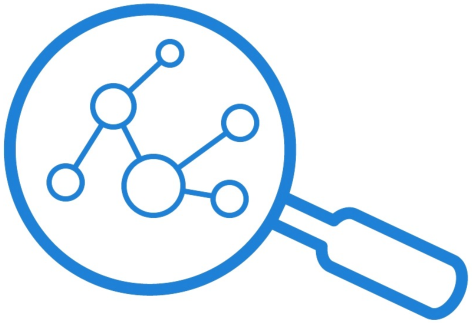
# 数据科学与大数据技术的数学基础

## 第五讲



计算机学院
余皓然

2024/5/9

# 课程内容

| |
|---|
| **Part1 随机化方法** |
| 一致性哈希 布隆过滤器 CM Sketch方法 最小哈希<br>欧氏距离下的相似搜索 Jaccard相似度下的相似搜索 |
| **Part2 谱分析方法** |
| 主成分分析 奇异值分解 谱图论 |
| **Part3 最优化方法** |
| 压缩感知 |

# 欧氏距离下的相似搜索

## 相似搜索问题及相似度衡量

# 相似搜索

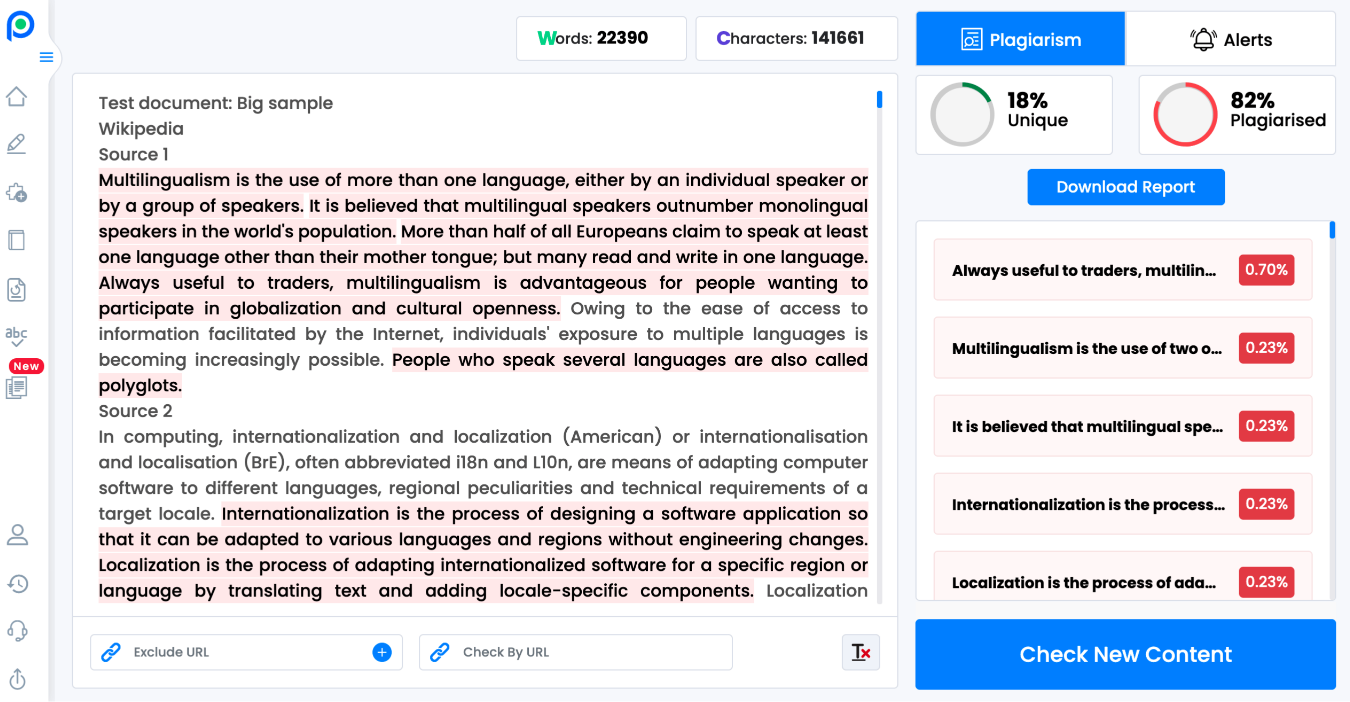
相似搜索（Similarity Search）：

- 给定数据集，如何找出其中相似度超过一定阈值的数据对（成对相似度搜索，All-Pairs Similarity Search）
- 给定数据集及一个新的数据，如何在数据集中找到最相似的数据（最近邻搜索，Nearest Neighbor Search）

# 相似搜索

应用场景：

- 相似文件/网页：代码/论文查重、检测镜像网站



# 相似搜索

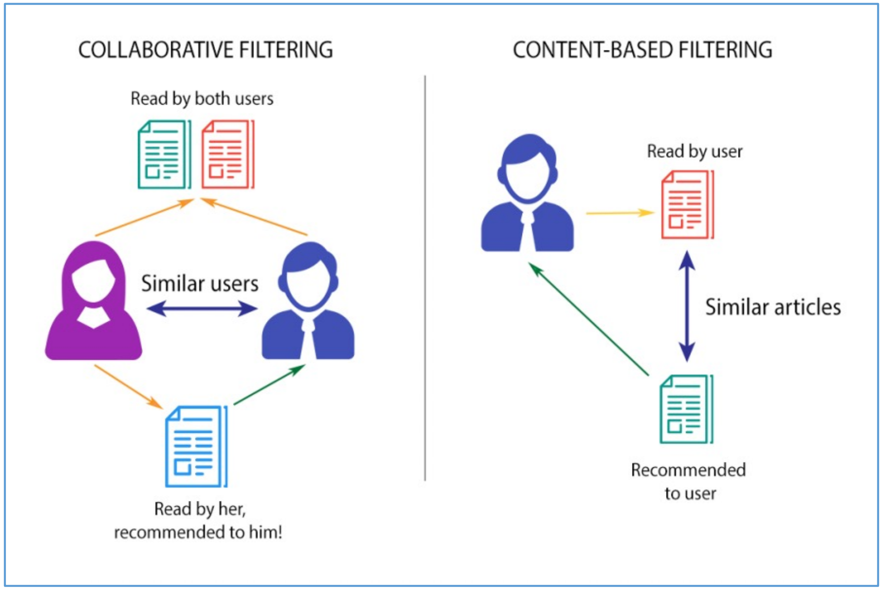
应用场景：

- 音频指纹：Shazam APP（根据输入的小段音频进行搜索）

# 相似搜索

应用场景:

- 协同过滤（collaborative filtering）：推荐系统中根据同一批用户浏览历史寻找相似产品；根据浏览及购买历史寻找相似用户




图片取自Cameron Musco <COMPSCI 514 - Algorithms for Data Science>

# 相似搜索

应用场景:

- 协同过滤（collaborative filtering）：微博/QQ/…的好友推荐功能，根据好友/关注情况分析用户之间的相似度

# 相似搜索

应用场景:

- 实体解析（Entity Resolution）：从不同源的数据集中识别哪些数据对应的是相同的人/物

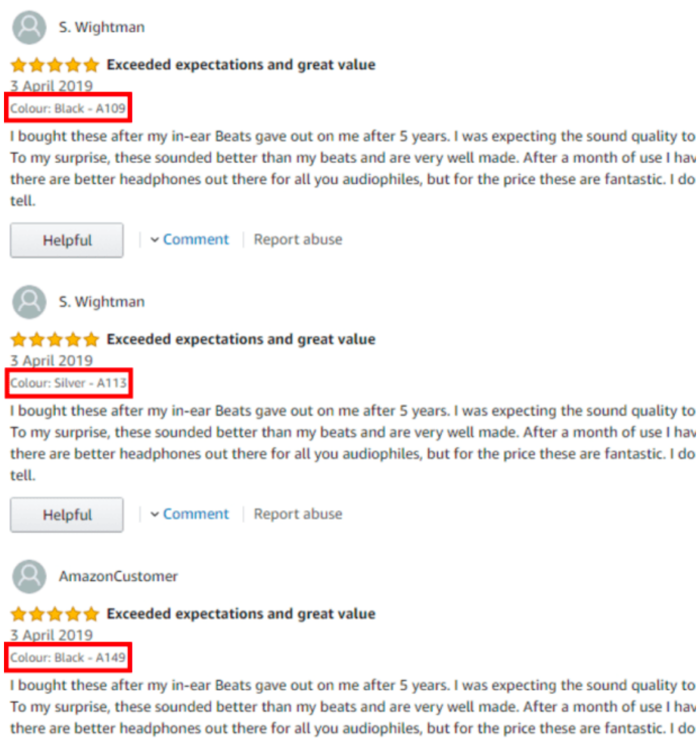
| ID | Name | Telephone | Address | Items Purchased |
|---|---|---|---|---|
| 233 | Angelica J. Jordan | 334-555-0178 | 111 Spring Ln, Greenville, AL | 5556, 7611 |
| 452 | Angie Jordan | 202-555-5477 | 45 Krakow St, Washington, DC | 2297 |
| 699 | Andrew Jordan | 334-555-0178 | 111 Spring Ln, Greenville, AL | 1185, 2299, 3720 |
| 720 | Angie Jrodon | | | 5556 |
| 821 | Angelica Jeffries Jordan | 202-555-5477 | 397 Hope Blvd, Greenville, AL | 7611 |

又如：合并多个天文望远镜对于同一个方向/天体的照片

# 相似搜索

应用场景:

- 垃圾/诈骗信息检测（Spam/Fraud Detection）：电商平台检测（相似度高的）虚假评论，邮件系统检测（收件地址列表高度相似的）垃圾邮件



# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）

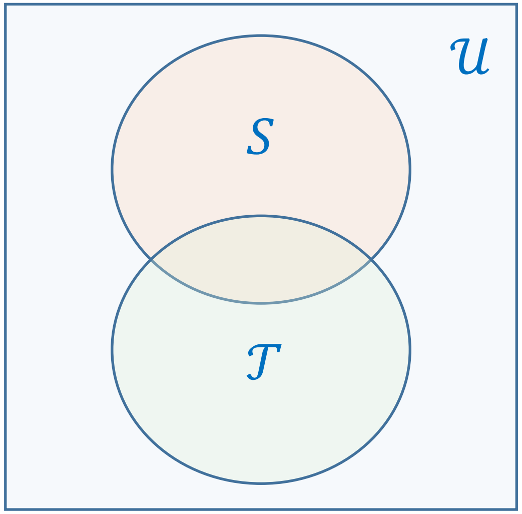
$$J(S,T) = \frac{|S \cap T|}{|S \cup T|}$$

# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）

$$J(S,T)=\frac{|S\cap T|}{|S\cup T|}$$




* 若两个集合都为空集，定义其距离为0

# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|}$$

若用向量的形式计算:

$$J(S,T) = J(v_S, v_T) = \frac{\sum_i \min(v_S(i), v_T(i))}{\sum_i \max(v_S(i), v_T(i))}.$$

* $v_S(i), v_T(i)$表示全集中第$i$个（非重复）元素在集合$S$和$T$中出现的次数

# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）
- 欧几里得距离（Euclidean Distance）/$l_2$距离

若$x, y$为$d$维实数空间的两个点，即$x, y \in \mathbb{R}^d$，它们之间的欧式距离为:

$$D_{euclidean}(x, y) = ||x - y||_2 = \sqrt{\sum_{i=1}^{d}(x(i) - y(i))^2}.$$

图片取自Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox>

# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）
- 欧几里得距离（Euclidean Distance）/$l_2$距离


图片取自新浪微博“爱可可-爱生活”

# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）
- 欧几里得距离（Euclidean Distance）/$l_2$距离
- $l_p$距离

若$x, y$为$d$维实数空间的两个点，即$x, y \in \mathbb{R}^d$，它们之间的$l_p$（$p \geq 1$）距离为:

$$||x-y||_p = \left( \sum_{i=1}^{d} |x(i) - y(i)|^p \right)^{1/p}.$$

图片取自Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox>

# 相似搜索

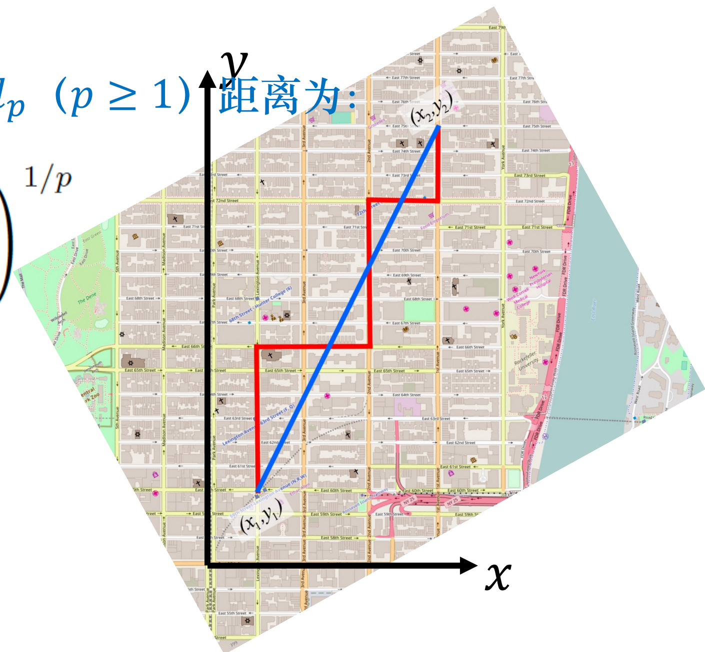
如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）
- 欧几里得距离（Euclidean Distance）/$l_2$距离
- $l_p$距离

若$x,y$为$d$维实数空间的两个点，即$x,y \in \mathbb{R}^d$，它们之间的$l_p$（$p \geq 1$）距离为：

$$||x-y||_p = \left( \sum_{i=1}^{d} |x(i)-y(i)|^p \right)^{1/p}$$

- $p=1$时，称为“曼哈顿距离”



# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）
- 欧几里得距离（Euclidean Distance）/$l_2$距离
- $l_p$距离

若$x, y$为$d$维实数空间的两个点，即$x, y \in \mathbb{R}^d$，它们之间的$l_p$（$p \geq 1$）距离为:

$$||x - y||_p = \left( \sum_{i=1}^{d} |x(i) - y(i)|^p \right)^{1/p}.$$

- $p = 1$时，称为“曼哈顿距离”
- 随着$p$增大，$\|x - y\|_p$越倾向于受令$|x(i) - y(i)|$最大的第$i$个维度的影响
- 定义$\|x - y\|_\infty = \max_i |x(i) - y(i)|$

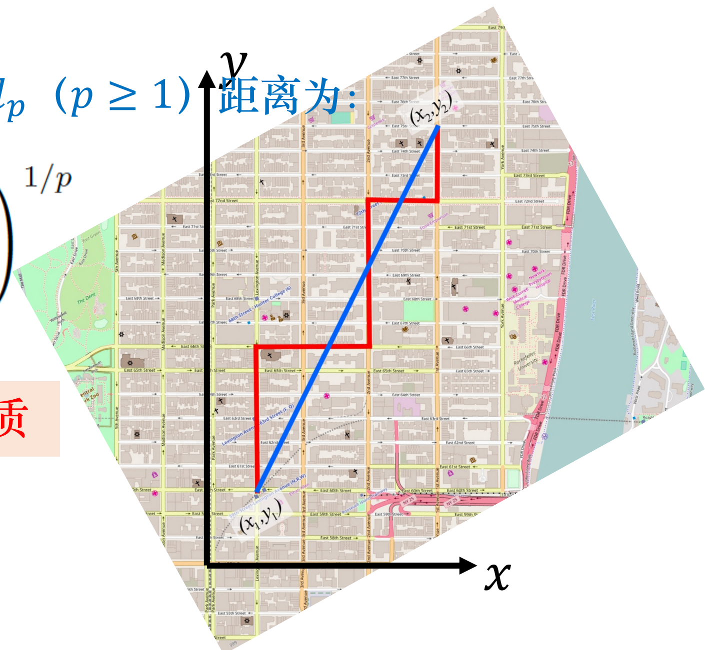
# 相似搜索

如何衡量相似度?

