



中南大學

CENTRAL SOUTH UNIVERSITY

基于核方法的广义中位数 计算算法

黄道煜

2024年1月19日

目录

1. 广义中位数
2. Weiszfeld 算法
3. 核方法
4. 基于核方法的广义中位数计算算法
5. 效果评估

广义中位数

给定定义域 \mathcal{O} , 对象集合 $O = \{o_1, o_2, \dots, o_n\}$, 距离函数 $\delta(o_i, o_j)$

广义中位数:

$$\bar{o} = \arg \min_{o \in \mathcal{O}} \sum_{o_i \in O} \delta(o_i, o)$$

广义中位数

给定定义域 \mathcal{O} , 对象集合 $O = \{o_1, o_2, \dots, o_n\}$, 距离函数 $\delta(o_i, o_j)$
广义中位数:

$$\bar{o} = \arg \min_{o \in \mathcal{O}} \sum_{o_i \in O} \delta(o_i, o)$$

很多问题都可以归类为广义中位数:
拉普拉斯分布和高斯分布的最大似然估计
欧氏空间的几何中位数
斯坦纳字符串
肯尼共识问题

广义中位数

对于未知对象 \bar{o} 的观测结果 $\{o_1, o_2, \dots, o_n\}$ 若满足拉普拉斯分布

$$\mathcal{L}(o|\bar{o}, \sigma) = \frac{1}{2\sigma} \exp\left(-\frac{\delta(o, \bar{o})}{\sigma}\right)$$

\bar{o} 的最大似然估计即为

$$\bar{o} = \arg \min_{o \in \mathcal{O}} \sum_{i=1}^n \delta(o_i, o)$$

广义中位数

对于未知对象 \bar{o} 的观测结果 $\{o_1, o_2, \dots, o_n\}$ 若满足高斯分布

$$\mathcal{N}(o|\bar{o}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\delta^2(o, \bar{o})}{2\sigma^2}\right)$$

\bar{o} 的最大似然估计即为

$$\bar{o} = \arg \min_{o \in \mathcal{O}} \sum_{i=1}^n \delta^2(o_i, o)$$

Weiszfeld 算法

求解欧氏空间的几何中位数的算法

点集 x_1, \dots, x_n , 初始解 y_0

迭代计算

$$y_{j+1} = \frac{\sum_{i=1}^n w_i^j x_i}{\sum_{i=1}^n w_i^j}$$

其中 $w_i^j = \frac{1}{\|y_j - x_i\|}$

Weiszfeld 算法

推导过程

对目标函数 $f(y) = \sum_{i=1}^n \|y - x_i\|$ 求梯度，其中梯度的第 j 个分量为：

$$\frac{\partial f}{\partial y_j} = \sum_{i=1}^n \frac{y_j - x_{ij}}{\|y - x_i\|}$$

将 y 的每个分量减去对应梯度得到 y' ，有：

$$y' = \left(\sum_{i=1}^n \frac{x_{i1}}{\|y - x_i\|}, \dots, \sum_{i=1}^n \frac{x_{im}}{\|y - x_i\|} \right) = \sum_{i=1}^n \frac{x_i}{\|y - x_i\|}$$

Weiszfeld 算法

算法本质是梯度下降法

欧氏空间中目标函数是凸函数，因此一定会收敛于全局最优解

核方法

核函数：对于二元函数 $K: R^d \times R^d \rightarrow R$ ，如果存在某个希尔伯特空间 \mathcal{H}_k 和函数 $\phi: R^d \rightarrow \mathcal{H}_k$ ，使得 $K(x, y) = \langle \phi(x), \phi(y) \rangle$ ，则称该函数为核函数。 \mathcal{H}_k 被称为特征空间， ϕ 被称为特征映射。

核方法：将目标通过特征映射 $\phi: R^d \rightarrow \mathcal{H}_k$ 转换到特征空间中，通过核函数 $K: R^d \times R^d \rightarrow R$ 在特征空间中求解问题。

核方法

核方法：将对象通过特征映射 $\phi: R^d \rightarrow \mathcal{H}_k$ 转换到特征空间中，通过核函数 $K: R^d \times R^d \rightarrow R$ 在特征空间中求解问题。

