Correlational Clustering Problem

- **Instância:** Um grafo G com um peso interno w_{ij}^{+} e um peso externo w_{ij}^{-} não-negativos em cada aresta ij.
- **Objetivo:** Obter partição P dos vértices que maximize $w(P)$: a soma do peso interno das arestas internas mais o peso externo das arestas externas à partição. Ou seja, maximize $w(P)=w^{+}(E_P^{+}) + w^{-}(E_P^{-})$, onde E_P^{+} contém arestas com extremidades em um mesmo conjunto de P e E_P^{-} contém arestas com extremidades em conjuntos distintos de P.

Algoritmo Cluster-0.5-aprox(G, w^{+}, w^{-})

1. **seja** $P_1 = \{\{v_1, \ldots, v_n\}\}$ e seja **seja** $P_2 = \{\{v_1\}, \{v_2\}, \ldots, \{v_n\}\}$
2. **retorne** o máximo entre $w^{+}(P_1)$ e $w^{-}(P_2)$

PROVA: Seja P a partição retornada pelo algoritmo.

$$opt \leq \sum_{ij \in E(G)} (w_{ij}^{+} + w_{ij}^{-}) = w^{+}(P_1) + w^{-}(P_2) \leq 2 \cdot w(P)$$

Clustering: Programação Quadrática

- Para cada vértice $i \in V(G)$, usa-se um vetor x_i de tam. n
- **Interpretação:** $x_i = e_k$ (vetor com 1 na coord. k e 0 nas demais) $\Rightarrow$ vértice i está no cluster k

- Produto escalar (ou produto interno)
  $x_i \cdot x_j = \sum_{k=1}^{n} x_{ik}x_{jk} = x_{i1}x_{j1} + \ldots + x_{in}x_{jn}$
- Módulo (norma euclidiana)
  $|x_i| := \|x_i\|_2 = \sqrt{x_{i1}^2 + \ldots + x_{in}^2} = \sqrt{x_i x_i} = 1$
- $e_i \cdot e_i = 1$ e $e_i \cdot e_j = 0$ para $i \neq j$

Formulação de Programação Quadrática

- **Maximizar** $\sum_{ij \in E(G)} \left(w_{ij}^{+}(x_i \cdot x_j) + w_{ij}^{-}(1 - x_i \cdot x_j) \right)$, **restrito a**:
- Para cada $i \in V(G)$: $x_i \in \{e_1, e_2, \ldots, e_n\}$
- É claramente viável; basta tomar $x_i = e_1$ para todo $i \in V(G)$
- Obtém soℓ. exata $opt(G, w^{+}, w_{-})$, mas problema NP-Difícil.

Clustering: Programação Quadrática

Formulação de Programação Quadrática

- e_k (vetor tam. n com 1 na coord. k e 0 nas demais)

- **Maximizar** $\sum_{ij \in E(G)} \left(w_{ij}^{+}(x_i \cdot x_j) + w_{ij}^{-}(1 - x_i \cdot x_j) \right)$, **restrito a**:
- Para cada $i \in V(G)$: $x_i \in \{e_1, e_2, \ldots, e_n\}$

Relaxação vetorial da Programação Quadrática

- Para cada $i \in V(G)$, usa-se um vetor v_i de tam. n e de módulo 1

- **Maximizar** $\sum_{ij \in E(G)} \left(w_{ij}^{+}(v_i \cdot v_j) + w_{ij}^{-}(1 - v_i \cdot v_j) \right)$, **restrito a**:
- Para cada $i \in V(G)$: $v_i \cdot v_i = 1$
- Para cada $i, j \in V(G)$: $v_i \cdot v_j \geq 0$

- É claramente viável; basta tomar $v_i = e_1$ para todo $i \in V(G)$
- Obtém soℓ. relaxada $\geq opt(G, w^+, w^-)$.

Clustering: Programação Semidefinida

Relaxação vetorial da Programação Quadrática

- **Maximizar** $\sum_{ij \in E(G)} \left(w_{ij}^{+}(v_i \cdot v_j) + w_{ij}^{-}(1 - v_i \cdot v_j) \right)$, **restrito a**:
- Para cada $i, j \in V(G)$: $\quad v_i \cdot v_i = 1 \quad$ e $\quad v_i \cdot v_j \geq 0$

Simplificação: encontrar vetores $v_1, \ldots, v_n$

- **Minimizar** $\sum_{ij \in E(G)} (w_{ij}^{-} - w_{ij}^{+})\, v_i \cdot v_j$, **restrito a**:
- Para cada $i, j \in V(G)$: $\quad v_i \cdot v_i = 1 \quad$ e $\quad v_i \cdot v_j \geq 0$

Reformulação: achar matriz Y (linhas de Y são vetores v_i)