- 杰卡德相似度（Jaccard Similarity）：刻画两个集合之间的距离（一个集合中同样的元素允许出现多次）
- 欧几里得距离（Euclidean Distance）/$l_2$距离
- $l_p$距离

若$x, y$为$d$维实数空间的两个点，即$x, y \in \mathbb{R}^d$，它们之间的$l_p$（$p \geq 1$）距离为：

$$||x-y||_p = \left( \sum_{i=1}^{d} |x(i) - y(i)|^p \right)^{1/p}$$




$l_2$距离不受坐标系旋转的影响，而其余$l_p$距离不具备此性质

# 欧氏距离下的相似搜索

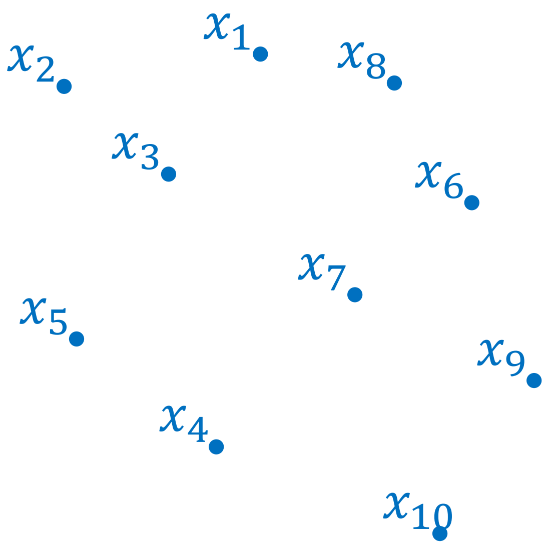
## k维树方法

# k维树

- k维树（k-d tree/k-dimension tree）
- 1975年由Stanford本科生Bentley提出的一种利于相似搜索的数据结构
- 对于低维数据非常有效（维度$d < 20$）
- 核心思想：用二叉搜索树对空间进行分割、存储数据
- 能准确找到数据集中与新数据的欧式距离最小的数据（即不是近似求解）

# k维树

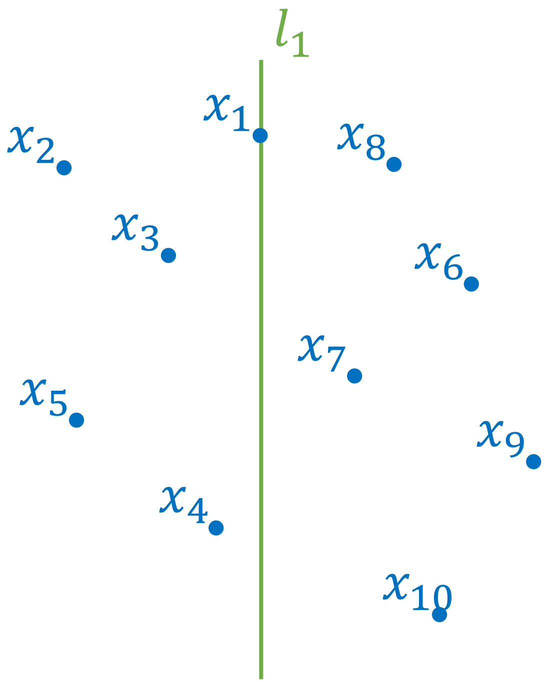
例，为10个2维数据建立k维树




二维平面

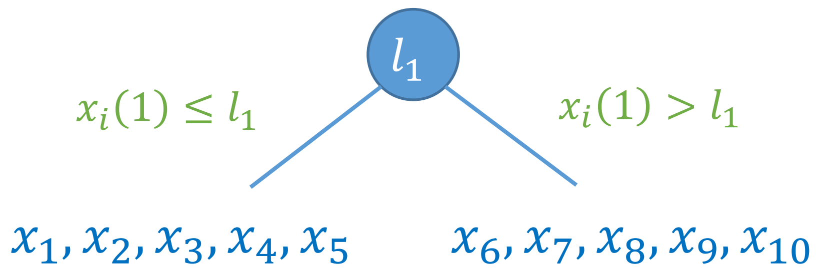
# k维树

例，为10个2维数据建立k维树




二维平面

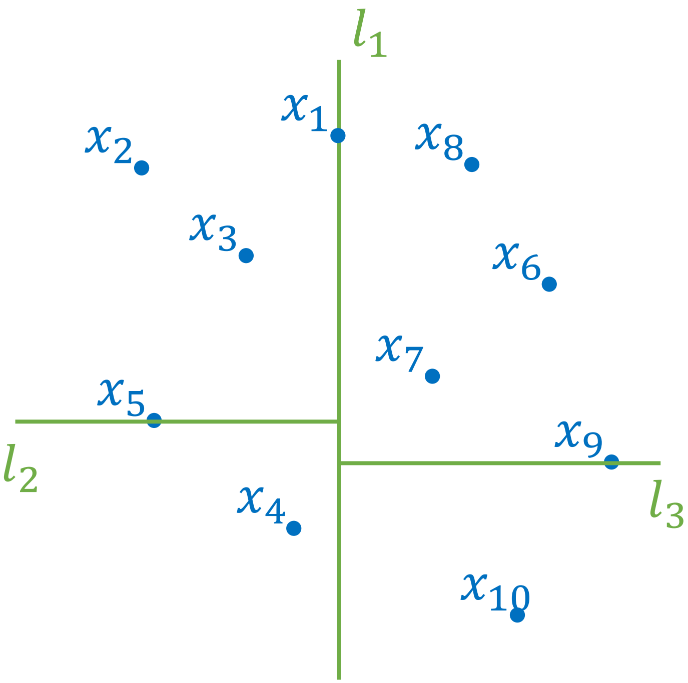
先考虑第1个维度（横轴），找到中值，将数据分成两半（令左半数目小于等于右半数目）




例子修改自Ioannis Emiris <Computational Geometry Search in High dimension and kd-trees>