在特征空间中，可以根据内积的定义诱导出距离（模长）的概念：

$$\begin{aligned}\|\phi(o_a) - \phi(o_b)\|_2 &= (\langle \phi(o_a) - \phi(o_b), \phi(o_a) - \phi(o_b) \rangle)^{\frac{1}{2}} \\ &= (\langle \phi(o_a), \phi(o_a) \rangle - 2\langle \phi(o_a), \phi(o_b) \rangle \\ &\quad + \langle \phi(o_b), \phi(o_b) \rangle)^{\frac{1}{2}} \\ &= (K(o_a, o_a) - 2K(o_a, o_b) + K(o_b, o_b))^{\frac{1}{2}} \quad (3)\end{aligned}$$

该距离在核方法中一般能反映出对象之间的相似度

核方法

核方法：将对象通过特征映射 $\phi: R^d \rightarrow \mathcal{H}_k$ 转换到特征空间中，通过核函数 $K: R^d \times R^d \rightarrow R$ 在特征空间中求解问题。

核方法优势：

- 特征空间中目标集合更可能满足线性可分性
- 可以通过核函数避免对特征映射和特征空间中的坐标进行显式计算
- 可以通过核函数将树、图、字符串等异构对象纳入统一的处理框架

核方法

核方法：将对象通过特征映射 $\phi: R^d \rightarrow \mathcal{H}_k$ 转换到特征空间中，通过核函数 $K: R^d \times R^d \rightarrow R$ 在特征空间中求解问题。

例子：

String Subsequence Kernel $\phi^{ssk}(s) = (\phi_{u_1}(s), \phi_{u_2}(s), \dots), \quad \phi_u(s) = \sum_{\mathbf{i}: u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})}$

每个维度代表一个字符串 u_i ，值表示 u_i 在 s 中出现的频率和紧密度
核函数 $K(s_a, s_b)$ 表示两个字符串间的相似度

基于核方法的广义中位数计算算法

1. 显式转换算法
2. 隐式转换算法
3. 对于非正定核的扩展算法

基于核方法的广义中位数计算算法

显式转换算法

1. 计算输入对象在特征空间中的映射
2. 使用Weiszfeld算法计算特征空间中的几何中位数

$$\bar{x}_{j+1} = \frac{\sum_{i=1}^n \omega_i^j \phi(o_i)}{\sum_{i=1}^n \omega_i^j} \quad (4)$$

$$\omega_i^j = \frac{1}{\|\bar{x}_j - \phi(o_i)\|_2} \quad (5)$$

\bar{x}_j 表示第 j 次迭代的计算结果
令 $w_i^0 = 1$, 因此 \bar{x}_0 为几何平均数

3. 根据特征空间中的几何中位数构造近似解

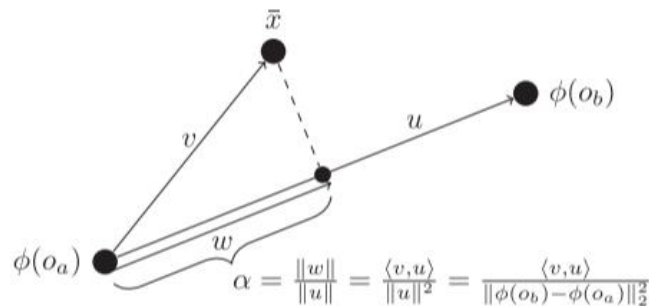


Fig. 1. Computation of the ratio α between objects for reconstruction.

找出距离几何中位数最近的 o_a, o_b
并计算 α
在问题空间中根据 o_a, o_b, α 构造近似解

基于核方法的广义中位数计算算法

隐式转换算法

$$\bar{x}_{j+1} = \frac{\sum_{i=1}^n \omega_i^j \phi(o_i)}{\sum_{i=1}^n \omega_i^j} \quad (4)$$

$$\omega_i^j = \frac{1}{\|\bar{x}_j - \phi(o_i)\|_2} \quad (5)$$

\bar{x} 是 $\phi(o_i)$ 的加权平均，权值是距离的反比

权值可以由核函数计算出，可以直接迭代计算 w_i^j

算法收敛时，权值越大的点距几何中位数越近

可以根据权值得到距中位数最近的对象

计算 α 只涉及点积运算，也可以由核函数计算出

基于核方法的广义中位数计算算法

隐式转换算法：中位数求解

Algorithm 3. Kernel-Based Generalized Median Framework

Input: Object set O , distance function $\delta()$, weighted mean function $wm()$, kernel function $K()$

