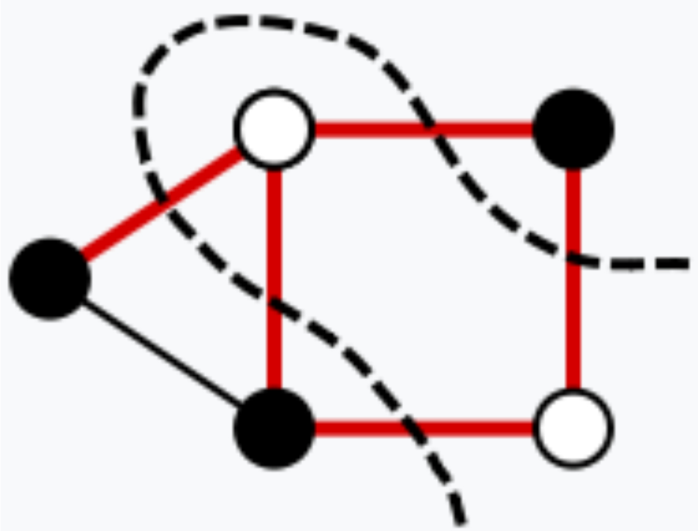
$$z_{uv} \le 2 - (x_u + x_v)$$

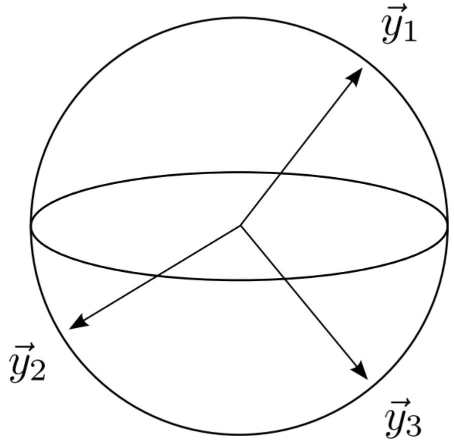
LP Relax $\Rightarrow$ let $x_v, z_e \in [0,1]$

一个case: 最大割问题 max-cut problem

使用**半正定规划**进行松弛(近似):

想法是这样的：我们把 $\{x_i\}$ 扩展为 $\mathbb{R}^n$ 中单位球上的单位向量 $\{\boldsymbol{v}_i\}$ ，使用两个向量的夹角表示两个顶点是否在同一个割集内。若 $\boldsymbol{v}_i^T\boldsymbol{v}_j=\cos\langle\boldsymbol{v}_i,\boldsymbol{v}_j\rangle$ 越接近1，我们越倾向于把 i,j 放在同一个割集内；若之越接近-1，我们越倾向于把 (i,j) 作为一个割边。





$$\max \sum_{uv\in E} w_{uv}\left(\frac{1}{2}-\frac{1}{2}\vec{y}_u\cdot\vec{y}_v\right)$$

$$\text{s.t.}\quad \vec{y}_v\cdot\vec{y}_v=1 \quad \forall v\in V$$

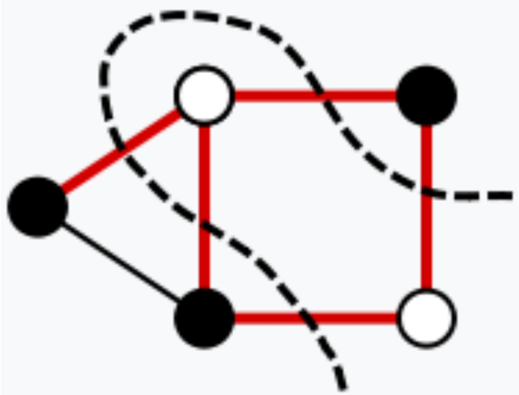
$$\max\ \frac{1}{4}\sum_{i,j} w_{ij}-\frac{1}{4}\boldsymbol{W}\cdot\boldsymbol{V},$$

$$\text{subj. to } \boldsymbol{V}_{ii}=1, i=1,2,\ldots,n.$$

从低维的线性空间转换到高维的空间重进行求解。

核心思想

一个case: 最大割问题 max-cut problem



使用**半正定规划**进行松弛(近似):

$$\max \ \frac{1}{4}\sum_{i,j} w_{ij} - \frac{1}{4}\boldsymbol{W}\cdot\boldsymbol{V},$$
$$\text{subj. to } \boldsymbol{V}_{ii} = 1, i = 1, 2, \ldots, n.$$