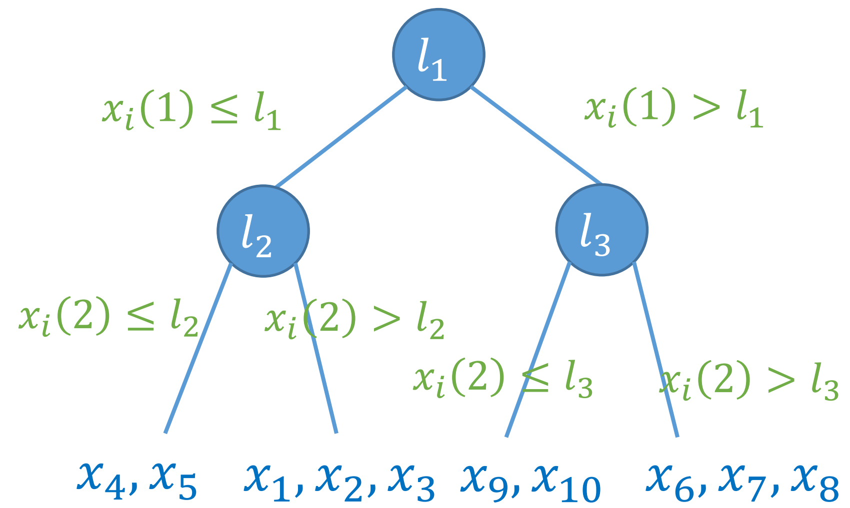
# k维树

例，为10个2维数据建立k维树



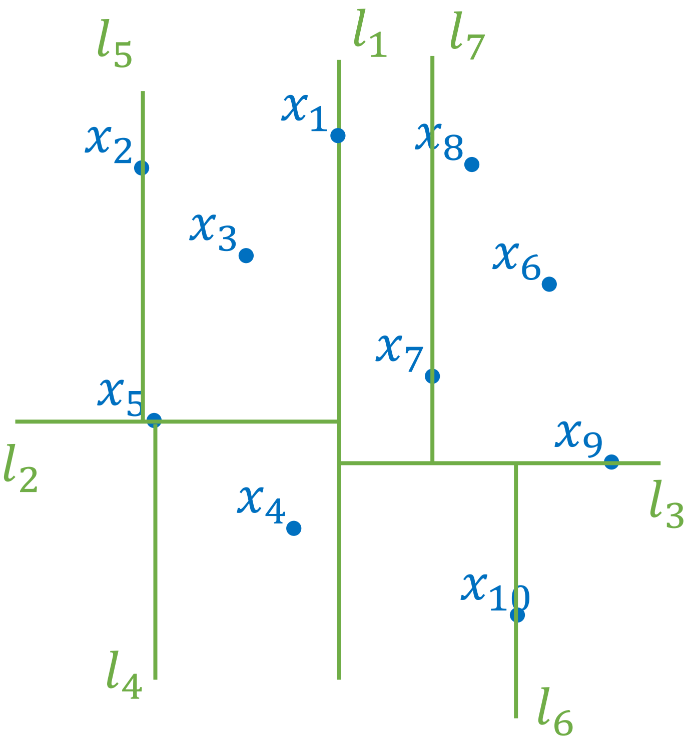

二维平面

再考虑第2个维度（纵轴），找到中值，将每部分数据分别分成两半（令左半数目小于等于右半数目）



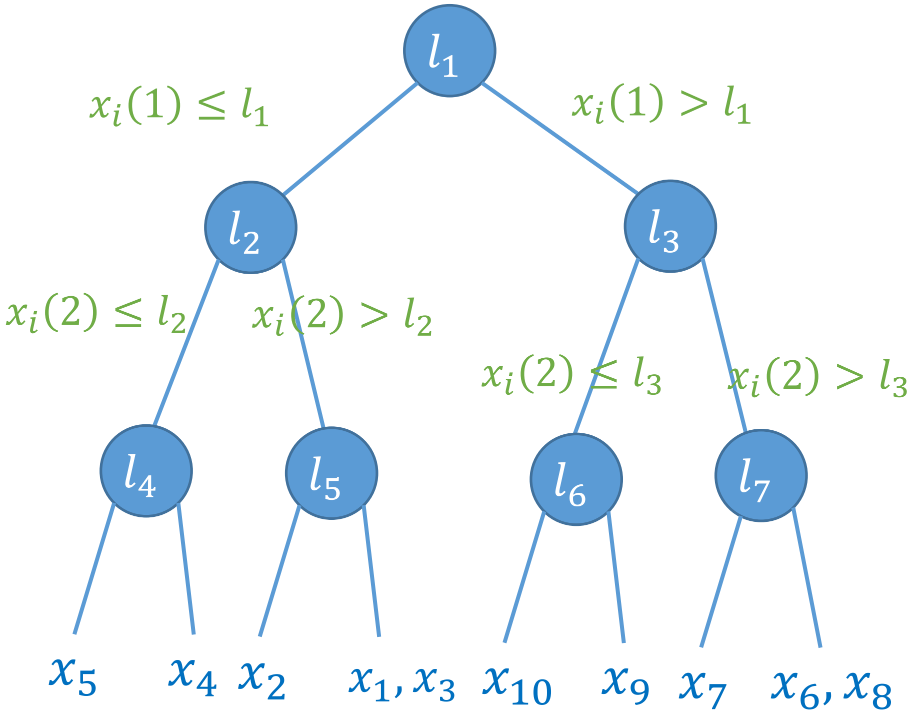
# k维树

例，为10个2维数据建立k维树



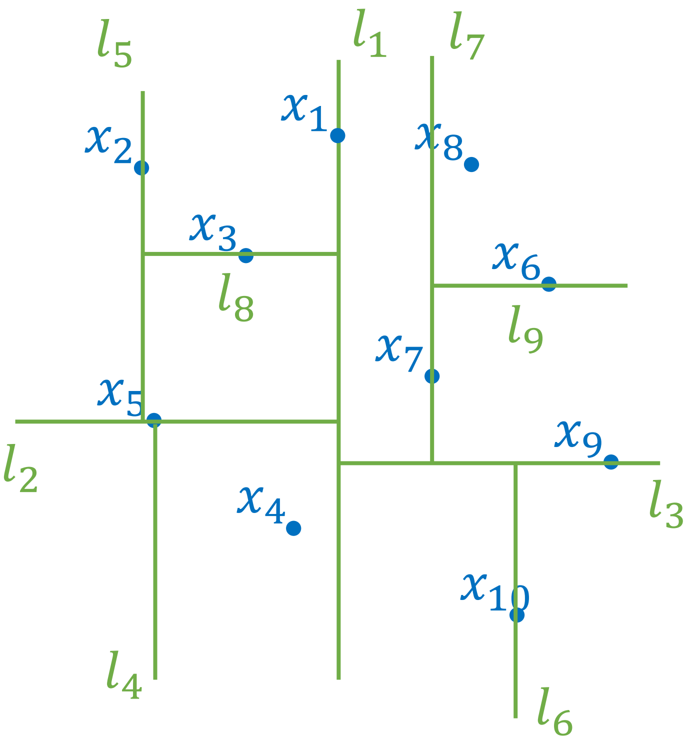

二维平面

再考虑第1个维度（横轴），找到中值，将每部分数据分别分成两半（令左半数目小于等于右半数目）



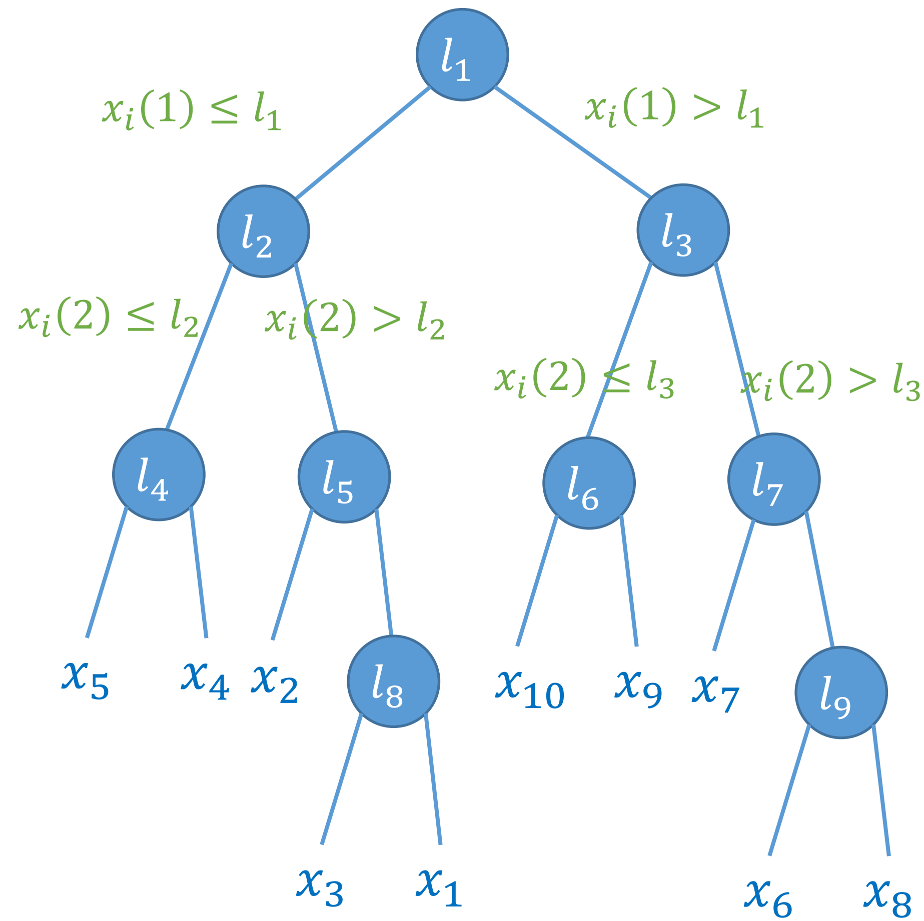
# k维树

例，为10个2维数据建立k维树



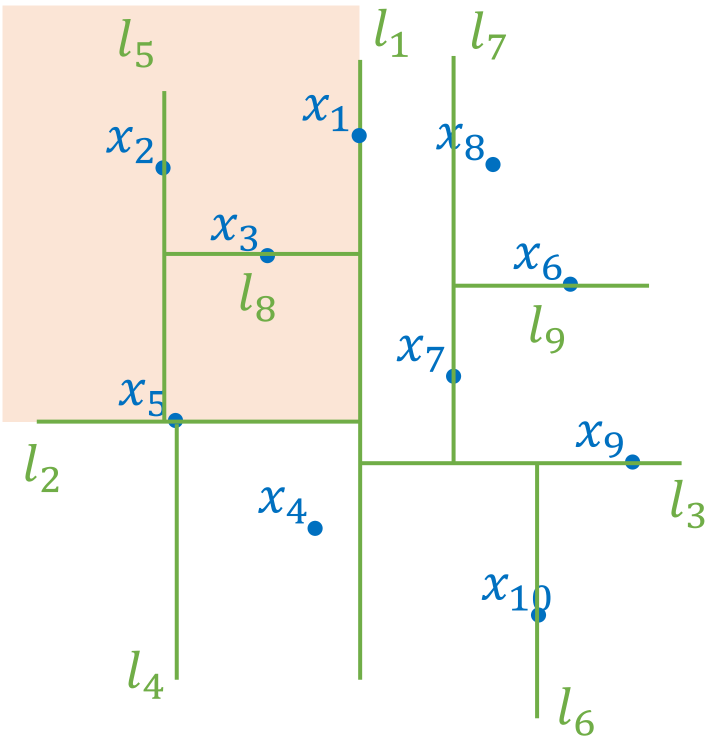

二维平面

再考虑第2个维度（纵轴），找到中值，将每部分数据分别分成两半（令左半数目小于等于右半数目）



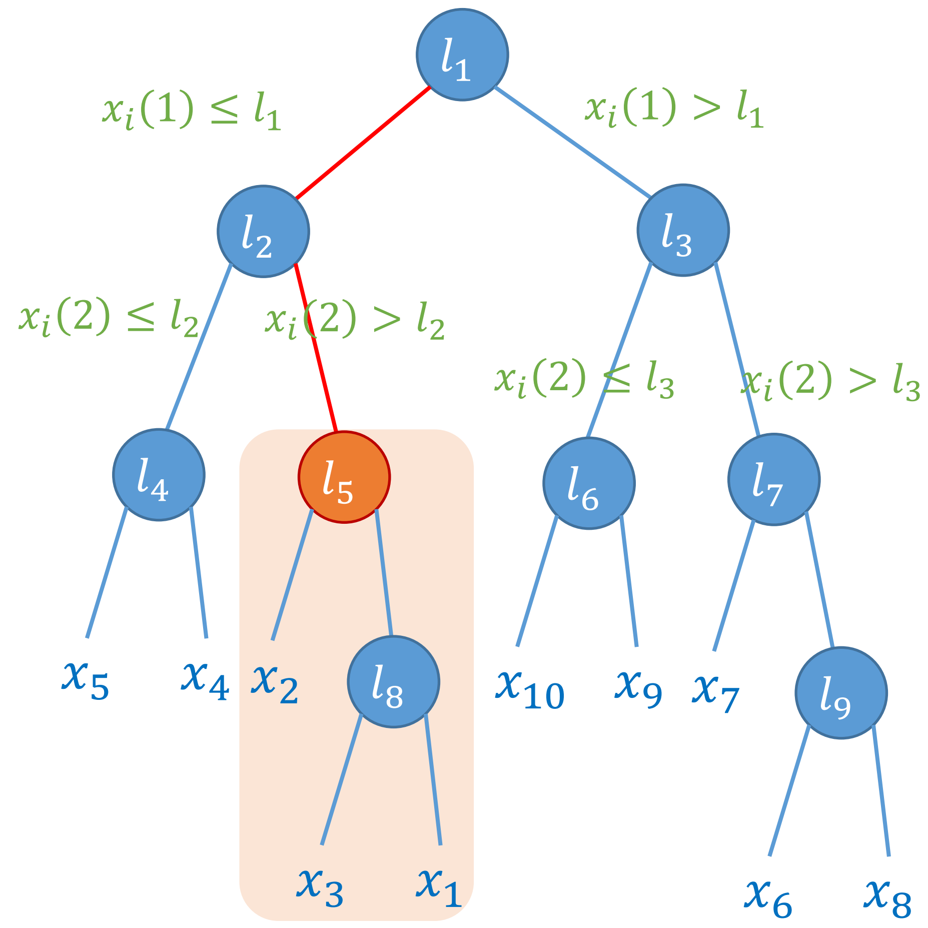
# k维树

例，为10个2维数据建立k维树



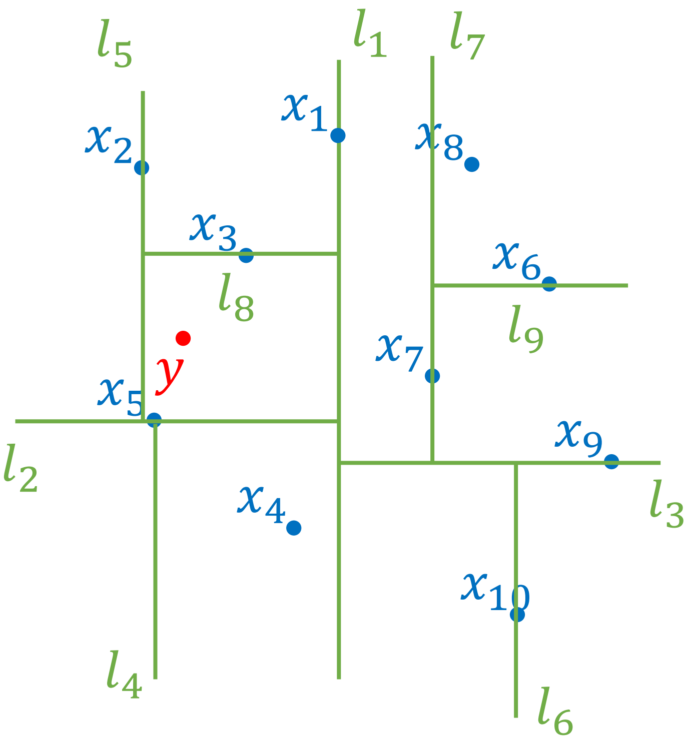

二维平面

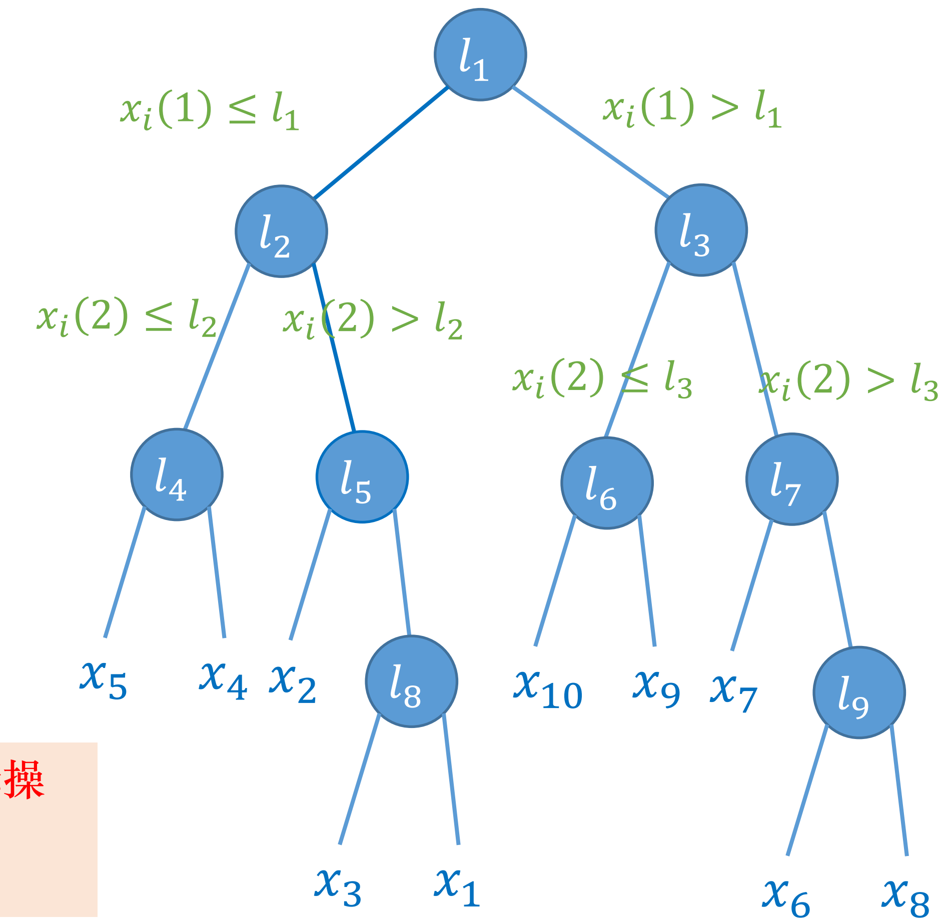
树中每一个分支对应空间中的一个子空间



# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?