Output: Median object \bar{o}

/ Computation of median weights */*

1: Initialize $\omega_i^0 = 1$ for all $1 \leq i \leq |O|$

2: **for** $j = 1$ to j_{max} **do**

3: **for** $i = 1$ to n **do**

4: Compute ω_i^j using Eq. (7), (8), and (9)

5: **end for**

6: **end for**

/ Reconstruction */*

7: Compute \bar{o} using reconstruction Algorithms 1 or 2

8: **return** \bar{o} with the related SOD

$$\omega_i^j = 1/(\langle \bar{x}_j, \bar{x}_j \rangle - 2\langle \bar{x}_j, \phi(o_i) \rangle + \langle \phi(o_i), \phi(o_i) \rangle)^{\frac{1}{2}} \quad (7)$$

$$\begin{aligned} \langle \bar{x}_j, \bar{x}_j \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}} \right\rangle \\ &= \frac{\sum_{u=1}^n \sum_{v=1}^n \omega_u^{j-1} \omega_v^{j-1} K(o_u, o_v)}{\left(\sum_{u=1}^n \omega_u^{j-1} \right)^2} \end{aligned} \quad (8)$$

$$\begin{aligned} \langle \bar{x}_j, \phi(o_i) \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \phi(o_i) \right\rangle \\ &= \frac{\sum_{u=1}^n \omega_u^{j-1} K(o_u, o_i)}{\sum_{u=1}^n \omega_u^{j-1}} \end{aligned} \quad (9)$$

基于核方法的广义中位数计算算法

隐式转换算法：重构

Algorithm 1. Kernel-Based Linear ($L = 2$) and Triangular ($L = 3$) Reconstruction

Input: Object set O , integer l , final weights ω_i , weighted mean function $wm()$

Output: Median object \bar{o}

- 1: Select o_1, \dots, o_l with the l maximal ω_i
 - 2: $\bar{o}_1 = o_1$
 - 3: **for** $j = 2$ to l **do**
 - 4: Compute α using Eq. (11) with objects \bar{o}_{j-1}, o_j
 - 5: $\bar{o}_j = wm(\bar{o}_{j-1}, o_j, \alpha)$
 - 6: **end for**
 - 7: **return** \bar{o}_l with the related SOD
-

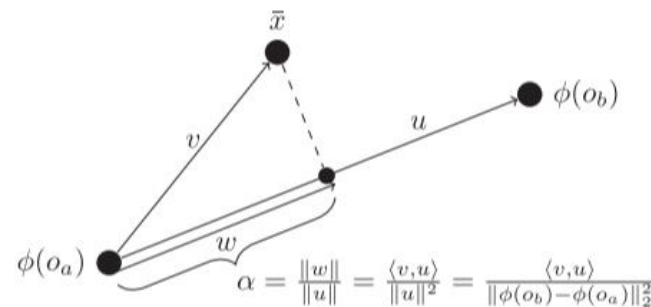


Fig. 1. Computation of the ratio α between objects for reconstruction.

$$\begin{aligned}
 \alpha &= \frac{\left\langle \frac{\sum_{i=1}^n \omega_i \phi(o_i)}{\sum_{i=1}^n \omega_i} - \phi(o_a), \phi(o_b) - \phi(o_a) \right\rangle}{\|\phi(o_b) - \phi(o_a)\|^2} \\
 &= \frac{\frac{\sum_{i=1}^n \omega_i (K(o_i, o_b) - K(o_i, o_a))}{\sum_{i=1}^n \omega_i} - K(o_a, o_b) + K(o_a, o_a)}{K(o_b, o_b) - 2K(o_b, o_a) + K(o_a, o_a)} \quad (11)
 \end{aligned}$$