其中 $\boldsymbol{V}$ 是 $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ 的Gram矩阵，即 $\boldsymbol{V}_{ij} = \boldsymbol{v}_i^T\boldsymbol{v}_j$ 。我们有约束 $\|v_i\|_2 = 1$ 是因为所有向量都是单位向量。系数为 $\frac{1}{4}$ 是因为矩阵中 (i, j) 和 (j, i) 会算两次。

现在我们作一次松弛，移除Gram矩阵的要求，只要求 $\boldsymbol{V}$ 是半正定矩阵，即：

$$\max \ \frac{1}{4}\sum_{i,j} w_{ij} - \frac{1}{4}\boldsymbol{W}\cdot\boldsymbol{V},$$
$$\text{subj. to } \boldsymbol{V}_{ii} = 1, i = 1, 2, \ldots, n,$$
$$\boldsymbol{V} \geq \boldsymbol{0}.$$

从低维的线性空间转换到高维的空间重进行求解。

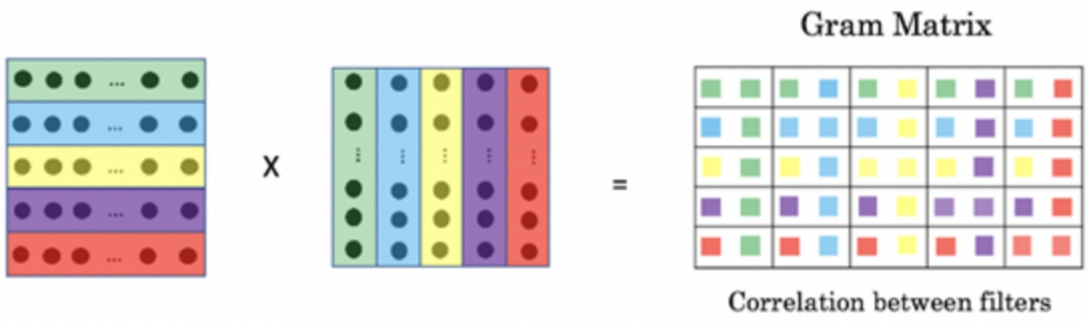
核心思想

Gram 矩阵 (格雷姆矩阵)

Gram矩阵是两两向量的内积组成，所以Gram矩阵可以反映出该组向量中各个向量之间的某种关系。

n维欧式空间中任意k个向量之间两两的内积所组成的矩阵，称为这k个向量的格拉姆矩阵(Gram matrix)，很明显，这是一个对称矩阵。

$$\Delta(\alpha_1,\alpha_2,\dots,\alpha_k)=\begin{pmatrix}(\alpha_1,\alpha_1) & (\alpha_1,\alpha_2) & \dots & (\alpha_1,\alpha_k)\\ (\alpha_2,\alpha_1) & (\alpha_2,\alpha_2) & \dots & (\alpha_2,\alpha_k)\\ \dots & \dots & \dots & \dots\\ (\alpha_k,\alpha_1) & (\alpha_k,\alpha_2) & \dots & (\alpha_k,\alpha_k)\end{pmatrix}$$



Some References

https://zhuanlan.zhihu.com/p/69126787

https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E8%A7%84%E5%88%92%E7%9A%84%E6%9D%BE%E5%BC%9B

https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture10.pdf

Thanks