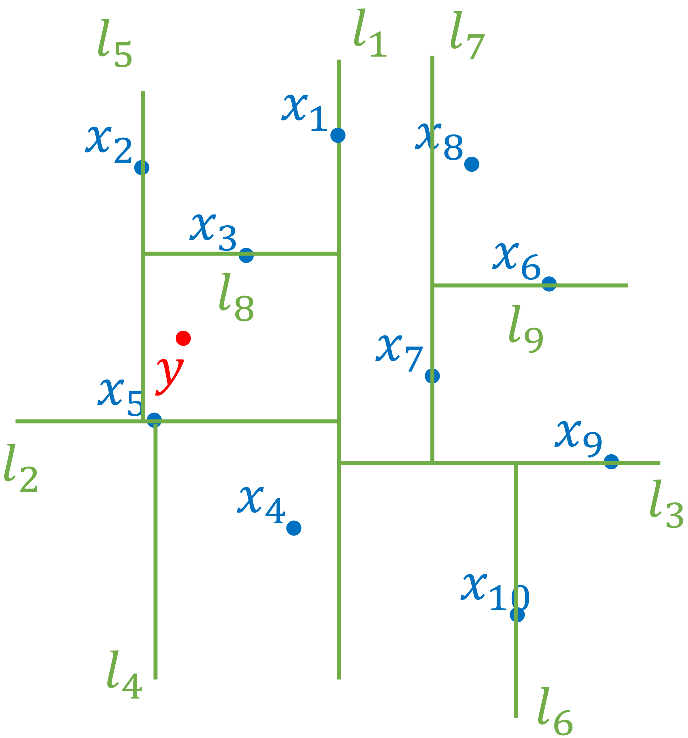



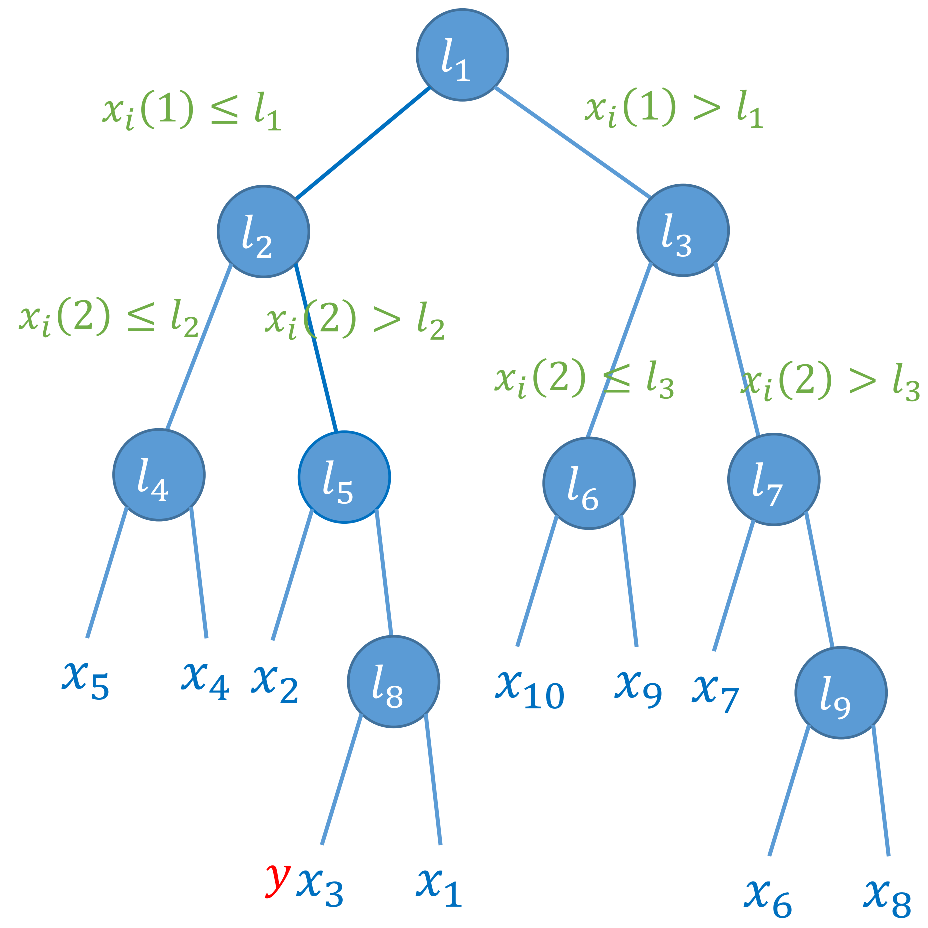

注意：左图只是为方便理解搜索过程，实际操作中只能看到右图中的二叉树

（对高维数据无法画出左图）

# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

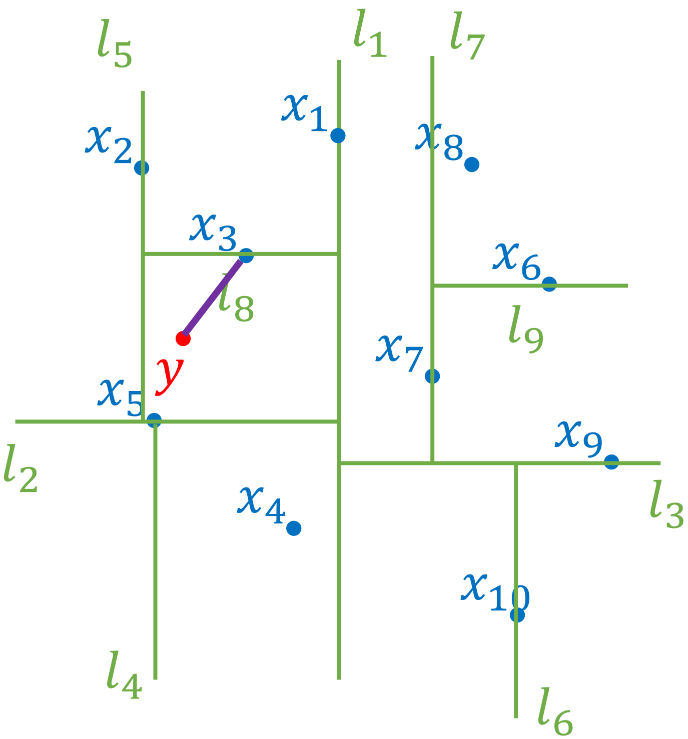


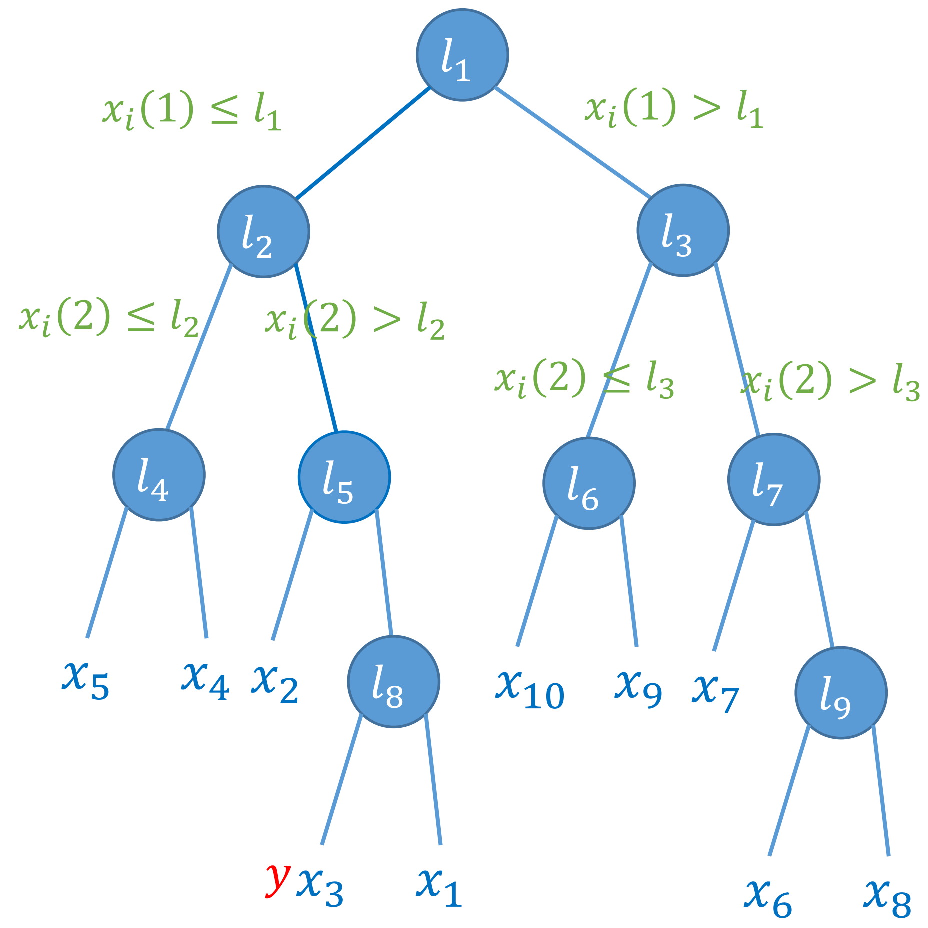




第一步：在二叉树中根据$(y(1), y(2))$找到$y$所属的分支

# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

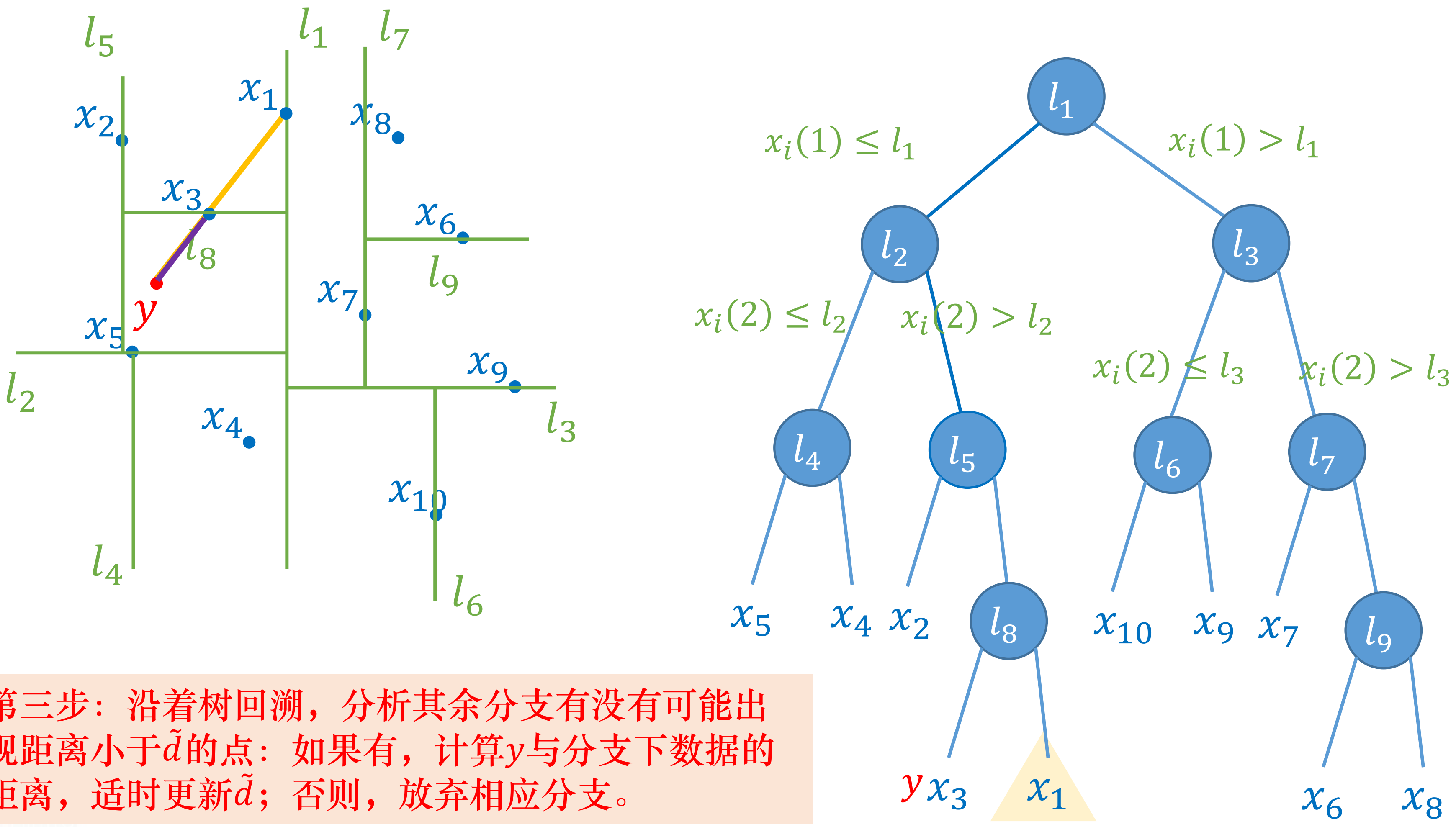



第二步：计算$y$与$y$所属的分支下数据的距离，记录下该点$x_3$及距离$\tilde{d} = D(y, x_3)$

# k维树

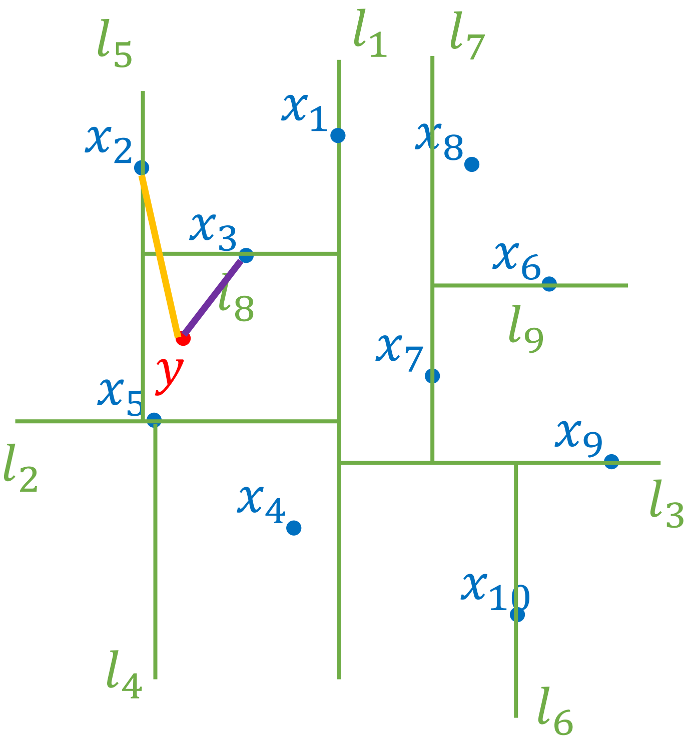
如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

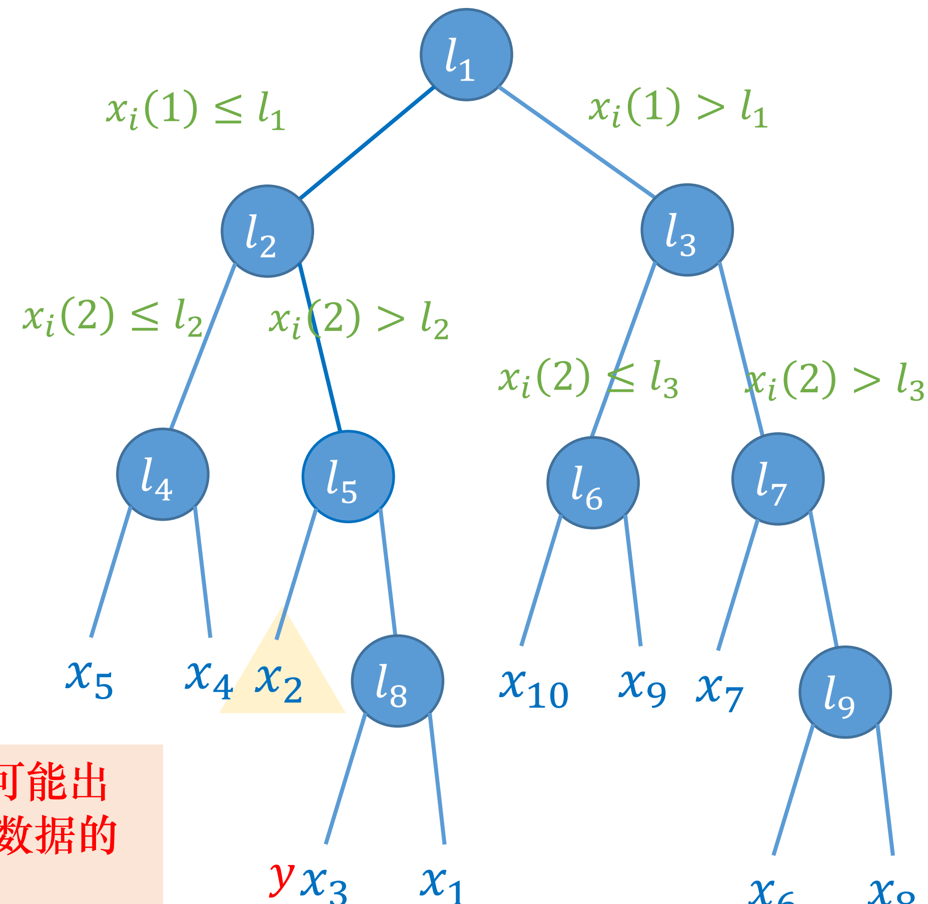



第三步：沿着树回溯，分析其余分支有没有可能出现距离小于$\tilde{d}$的点：如果有，计算$y$与分支下数据的距离，适时更新$\tilde{d}$；否则，放弃相应分支。

# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

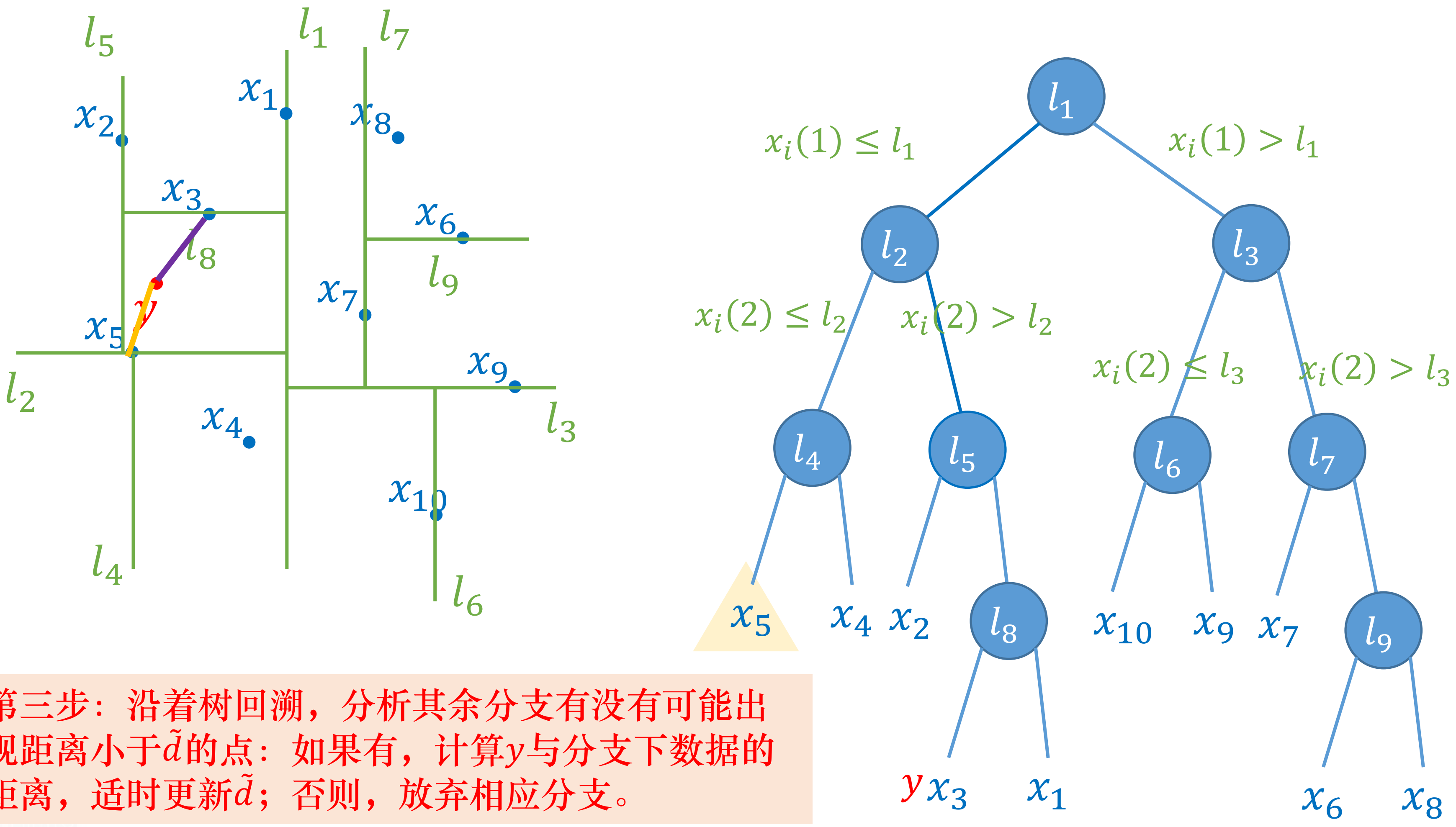



第三步：沿着树回溯，分析其余分支有没有可能出现距离小于$\tilde{d}$的点：如果有，计算$y$与分支下数据的距离，适时更新$\tilde{d}$；否则，放弃相应分支。

# k维树

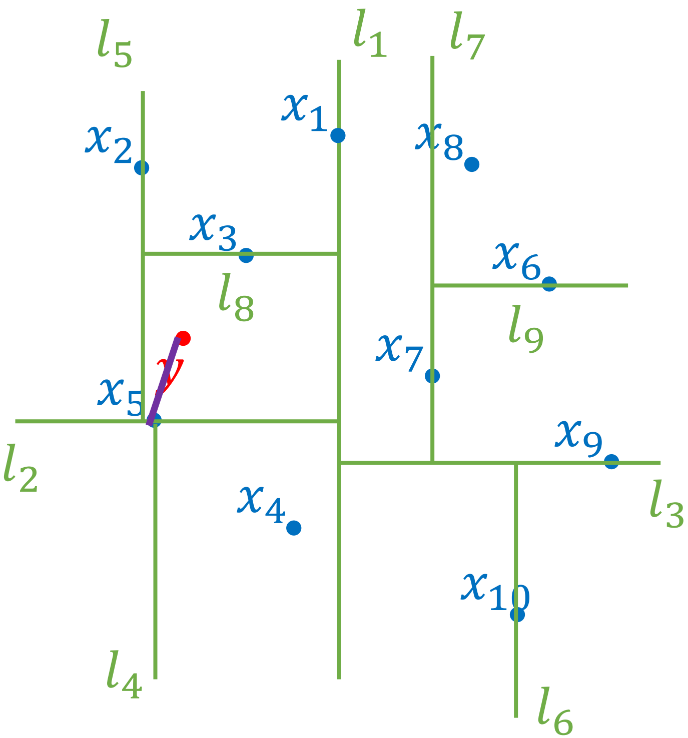
如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

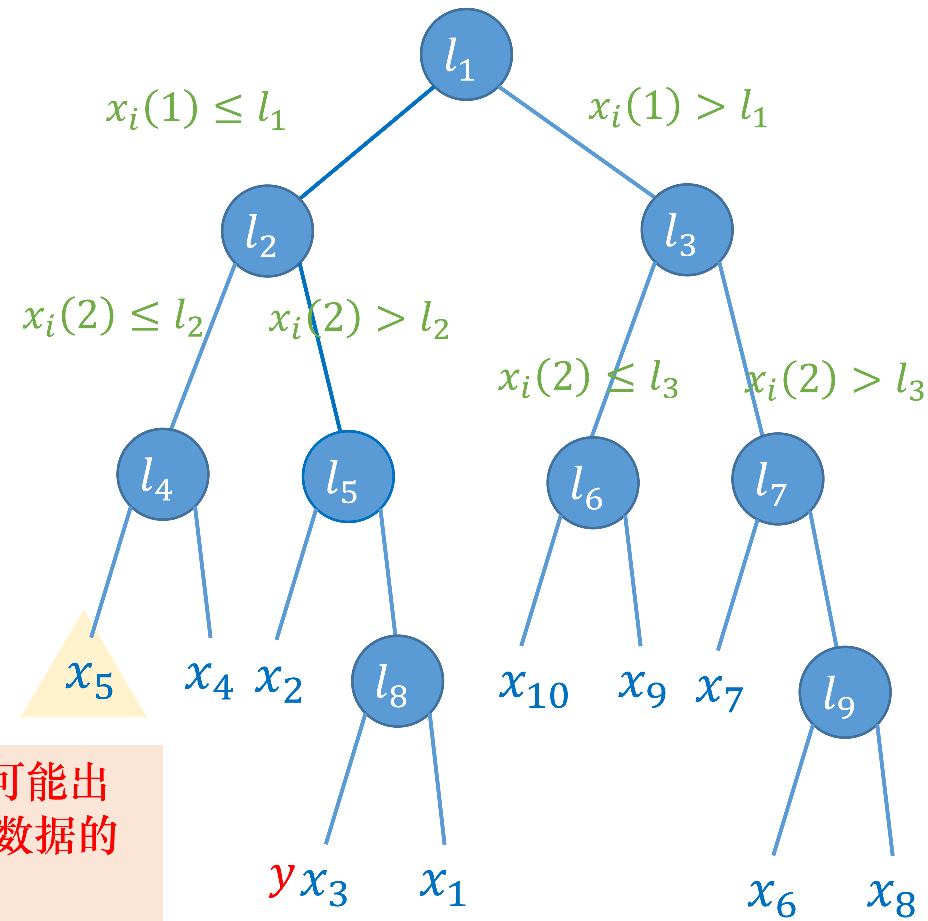



第三步：沿着树回溯，分析其余分支有没有可能出现距离小于$\tilde{d}$的点：如果有，计算$y$与分支下数据的距离，适时更新$\tilde{d}$；否则，放弃相应分支。

# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

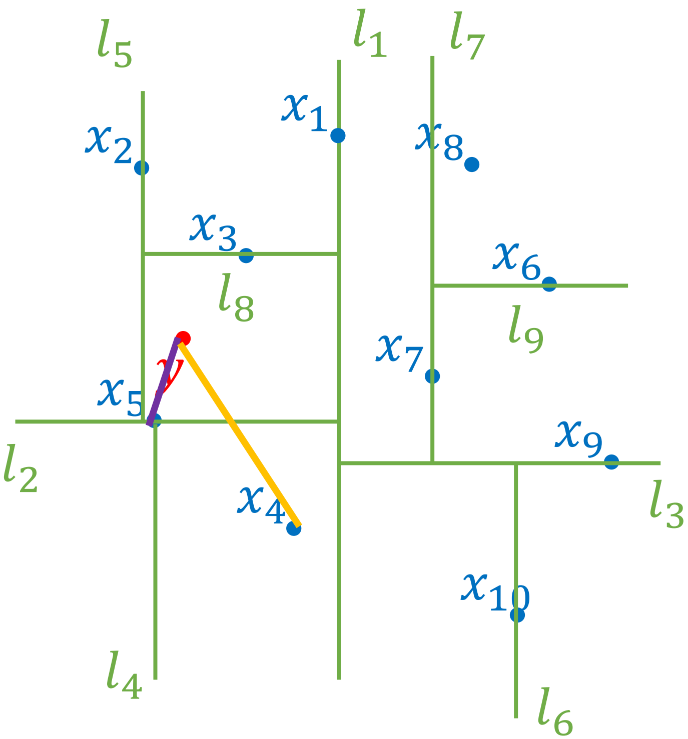


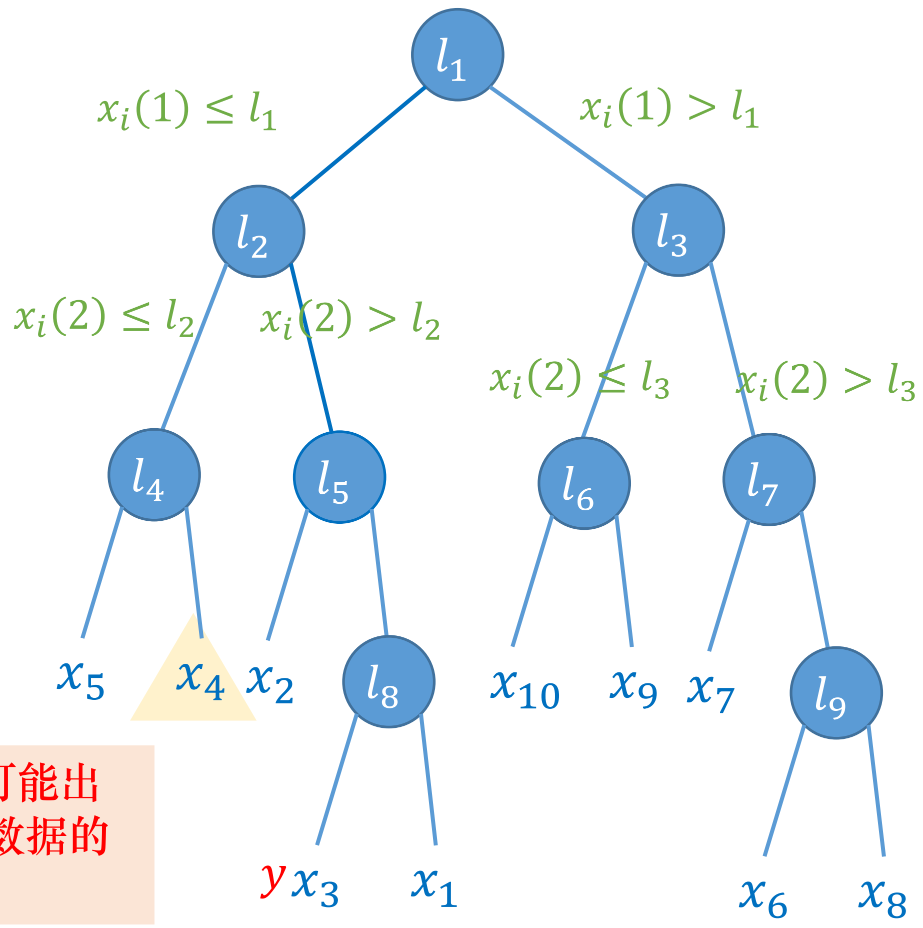




第三步：沿着树回溯，分析其余分支有没有可能出现距离小于$\tilde{d}$的点：如果有，计算$y$与分支下数据的距离，适时更新$\tilde{d}$；否则，放弃相应分支。

# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?