基于核方法的广义中位数计算算法

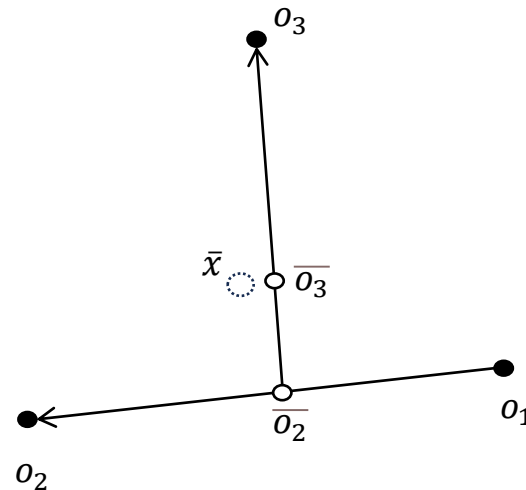
隐式转换算法：重构

Algorithm 1. Kernel-Based Linear ($L = 2$) and Triangular ($L = 3$) Reconstruction

Input: Object set O , integer l , final weights ω_i , weighted mean function $wm()$

Output: Median object \bar{o}

- 1: Select o_1, \dots, o_l with the l maximal ω_i
 - 2: $\bar{o}_1 = o_1$
 - 3: **for** $j = 2$ to l **do**
 - 4: Compute α using Eq. (11) with objects \bar{o}_{j-1}, o_j
 - 5: $\bar{o}_j = wm(\bar{o}_{j-1}, o_j, \alpha)$
 - 6: **end for**
 - 7: **return** \bar{o}_l with the related SOD
-



$$\begin{aligned} \alpha &= \frac{\left\langle \frac{\sum_{i=1}^n \omega_i \phi(o_i)}{\sum_{i=1}^n \omega_i} - \phi(o_a), \phi(o_b) - \phi(o_a) \right\rangle}{\|\phi(o_b) - \phi(o_a)\|^2} \\ &= \frac{\frac{\sum_{i=1}^n \omega_i (K(o_i, o_b) - K(o_i, o_a))}{\sum_{i=1}^n \omega_i} - K(o_a, o_b) + K(o_a, o_a)}{K(o_b, o_b) - 2K(o_b, o_a) + K(o_a, o_a)} \quad (11) \end{aligned}$$

基于核方法的广义中位数计算算法

隐式转换算法：重构

Algorithm 2. Kernel-Based Linear (Triangular) Recursive Reconstruction

Input: Object set O , final weights ω_i , weighted mean function $wm()$

Output: Median object \bar{o}

```
1:  $\bar{o}_{best} = \emptyset$ 
2: while  $|O| > 1$  do
3:   Divide  $O$  into  $|O|/2$  pairs ( $|O|/3$  triples) by grouping maximal  $\omega_i$  first.
4:    $O' = \emptyset$ 
5:   for each pair  $(o_a, o_b)$  (triple  $(o_a, o_b, o_c)$ ) do
6:     Compute  $\bar{o}$  using Algorithm 1 with  $l = 2$  ( $l = 3$ )
7:      $O' = O' \cup \{\bar{o}\}$ 
8:   end for
9:    $\bar{o}_{best} = \arg \min_{o \in \{\bar{o}_{best}\} \cup O'} SOD(o)$ 
10:   $O = O'$ 
11: end while
12: return  $\bar{o}_{best}$  with the related SOD
```

将对象集合 O 分组
每一组根据算法1计算出近似解 \bar{o}
将所有 \bar{o} 加入 O'
 O' 为下次迭代的对象集合

基于核方法的广义中位数计算算法

隐式转换算法：对于非正定核的扩展算法

二元函数正定性：二元函数 $K: R^d \times R^d \rightarrow R$ 满足正定性，当前仅当对于任意 n ，任意向量组 x_1, \dots, x_n ，任意非零向量 a ，都满足 $\sum_{i=1}^n \sum_{j=1}^n a_i a_j K(x_i, x_j) > 0$ 。即对于任意向量组 x_1, \dots, x_n ，矩阵 $[K(x_i, x_j)]_{i,j=1}^n$ 都满足正定性。