- **Minimizar** $\sum_{ij \in E(G)} (w_{ij}^{-} - w_{ij}^{+})(YY^T)_{ij}$, **restrito a**:
- Para cada $i, j \in V(G)$: $\quad (YY^T)_{ii} = 1 \quad$ e $\quad (YY^T)_{ij} \geq 0$

Reform: achar matriz X positiva semidefinida ($X = YY^T$)

- **Minimizar** $\sum_{ij \in E(G)} (w_{ij}^{-} - w_{ij}^{+}) X_{ij}$, **restrito a**:
- Para cada $i \in V(G)$: $\quad X_{ii} = 1 \quad$ e $\quad X_{ij} \geq 0$
- Existem algoritmos que obtém Y a partir de X positiva semidefinida e que resolvem programas gerais com matriz positiva semidefinida

Algoritmo MaxCluster-PS (Programação Semidefinida)

Relaxação vetorial da Programação Quadrática

- **Maximizar** $\sum_{ij \in E(G)} \left(w_{ij}^{+}(v_i \cdot v_j) + w_{ij}^{-}(1 - v_i \cdot v_j) \right)$, **restrito a**:
- Para cada $i, j \in V(G)$: $\quad v_i \cdot v_i = 1 \quad$ e $\quad v_i \cdot v_j \geq 0$

Algoritmo MaxCluster-PS(G, w^+, w^-)

1. **sejam** $v_1, \ldots, v_n$ uma solução ótima da Relaxação vetorial acima
2. **sejam** $s_1, s_2 \leftarrow$ *RandEsfera*(n) (vetores aleat. tam. n e módulo 1)
3. $A_1 \leftarrow \{i \in V(G) : \; s_1 \cdot v_i > 0 \;\; e \;\; s_2 \cdot v_i > 0\}$
4. $A_2 \leftarrow \{i \in V(G) : \; s_1 \cdot v_i > 0 \;\; e \;\; s_2 \cdot v_i \leq 0\}$
5. $A_3 \leftarrow \{i \in V(G) : \; s_1 \cdot v_i \leq 0 \;\; e \;\; s_2 \cdot v_i > 0\}$
6. $A_4 \leftarrow \{i \in V(G) : \; s_1 \cdot v_i \leq 0 \;\; e \;\; s_2 \cdot v_i \leq 0\}$
7. **retorne** (A_1, A_2, A_3, A_4)

Algoritmo MaxCluster-PS (Programação Semidefinida)

Algoritmo MaxCluster-PS(G, w^+, w^-)

1. **sejam** $v_1, \ldots, v_n$ uma solução ótima da Relaxação vetorial acima
2. **sejam** $s_1, s_2 \leftarrow$ *RandEsfera*(n) (vetores aleat. tam. n e módulo 1)
3. $A_1 \leftarrow \{i \in V(G) : s_1 \cdot v_i > 0 \text{ e } s_2 \cdot v_i > 0\}$
4. $A_2 \leftarrow \{i \in V(G) : s_1 \cdot v_i > 0 \text{ e } s_2 \cdot v_i \leq 0\}$
5. $A_3 \leftarrow \{i \in V(G) : s_1 \cdot v_i \leq 0 \text{ e } s_2 \cdot v_i > 0\}$
6. $A_4 \leftarrow \{i \in V(G) : s_1 \cdot v_i \leq 0 \text{ e } s_2 \cdot v_i \leq 0\}$
7. **retorne** (A_1, A_2, A_3, A_4)

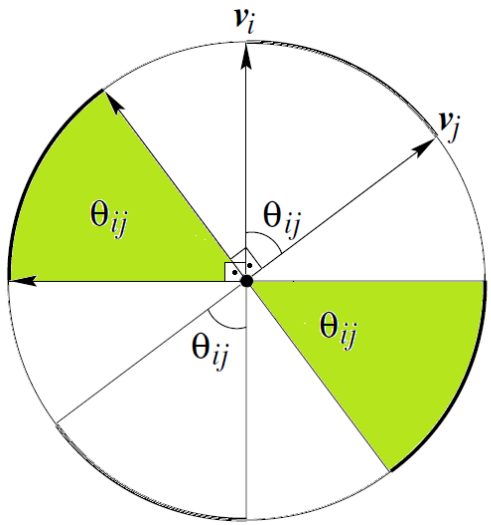
Lema: $\mathbb{P}(s_1 \cdot v_i > 0 \text{ e } s_1 \cdot v_j \leq 0) = \arccos(v_i \cdot v_j)/2\pi$