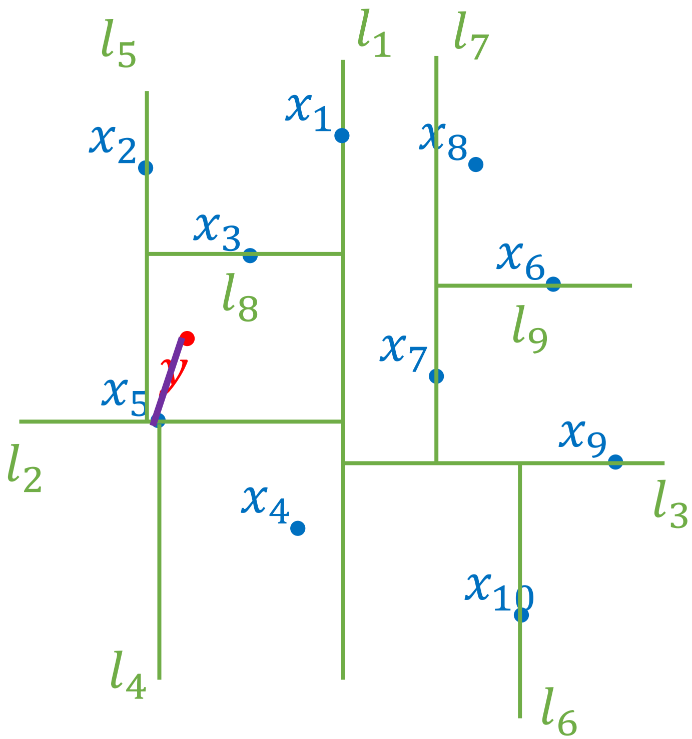



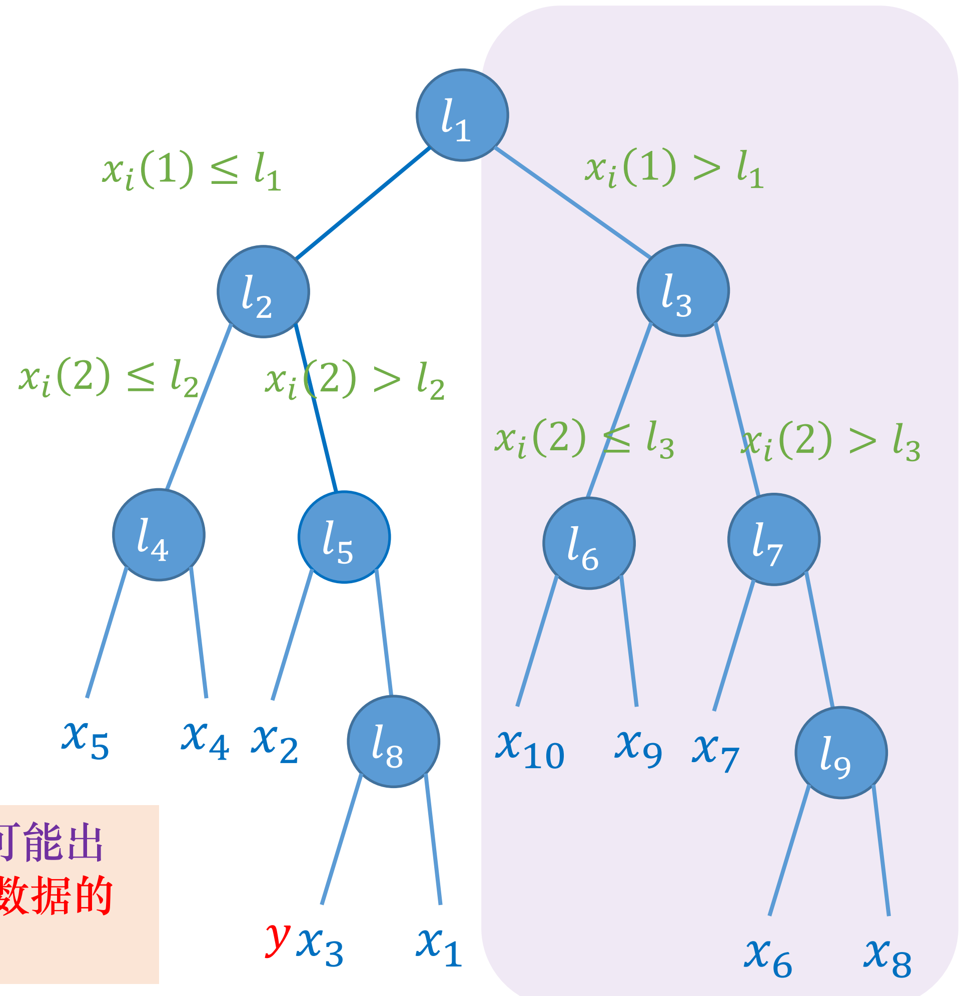

第三步：沿着树回溯，分析其余分支有没有可能出现距离小于$\tilde{d}$的点：如果有，计算$y$与分支下数据的距离，适时更新$\tilde{d}$；否则，放弃相应分支。

# k维树

当$x_i(1) > l_1$时，可得$D(y, x_i) > \tilde{d} = D(y, x_5)$
所以可以放弃搜索$x_i(1) > l_1$对应的分支

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

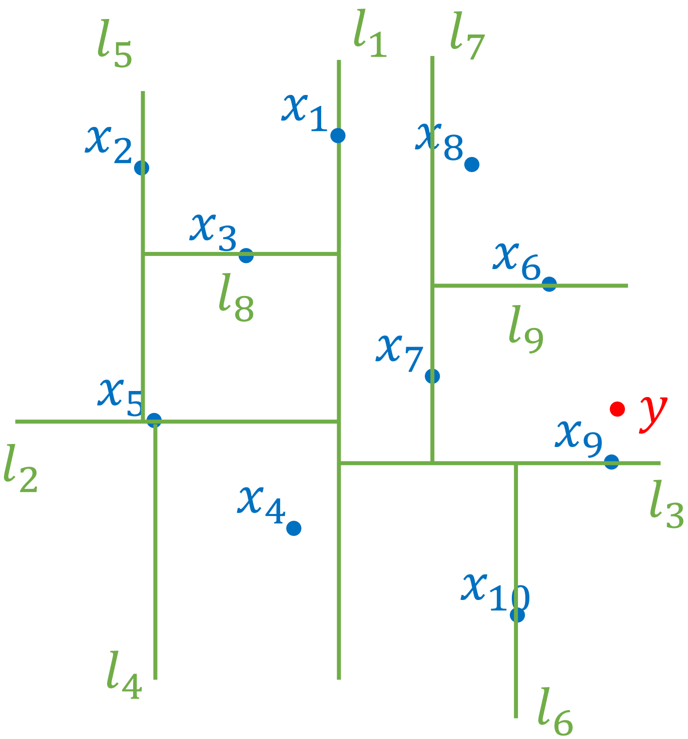


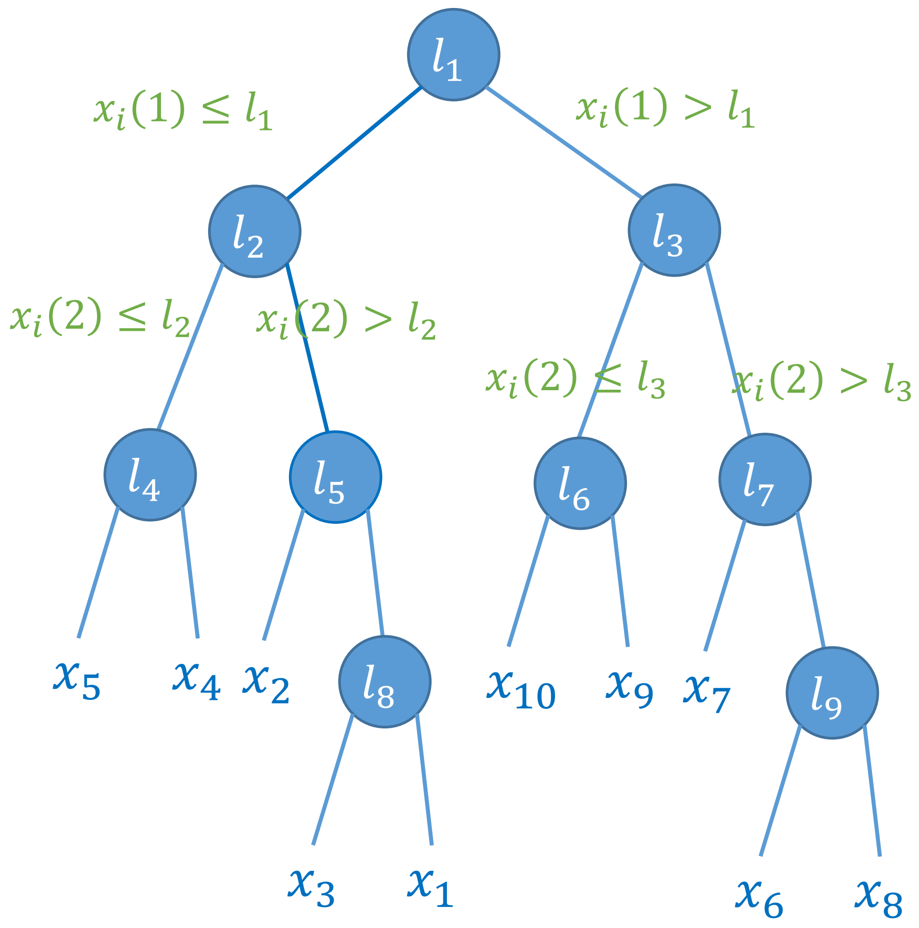




第三步：沿着树回溯，分析其余分支有没有可能出现距离小于$\tilde{d}$的点：如果有，计算$y$与分支下数据的距离，适时更新$\tilde{d}$；否则，放弃相应分支。

# k维树

如何用k维树查找数据集中与给定新数据$y$的欧式距离最新的数据?

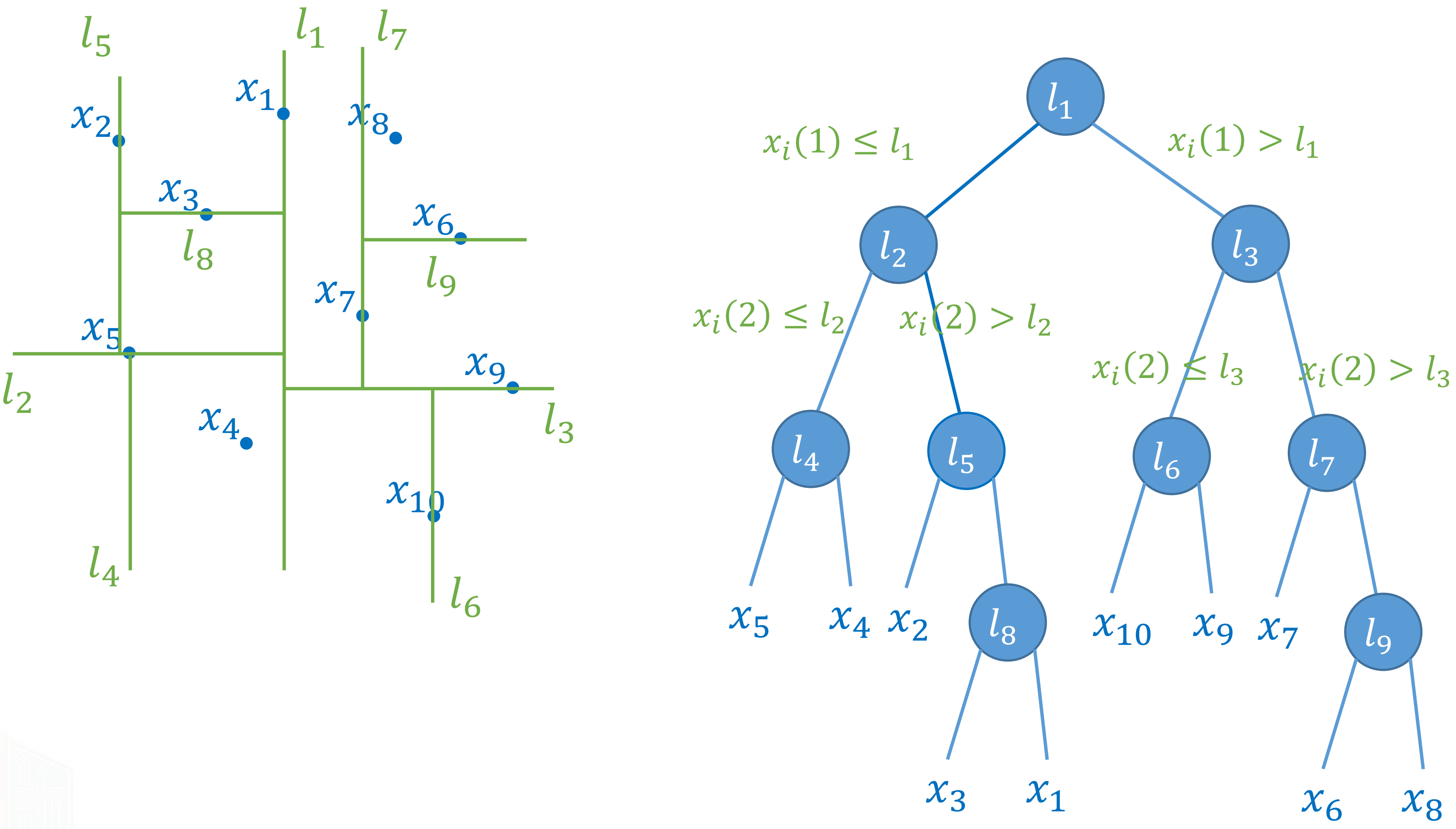



按照前述搜索过程，当$y$处于图示位置时，需要和哪些数据进行对比?

# k维树

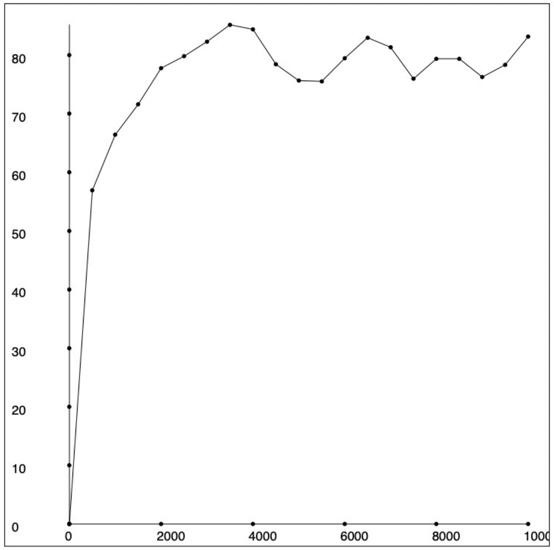
此例仅展示维数为2的情况。当数据维数小于等于20时，非常有效



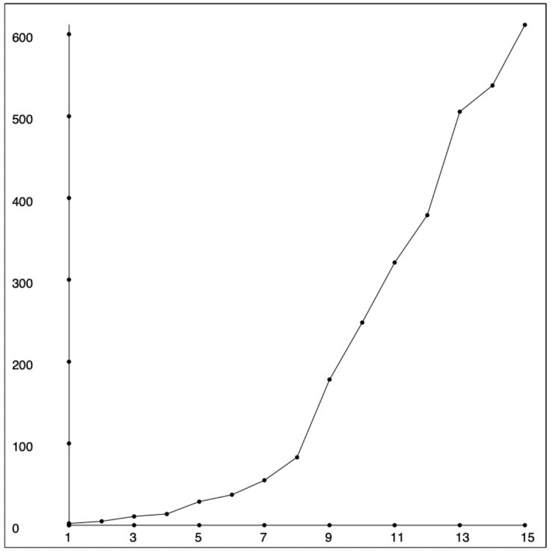
# k维树

维数灾难（Curse of dimensionality）： 复杂度随数据维度$d$指数增长

检索次数




数据集大小$N$




数据维度$d$

为什么？ 是由于k维树的缺陷还是由于难以对高维数据进行相似搜索？

实验结果取自Sham Kakade < CSE547/STAT548 - Machine Learning for Big Data>

# k维树

高维空间缺少像低维空间（比如二维/三维）的几何特性，比如在高维空间有大量的数据两两之间距离类似

**Example 4.1** What is the largest number of points that fit in $d$-dimensional space, with the property that all pairwise distances are in the interval $[0.75, 1]$?

例子取自Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox>

# k维树

高维空间缺少像低维空间（比如二维/三维）的几何特性，比如在高维空间有大量的数据两两之间距离类似

**Example 4.1** What is the largest number of points that fit in $d$-dimensional space, with the property that all pairwise distances are in the interval $[0.75, 1]$?