性质：二元函数 $K: R^d \times R^d \rightarrow R$ 是核函数，当且仅当该函数满足正定性。

基于核方法的广义中位数计算算法

隐式转换算法：对于非正定核的扩展算法

$$\omega_i^j = \frac{1}{\|\bar{x}_j - \phi(o_i)\|_2} \quad (5)$$

$$\omega_i^j = 1/(\langle \bar{x}_j, \bar{x}_j \rangle - 2\langle \bar{x}_j, \phi(o_i) \rangle + \langle \phi(o_i), \phi(o_i) \rangle)^{\frac{1}{2}} \quad (7)$$

$$\begin{aligned} \langle \bar{x}_j, \bar{x}_j \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}} \right\rangle \\ &= \frac{\sum_{u=1}^n \sum_{v=1}^n \omega_u^{j-1} \omega_v^{j-1} K(o_u, o_v)}{\left(\sum_{u=1}^n \omega_u^{j-1}\right)^2} \end{aligned} \quad (8)$$

$$\begin{aligned} \langle \bar{x}_j, \phi(o_i) \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \phi(o_i) \right\rangle \\ &= \frac{\sum_{u=1}^n \omega_u^{j-1} K(o_u, o_i)}{\sum_{u=1}^n \omega_u^{j-1}} \end{aligned} \quad (9)$$

推导过程中向量运算需要改写成复数形式：

$$u \cdot v = \sum_{i=1}^n u_i \bar{v}_i$$

$$\|u\| = \sqrt{\sum_{i=1}^n |u_i|^2}$$

复向量运算满足如下性质：

$$\langle u, v \rangle = \overline{\langle v, u \rangle}$$

$$\langle u, \lambda v \rangle = \bar{\lambda} \langle u, v \rangle$$

基于核方法的广义中位数计算算法

隐式转换算法：对于非正定核的扩展算法

$$\omega_i^j = \frac{1}{\|\bar{x}_j - \phi(o_i)\|_2} \quad (5)$$

$$\omega_i^j = 1/(\langle \bar{x}_j, \bar{x}_j \rangle - 2\langle \bar{x}_j, \phi(o_i) \rangle + \langle \phi(o_i), \phi(o_i) \rangle)^{\frac{1}{2}} \quad (7)$$

$$\begin{aligned} \langle \bar{x}_j, \bar{x}_j \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}} \right\rangle \\ &= \frac{\sum_{u=1}^n \sum_{v=1}^n \omega_u^{j-1} \omega_v^{j-1} K(o_u, o_v)}{\left(\sum_{u=1}^n \omega_u^{j-1}\right)^2} \end{aligned} \quad (8)$$

$$\begin{aligned} \langle \bar{x}_j, \phi(o_i) \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \phi(o_i) \right\rangle \\ &= \frac{\sum_{u=1}^n \omega_u^{j-1} K(o_u, o_i)}{\sum_{u=1}^n \omega_u^{j-1}} \end{aligned} \quad (9)$$

推导过程中向量运算需要改写成复数形式：

$$\begin{aligned} \omega_i^j &= 1/(\langle \bar{x}_j, \bar{x}_j \rangle - \langle \bar{x}_j, \phi(o_i) \rangle \\ &\quad - \overline{\langle \bar{x}_j, \phi(o_i) \rangle} + \langle \phi(o_i), \phi(o_i) \rangle)^{\frac{1}{2}} \end{aligned} \quad (14)$$

$$\begin{aligned} \langle \bar{x}_j, \bar{x}_j \rangle &= \left\langle \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}}, \frac{\sum_{u=1}^n \omega_u^{j-1} \phi(o_u)}{\sum_{u=1}^n \omega_u^{j-1}} \right\rangle \\ &= \frac{\sum_{u=1}^n \sum_{v=1}^n \omega_u^{j-1} \overline{\omega_v^{j-1}} K(o_u, o_v)}{\left(\sum_{u=1}^n \omega_u^{j-1}\right) \left(\sum_{u=1}^n \omega_u^{j-1}\right)} \end{aligned} \quad (15)$$

基于核方法的广义中位数计算算法

核函数：正定核函数

String Subsequence Kernel $\phi^{ssk}(s) = (\phi_{u_1}(s), \phi_{u_2}(s), \dots), \quad \phi_u(s) = \sum_{\mathbf{i}: u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})}$