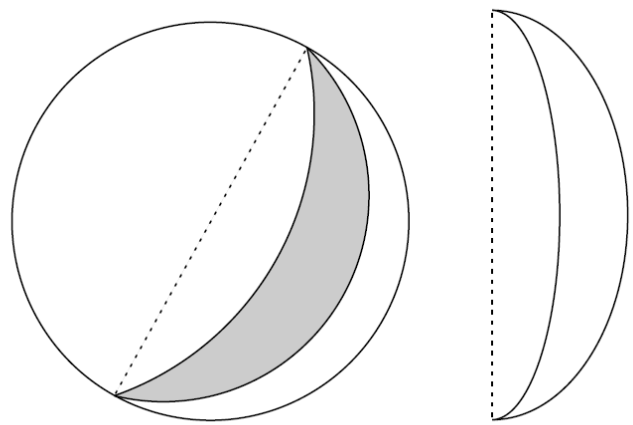
Prova: $\mathbb{P}(s_1 \cdot v_i > 0, s_1 \cdot v_j \leq 0) = \frac{\Theta_{ij}}{2\pi}$, onde Θ_{ij} é o ângulo entre v_i e v_j (deve $\in$ região esfera interseção dos 2 semiespaços). $\Theta_{ij} = \arccos(v_i \cdot v_j)$, pois $v_i \cdot v_j = |v_i| \cdot |v_j| \cdot \cos(\Theta_{ij})$.

Algoritmo MaxCluster-PS (Programação Semidefinida)

Algoritmo MaxCluster-PS(G, w^+, w^-)

1. **sejam** $v_1, \ldots, v_n$ uma solução ótima da Relaxação vetorial acima
2. **sejam** $s_1, s_2 \leftarrow$ *RandEsfera*(n) (vetores aleat. tam. n e módulo 1)
3. $A_1 \leftarrow \{i \in V(G) : s_1 \cdot v_i > 0 \;\; e \;\; s_2 \cdot v_i > 0\}$
4. $A_2 \leftarrow \{i \in V(G) : s_1 \cdot v_i > 0 \;\; e \;\; s_2 \cdot v_i \leq 0\}$
5. $A_3 \leftarrow \{i \in V(G) : s_1 \cdot v_i \leq 0 \;\; e \;\; s_2 \cdot v_i > 0\}$
6. $A_4 \leftarrow \{i \in V(G) : s_1 \cdot v_i \leq 0 \;\; e \;\; s_2 \cdot v_i \leq 0\}$
7. **retorne** (A_1, A_2, A_3, A_4)





Uma "fatia" de uma esfera no $\mathbb{R}^3$

Algoritmo MaxCluster-PS (Programação Semidefinida)

Algoritmo MaxCluster-PS(G, w^+, w^-)

1. **sejam** $v_1, \ldots, v_n$ uma solução ótima da Relaxação vetorial acima
2. **sejam** $s_1, s_2 \leftarrow$ *RandEsfera*(n) (vetores aleat. tam. n e módulo 1)
3. $A_1 \leftarrow \{i \in V(G) : s_1 \cdot v_i > 0 \text{ e } s_2 \cdot v_i > 0\}$
4. $A_2 \leftarrow \{i \in V(G) : s_1 \cdot v_i > 0 \text{ e } s_2 \cdot v_i \leq 0\}$
5. $A_3 \leftarrow \{i \in V(G) : s_1 \cdot v_i \leq 0 \text{ e } s_2 \cdot v_i > 0\}$
6. $A_4 \leftarrow \{i \in V(G) : s_1 \cdot v_i \leq 0 \text{ e } s_2 \cdot v_i \leq 0\}$
7. **retorne** (A_1, A_2, A_3, A_4)

Lema: $\mathbb{P}(i, j \text{ no mesmo cluster}) = (1 - \arccos(v_i \cdot v_j)/\pi)^2$

Prova: Para estarem no mesmo cluster, devem estar do mesmo lado tanto em relação a s_1 quanto em relação a s_2.

Lema: $\mathbb{P}(i, j \text{ em clusters dif.}) = 1 - (1 - \arccos(v_i \cdot v_j)/\pi)^2$

Algoritmo MaxCluster-PS é 0.75-aprox. prob.

- **Maximizar** $\sum_{ij \in E(G)} \left(w_{ij}^{+}(v_i \cdot v_j) + w_{ij}^{-}(1 - v_i \cdot v_j) \right)$, **restrito a**:
- Para cada $i, j \in V(G)$: $\quad v_i \cdot v_i = 1 \quad$ e $\quad v_i \cdot v_j \geq 0$

Teorema: MaxCluster-PS é 0.75-aprox. probabilístico

Prova: Seja X o peso $w(P)$ da partição retornada pelo algoritmo. Seja $X_{ij} = 1$ se i e j estão no mesmo cluster, e 0, cc.