- $d = 1$: At most 2 points have this property...if you try to fit a third point, at least one of the 3 pairwise distances will be off.
- $d = 2$: At most 3 points have this property...if you try to fit a fourth point, at least one of the 6 pairwise distances will be off.
- $d = 100$: You will be able to fit several thousand points!
- In general, you will be able to fit an exponential number of points (a quick calculation shows that a random set of $exp(\sqrt{d})$ will satisfy this property with high probability).

$d = 100$时，若选其中任何一个数据作为参考数据、寻找相似数据，可能有大量距离相似的数据

例子取自Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox>

# 欧氏距离下的相似搜索

## 数据降维

# 高维数据

大数据不仅说明数据量大，还意味数据的**维度高**：

- 微博有上亿日活用户，每个用户的记录可以有上千个维度：关注/被关注/浏览记录/点赞时间/发微博量/微博内容/...
- 一个时长3分钟、500x500像素、每秒15帧、3颜色通道的视频，有至少二十亿个像素值

# 高维数据

大数据不仅说明数据量大，还意味数据的**维度高**：

- 微博有上亿日活用户，每个用户的记录可以有上千个维度：关注/被关注/浏览记录/点赞时间/发微博量/微博内容/...
- 一个时长3分钟、500x500像素、每秒15帧、3颜色通道的视频，有至少二十亿个像素值

对高维数据，通过k维树等方法难以有效解决相似搜索的问题

数据降维（Dimensionality Reduction）：把**高维空间**中的数据用**低维空间**中的数据表示，**并尽可能保留原始数据之间的特定性质**

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n \in \mathbb{R}^k \Rightarrow \widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, \dots, \widetilde{\boldsymbol{x}}_n \in \mathbb{R}^d \text{（} d \ll k \text{）}$$

* 对于相似搜索，需要保留的是不同数据之间的相对距离

# 高维数据

**欧几里得低失真嵌入（Euclidean Low Distortion Embedding）**

给定$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^k$及误差$\varepsilon \geq 0$，求满足以下条件的$\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_n \in \mathbb{R}^d$（$d \ll k$）：

$$(1-\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \leq \left\|\widetilde{\boldsymbol{x}}_i - \widetilde{\boldsymbol{x}}_j\right\|_2 \leq (1+\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2, \forall i, j = 1, \ldots, n.$$

**相似搜索：** 先对高维数据$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$降维、得到$\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_n$，再用k维树等方法做相似搜索

# 欧氏距离下的相似搜索

## JL转换

# JL转换

JL转换（Johnson - Lindenstrauss Transform）：

- 1984年由William Buhmann Johnson与Joram Lindenstrauss提出
- 可以用于解决欧几里得低失真嵌入问题

[PDF] Extensions of Lipschitz mappings into a Hilbert space 26

WB Johnson, J Lindenstrauss - Contemporary mathematics, 1984 - a.pomf.se

In this note we consider the following extension problem for Lipschitz functions: Given a metric space X and n= 2, 3, 4,...'estimate the smallest constant L= L (X, n) so that every~ pping f from every n-element subset of X into t 2 extends to a mapping f from X into t 2 with (Here ll&lltip is the Lipschitz constant of the function g.) A classical result of Kirszbraun's [14, p. 48] states that L (t2, n)= 1 for all n, but it is easy to see that L (X, n)~~ as n~~ for many metric spaces X. Marcus and Pisier [10] initiated the study of L (X, n) for X= Lp.(For brevity …

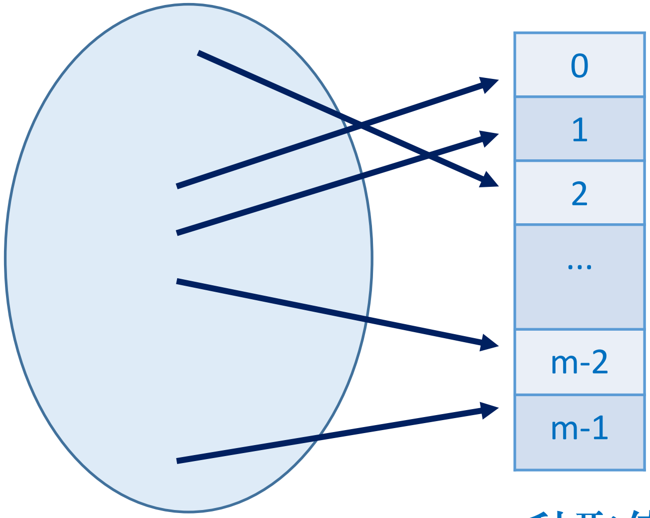
☆ Save 99 Cite Cited by 3481 Related articles All 3 versions

泛函分析方向的研究

# JL转换

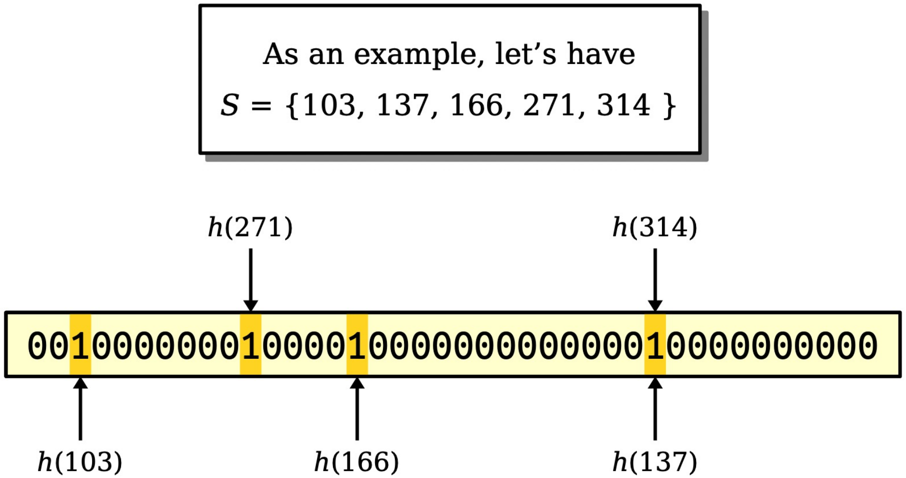
前四讲中多次利用随机哈希函数将取值范围大的输入元素映射为取值范围较小的结果

例如解决从属问题的布隆过滤器：




比如URL的集合

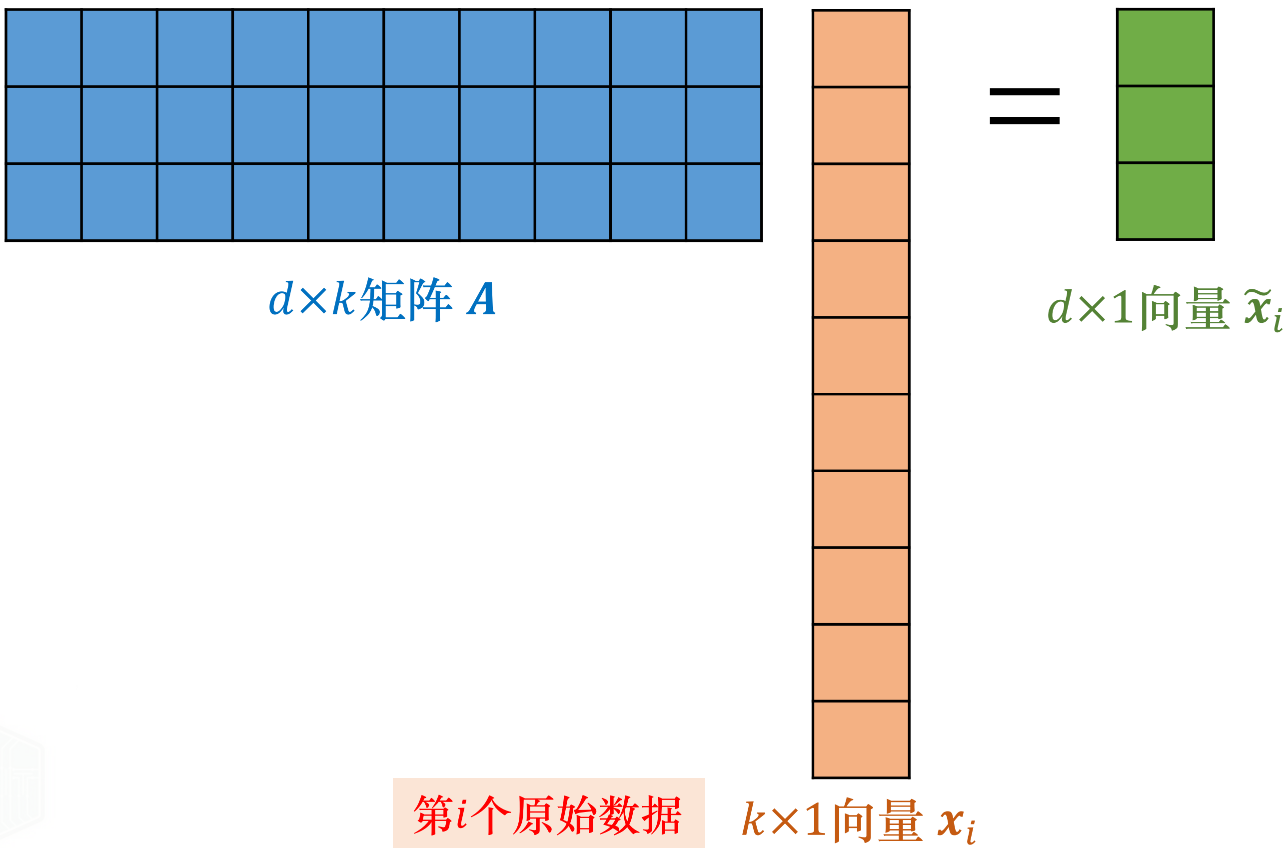
m种取值



# JL转换

前四讲中多次利用随机哈希函数将取值范围大的输入元素映射为取值范围较小的结果

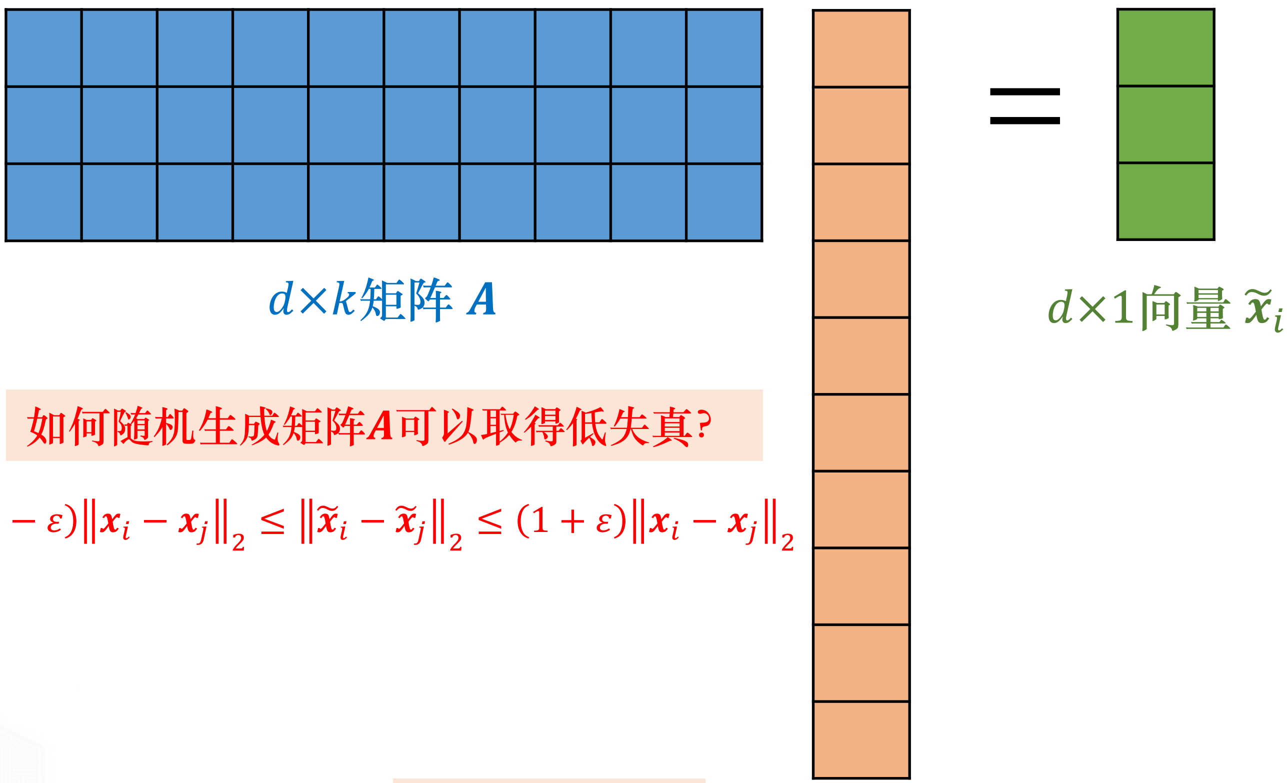
JL转换的思想：利用随机生成的$d\times k$大小的矩阵，将$k$维数据压缩至$d$维数据



# JL转换

前四讲中多次利用随机哈希函数将取值范围大的输入元素映射为取值范围较小的结果

JL转换的思想：利用随机生成的$d\times k$大小的矩阵，将$k$维数据压缩至$d$维数据




如何随机生成矩阵$\boldsymbol{A}$可以取得低失真?

$$(1-\varepsilon)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2 \le \|\tilde{\boldsymbol{x}}_i-\tilde{\boldsymbol{x}}_j\|_2 \le (1+\varepsilon)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$$