每个维度代表一个字符串 u_i ，值表示 u_i 在 s 中出现的频率和紧密度

核函数 $K(s_a, s_b)$ 表示两个字符串间的相似度

基于核方法的广义中位数计算算法

核函数：非正定核函数

Distance Substitution Kernels $K_{\delta}^{lin}(o_a, o_b) = \frac{1}{2} \left(\delta(o_a, o)^2 + \delta(o_b, o)^2 - \delta(o_a, o_b)^2 \right)$

其中 o 为参照对象

核函数 $K(o_a, o_b)$ 就等于 o_a, o_b 在问题空间中的距离

效果评估

Benchmark Algorithm:

- Prototype-Embedding

p_1, p_2, \dots, p_d 为 n 个输入对象中的特定 d 个，映射函数 $\phi(o) = (\delta(o, p_1), \delta(o, p_2), \dots, \delta(o, p_d))$

- Curvilinear Component Analysis(CCA)

x_1, x_2, \dots, x_d 为欧氏空间的 d 个点， F 为有界的单调递减函数， λ_t 是依赖迭代次数的序列。
使用神经网络迭代调整 x_1, x_2, \dots, x_d 最小化函数 $E = \frac{1}{2} \sum_i \sum_{j \neq i} (\delta(o_i, o_j) - \delta_e(x_i, x_j))^2 F(\delta_e(x_i, x_j), \lambda_t)$

函数的目的是尽可能保留对象映射后的距离关系。CCA通过函数 F 控制迭代过程中优化的重点，最初迭代时距离较大的点对会有较高的权重，随后后距离较小点对的影响会逐渐增大。

效果评估

算法理论比较:

- 对于 PE 等存在显式映射函数的算法，可以与核方法等价。但映射后的向量只是对象的特征值，这些点在特征空间内的距离无实际意义。距离保持核能够保留所有距离信息。但因为特征空间中向量加减运算不能通过核函数计算，重构时只能使用比例信息。
- 对于 CCA 等不存在映射函数的算法，在特征空间中保留了一定的距离信息。虽然距离信息有一定损失，但可以设计更灵活准确的重构方法。

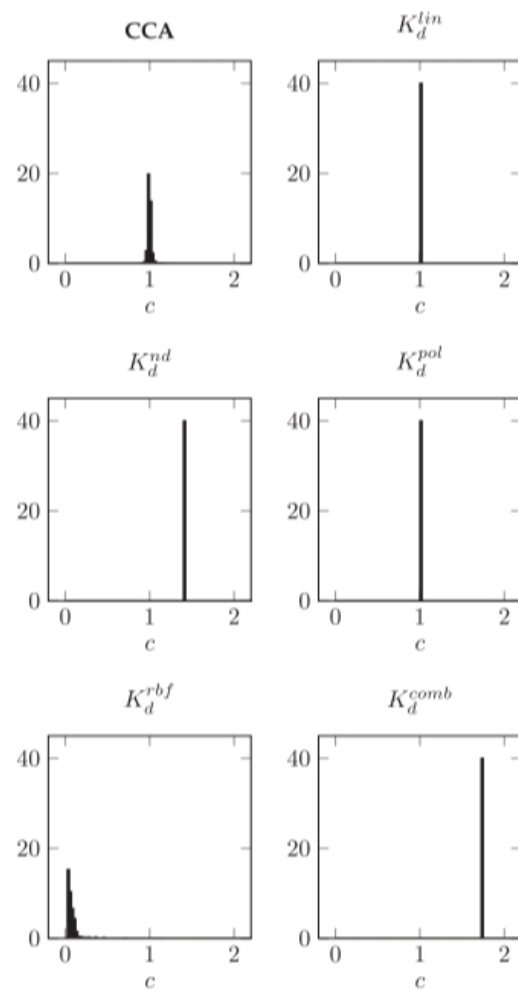
对于只需要特征空间中距离的算法，核方法（距离保持核）最适合。

效果评估

距离保持性：核函数得出的距离与欧氏距离的比例保持不变

$$\delta(o_i, o_j) = c \cdot \|\phi(o_i) - \phi(o_j)\|_2 \quad (17)$$

论文中的常用核函数只有 K_d^{rbf} 不满足距离保持性



效果评估

Weiszfeld算法收敛速度

每列左边数字代表非正定核， 右边代表正定核

论文中说算法在非正定核中的收敛性待证明。
目标函数在正定核空间中能够保持凸性， 梯度在正定核中也能够更好保持（待验证）

算法的迭代次数可以看成常数， 整个框架的时间复杂度为 $O(n^2)$

TABLE 3
Convergence of the Weiszfeld Algorithm for 5 Indefinite Kernel Functions K_{δ}^{lin} , K_{δ}^{nd} , K_{δ}^{pol} , K_{δ}^{rbf} , K_{δ}^{comb} (First Value in Each Column) and 3 Positive Definite Kernel Functions K^{ssk} , K^{part} , K^{kend} (Second Value in Each Column).