$$\mathbb{E}(X) \;=\; \sum_{ij \in E(G)} \left[w_{ij}^{+} \cdot \mathbb{E}(X_{ij}) + w_{ij}^{-} \cdot (1 - \mathbb{E}(X_{ij}) \right] \;=$$

$$= \sum_{ij \in E(G)} \left[w_{ij}^{+} \cdot \left(1 - \frac{arccos(v_i v_j)}{\pi} \right)^2 + w_{ij}^{-} \cdot \left(1 - \left(1 - \frac{arccos(v_i v_j)}{\pi} \right)^2 \right) \right]$$

$$\Rightarrow \; \mathbb{E}(X) \;\geq\; \sum_{ij \in E(G)} \left[w_{ij}^{+} \cdot 0.75(v_i v_j) \;+\; w_{ij}^{-} \cdot 0.75(1 - v_i v_j) \right] \;\geq\; 0.75 \cdot opt$$

Algoritmo MaxCluster-PS é 0.75-aprox. prob.

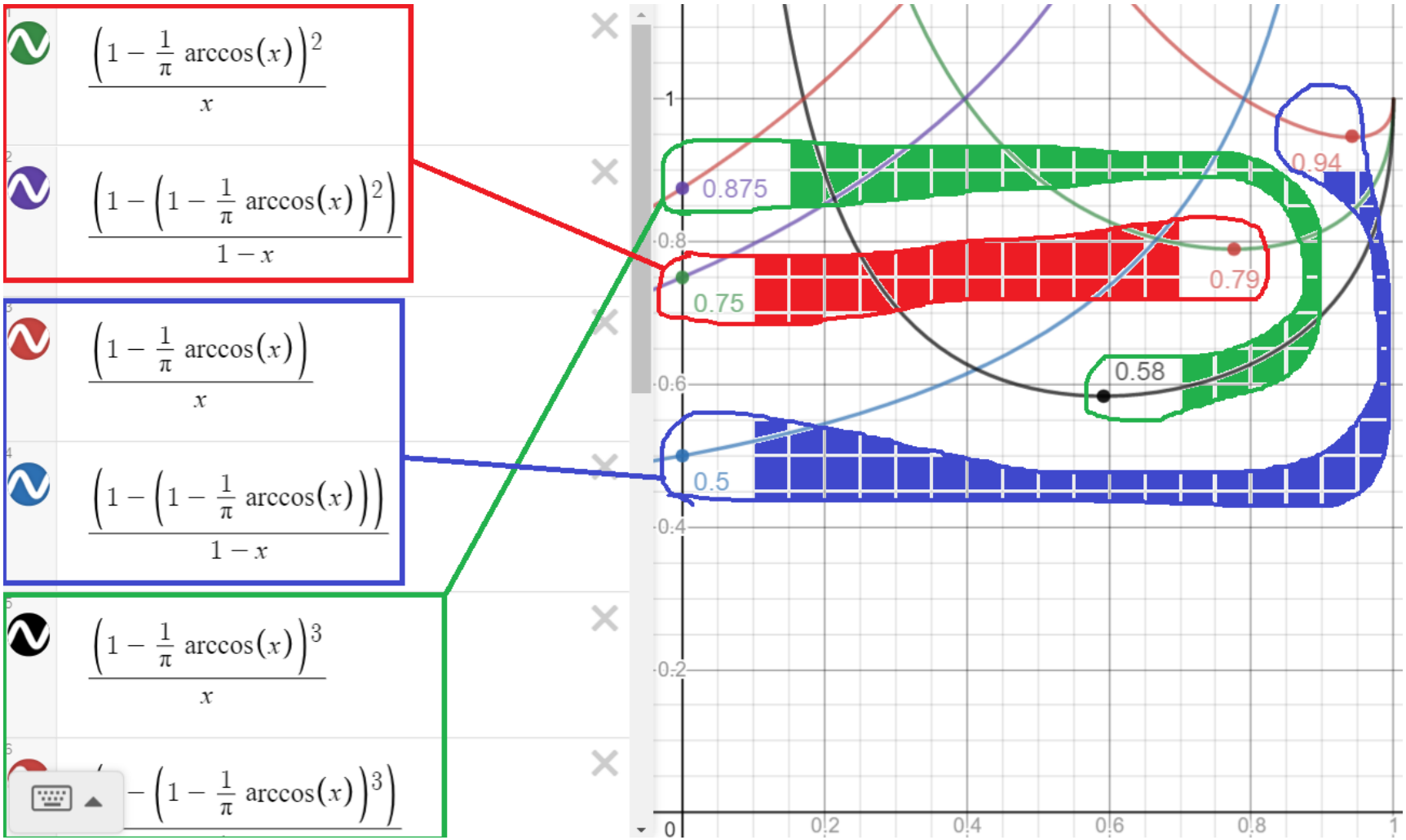


Figure: Gráficos no Desmos: fator aprox. $= 0.75$