Dataset (num sets)	max-iter	med-iter	complex weights
Darwin (36)	29 / 13	9 / 11	28/0 (of 180)
CCD (22)	17 / 33	12 / 24	0/0 (of 110)
Gen-Cluster (8)	14 / 12	10 / 10	0/0 (of 40)
UCI-Cluster (8)	27 / 19	20 / 15	0/0 (of 40)
ranking-bio (40)	144 / 85	42 / 76	2/0 (of 200)
ranking-real (40)	31 / 23	17 / 17	0/0 (of 200)
random-string (1000)	11 / 11	10 / 10	0/0 (of 5000)
random-cluster (1000)	14 / 11	12 / 10	0/0 (of 5000)
random-ranking (1000)	8 / 9	6 / 8	0/0 (of 5000)

效果评估

Weiszfeld算法收敛速度:

每列左边数字代表非正定核，右边代表正定核

论文中说算法在非正定核中的收敛性待证明。
目标函数在正定核空间中能够保持凸性，梯度在正定核中也能够更好保持（待验证）

算法的迭代次数可以看成常数，整个框架的时间复杂度为 $O(n^2)$

TABLE 3
Convergence of the Weiszfeld Algorithm for 5 Indefinite Kernel Functions $K_{\delta}^{lin}, K_{\delta}^{nd}, K_{\delta}^{pol}, K_{\delta}^{rbf}, K_{\delta}^{comb}$ (First Value in Each Column) and 3 Positive Definite Kernel Functions $K^{ssk}, K^{part}, K^{kend}$ (Second Value in Each Column).

Dataset (num sets)	max-iter	med-iter	complex weights
Darwin (36)	29 / 13	9 / 11	28/0 (of 180)
CCD (22)	17 / 33	12 / 24	0/0 (of 110)
Gen-Cluster (8)	14 / 12	10 / 10	0/0 (of 40)
UCI-Cluster (8)	27 / 19	20 / 15	0/0 (of 40)
ranking-bio (40)	144 / 85	42 / 76	2/0 (of 200)
ranking-real (40)	31 / 23	17 / 17	0/0 (of 200)
random-string (1000)	11 / 11	10 / 10	0/0 (of 5000)
random-cluster (1000)	14 / 11	12 / 10	0/0 (of 5000)
random-ranking (1000)	8 / 9	6 / 8	0/0 (of 5000)

效果评估

效果对比:

CCA为之前广义中位数研究中表现较好的算法

Prototype为较经典的广义中位数算法

五个常用核函数中有四个的表现优于之前的算法
(K_{δ}^{lin} , K_{δ}^{nd} , K_{δ}^{pol} and K_{δ}^{comb})

(a) Darwin dataset					
	linear	triangular	lin-rec	triang-rec	lin-search
K_{δ}^{lin}	0.3683*	0.3090*	0.2070+	0.1969*	0.1056*
K_{δ}^{nd}	0.3683+	0.3090*	0.2070+	0.1969*	0.1056*
K_{δ}^{pol}	0.3683*	0.3090*	0.2070+	0.1969*	0.1056*
K_{δ}^{rbf}	0.8284	0.8284	0.8284	0.8284	0.1925
K_{δ}^{comb}	0.3683+	0.3090*	0.2070+	0.2041*	0.1056*
K_{δ}^{ssk}	0.4107	0.3682	0.4023	0.2932	0.1487
CCA	0.3691	0.3144	0.2425	0.1988	0.0988
Prototype	0.6411	0.5705	0.2357*	0.2912	0.1275

(b) CCD dataset					
	linear	triangular	lin-rec	triang-rec	lin-search
K_{δ}^{lin}	0.3103*	0.3024*	0.2693+	0.2769*	0.2330+
K_{δ}^{nd}	0.3103*	0.3024*	0.2693+	0.2769*	0.2330+
K_{δ}^{pol}	0.3103*	0.3024*	0.2693+	0.2769*	0.2330+
K_{δ}^{rbf}	0.6376	0.6376	0.6376	0.6376	0.2948
K_{δ}^{comb}	0.3103*	0.3024*	0.2693+	0.2769*	0.2330+
K_{δ}^{ssk}	0.4513	0.4455	0.4829	0.4560	0.2970
CCA	0.3177	0.3177	0.2812	0.2835	0.2590
Prototype	0.4693	0.4241	0.3001	0.2944	0.2452*

(c) Gen. Cluster dataset					
	linear	triangular	lin-rec	triang-rec	lin-search
K_{δ}^{lin}	0.4575*	0.4567+	0.4412+	0.4567+	0.4075*
K_{δ}^{nd}	0.4575*	0.4567+	0.4412+	0.4567*	0.4098*
K_{δ}^{pol}	0.4575*	0.4567+	0.4412+	0.4567+	0.4075*
K_{δ}^{rbf}	0.6968	0.6968	0.6968	0.6968	0.4560*
K_{δ}^{comb}	0.4575*	0.4567*	0.4412+	0.4428+	0.4098*
K_{δ}^{part}	0.4583	0.4583	0.4412*	0.4583	0.4098*
CCA	0.4575	0.4567	0.4575	0.4543	0.4183
Prototype	0.6055	0.5706	0.4966	0.4862	0.4176*

(d) UCI Cluster dataset					
	linear	triangular	lin-rec	triang-rec	lin-search
K_{δ}^{lin}	0.3055+	0.3055+	0.3055*	0.3055+	0.2844*
K_{δ}^{nd}	0.3055+	0.3055+	0.3055*	0.3055+	0.2844*
K_{δ}^{pol}	0.3055+	0.3055+	0.3055*	0.3055+	0.2844*
K_{δ}^{rbf}	1.4435	1.4435	1.4435	1.4435	0.2907*
K_{δ}^{comb}	0.3055+	0.3055+	0.3055*	0.3055+	0.2844*
K_{δ}^{part}	0.3055*	0.3055*	0.3055*	0.3055*	0.2844+
CCA	0.3078	0.3078	0.3078	0.3078	0.2929
Prototype	0.4385	0.4301	0.3321	0.3427	0.2964*

(e) ranking-bio dataset					
	linear	triangular	lin-rec	triang-rec	lin-search
K_{δ}^{lin}	0.1113	0.1113+	0.1113+	0.1113+	0.1058+
K_{δ}^{nd}	0.1113	0.1113+	0.1113+	0.1113+	0.1058+
K_{δ}^{pol}	0.1113	0.1113+	0.1113+	0.1113+	0.1058+
K_{δ}^{rbf}	0.1458	0.1458	0.1458	0.1458	0.1324
K_{δ}^{comb}	0.1113	0.1113+	0.1113+	0.1113+	0.1058+
K_{δ}^{kend}	0.1187	0.1187	0.1412	0.1187	0.1063
CCA	0.1113	0.1113	0.1113	0.1113	0.1087
Prototype	0.1113	0.1113	0.1113	0.1113	0.1087

(f) ranking-real dataset					
	linear	triangular	lin-rec	triang-rec	lin-search
K_{δ}^{lin}	0.2818+	0.2818*	0.2649+	0.2698+	0.2200+
K_{δ}^{nd}	0.2818+	0.2818*	0.2649+	0.2698+	0.2200+
K_{δ}^{pol}	0.2818+	0.2818*	0.2649+	0.2698+	0.2200+
K_{δ}^{rbf}	0.5647	0.5647	0.5647	0.5647	0.2662
K_{δ}^{comb}	0.2818+	0.2818*	0.2649+	0.2743+	0.2200+
K_{δ}^{kend}	0.3058	0.3058	0.2783	0.3058	0.2240*
CCA	0.2968	0.2812	0.2723	0.2831	0.2216
Prototype	0.3841	0.3826	0.3000	0.3203	0.2363