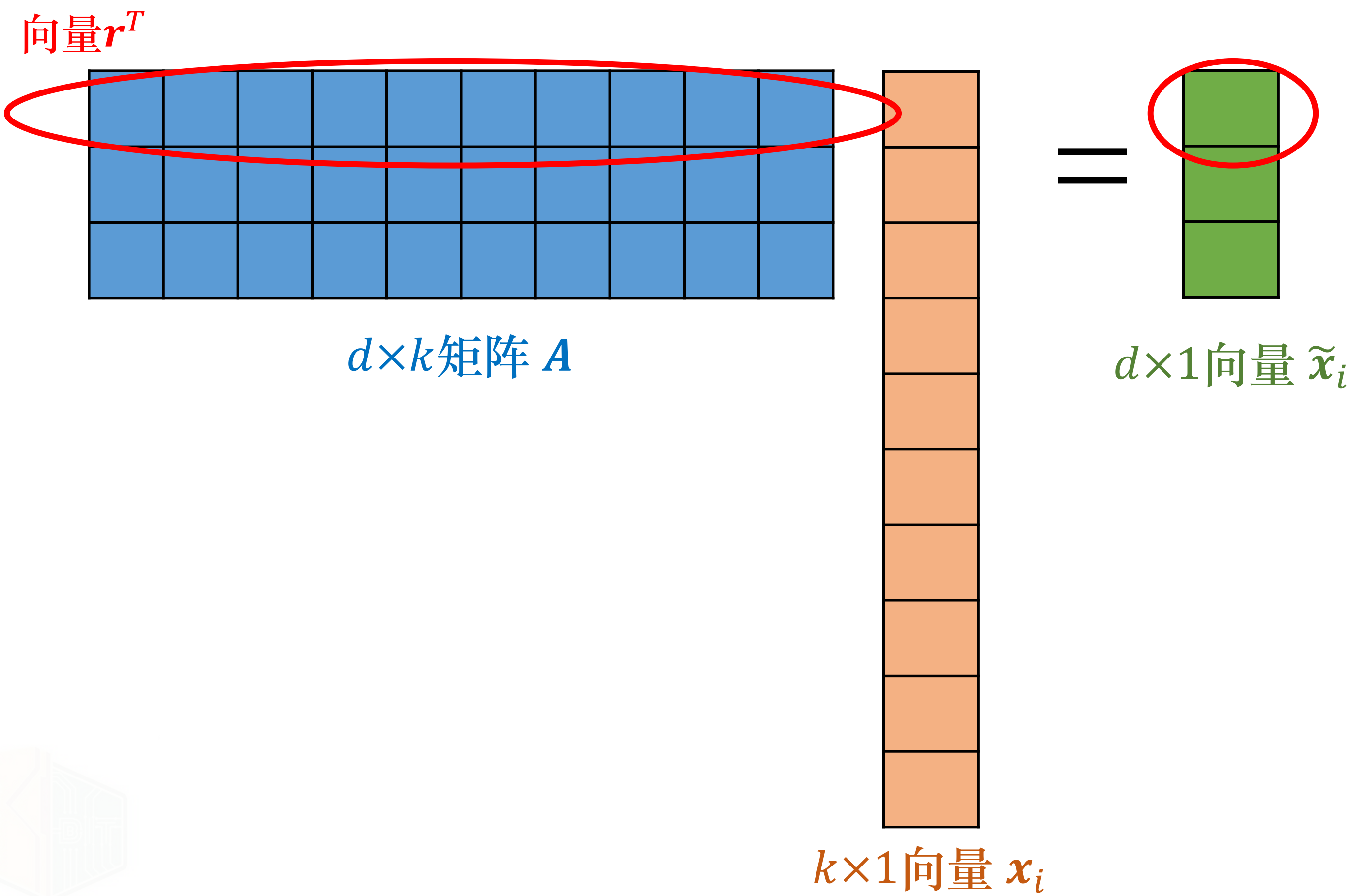
第$i$个原始数据

# JL转换

矩阵$\boldsymbol{A}$的每一个$1\times k$的行向量 与 $k\times 1$的列向量相乘

可以定义为用由向量$\boldsymbol{r}$定义的函数$f(\cdot)$对数据$\boldsymbol{x}$做变换

$$f_{\mathbf{r}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{r} \rangle = \sum_{j=1}^{k} x_j r_j.$$

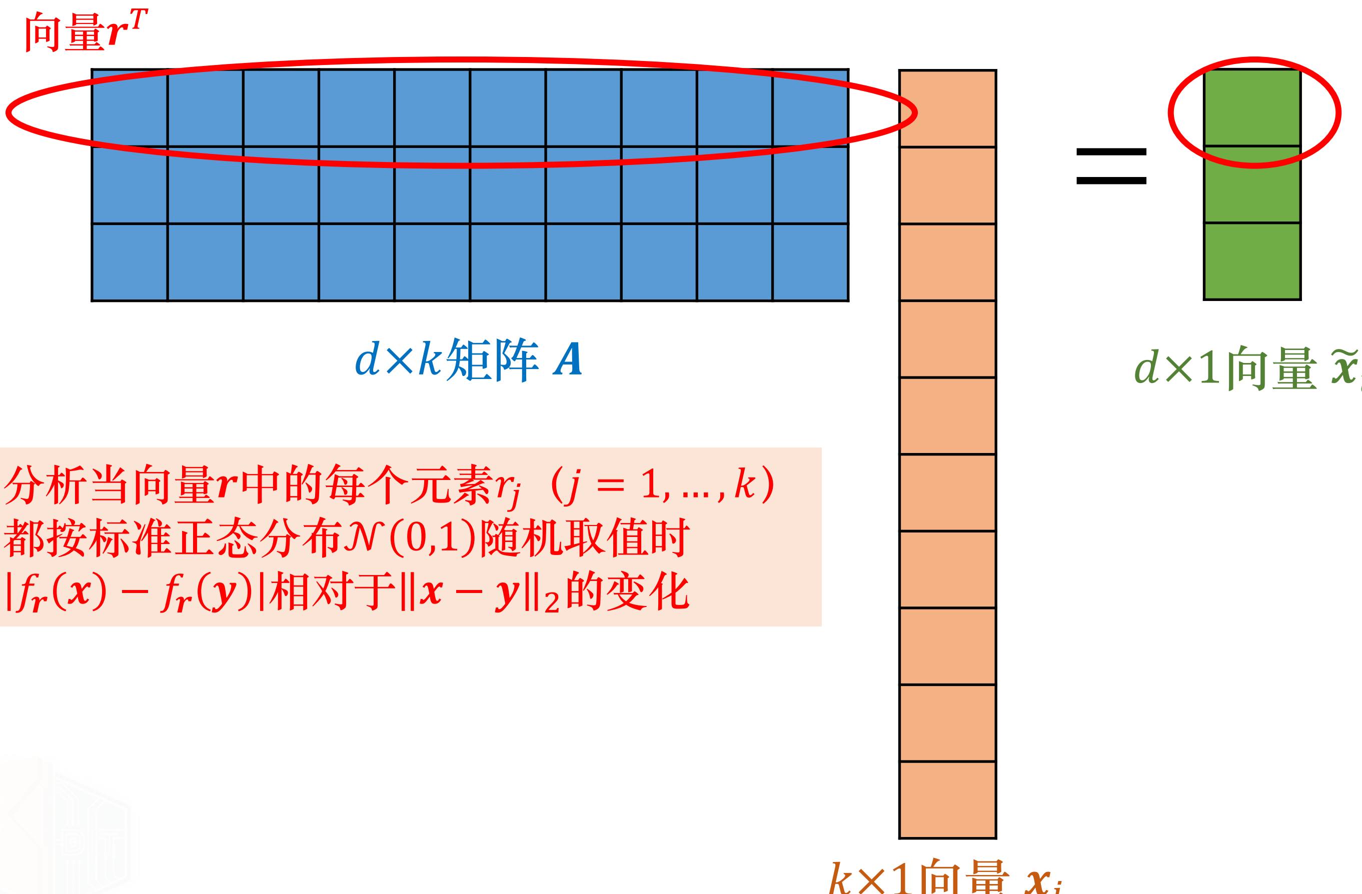


# JL转换

矩阵$\boldsymbol{A}$的每一个$1\times k$的行向量 与 $k\times 1$的列向量相乘

可以定义为用由向量$\boldsymbol{r}$定义的函数$f(\cdot)$对数据$\boldsymbol{x}$做变换

$$f_{\mathbf{r}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{r} \rangle = \sum_{j=1}^{k} x_j r_j .$$




分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j = 1, \dots, k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$|f_r(\boldsymbol{x}) - f_r(\boldsymbol{y})|$相对于$\|\boldsymbol{x} - \boldsymbol{y}\|_2$的变化

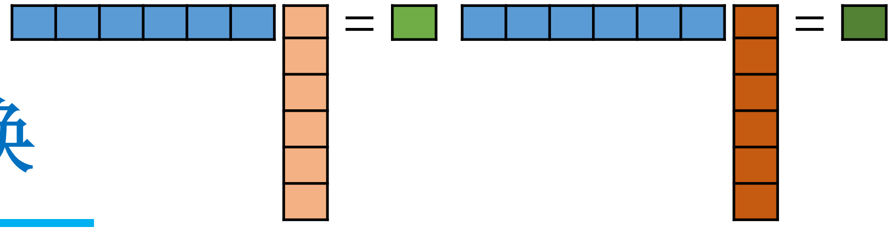
# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\ldots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_jr_j.$$

$$f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_jr_j-\sum_{j=1}^{k}y_jr_j=\sum_{j=1}^{k}\underbrace{(x_j-y_j)}_{\text{看作常数}}r_j.$$

每个$r_j$都是服从正态分布的随机变量



# JL转换

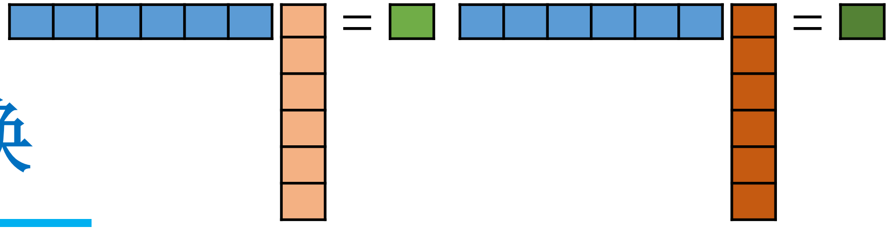
分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j = 1, \ldots, k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x} - \boldsymbol{y}$的变化

$$f_{\mathbf{r}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{r} \rangle = \sum_{j=1}^{k} x_j r_j .$$

看作常数

$$f_{\mathbf{r}}(\mathbf{x}) - f_{\mathbf{r}}(\mathbf{y}) = \sum_{j=1}^{k} x_j r_j - \sum_{j=1}^{k} y_j r_j = \sum_{j=1}^{k} (x_j - y_j) r_j .$$

每个$r_j$都是服从正态分布的随机变量



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\dots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_r(\boldsymbol{x})-f_r(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

看作常数

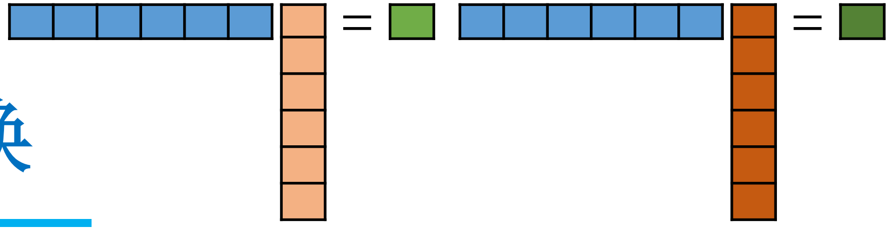
$$\boxed{f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_j r_j-\sum_{j=1}^{k}y_j r_j=\sum_{j=1}^{k}\underline{(x_j-y_j)}r_j.}$$

每个$r_j$都是服从正态分布的随机变量

$k$个服从正态分布的随机变量的加权和依然是正态分布

正态分布的优异性质

随机变量$\sum_{j=1}^{k}(x_j-y_j)r_j$服从正态分布，期望为___，方差为______？



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\ldots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

$$\boxed{f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_j r_j-\sum_{j=1}^{k}y_j r_j=\sum_{j=1}^{k}\overset{\text{看作常数}}{\underline{(x_j-y_j)}}r_j.}$$

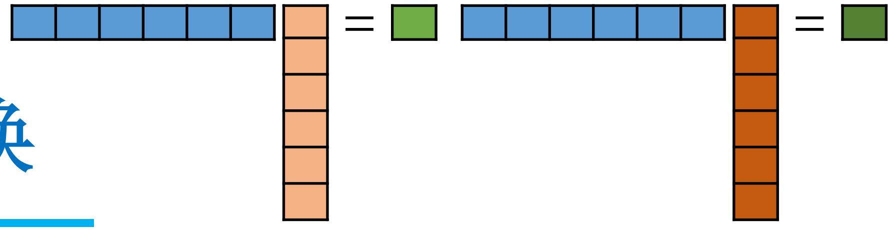
每个$r_j$都是服从正态分布的随机变量

$k$个服从正态分布的随机变量的加权和依然是正态分布　　正态分布的优异性质

随机变量 $\sum_{j=1}^{k}(x_j-y_j)r_j$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j-y_j)^2$

If $X_1, X_2, \ldots, X_n$ are independent, then

$$Var[c_0+c_1X_1+c_2X_2+\ldots+c_nX_n]=c_1^2\,Var[X_1]+c_2^2\,Var[X_2]+\ldots+c_n^2\,Var[X_n]$$



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\dots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle \mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

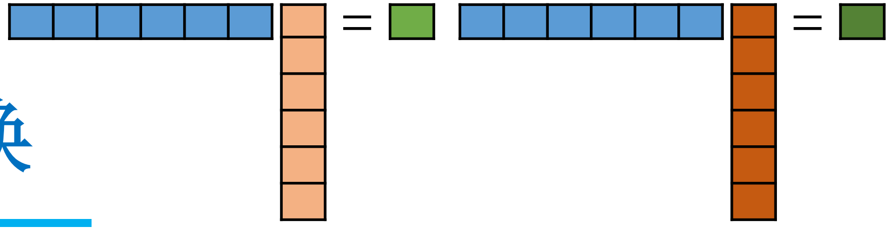
看作常数

$$\boxed{f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_j r_j-\sum_{j=1}^{k}y_j r_j=\sum_{j=1}^{k}\underline{(x_j-y_j)}r_j.}$$

每个$r_j$都是服从正态分布的随机变量

随机变量 $f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j-y_j)^2$ 即 $\|\mathbf{x}-\mathbf{y}\|_2^2$.

$$\mathbf{Var}[f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})]=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\ldots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

看作常数

$$\boxed{f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_j r_j-\sum_{j=1}^{k}y_j r_j=\sum_{j=1}^{k}\underline{(x_j-y_j)}r_j.}$$

每个$r_j$都是服从正态分布的随机变量

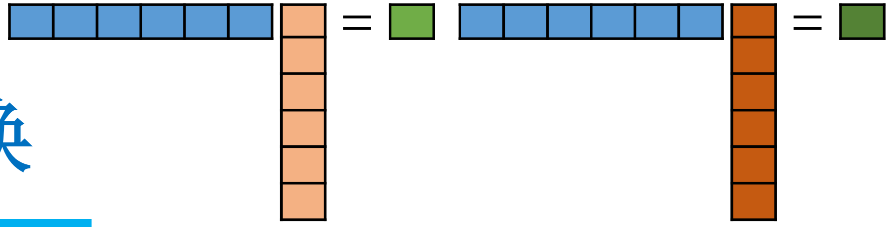
目标：降维过程中保护数据之间距离

$(1-\varepsilon)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2\le\|\tilde{\boldsymbol{x}}_i-\tilde{\boldsymbol{x}}_j\|_2\le(1+\varepsilon)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$

需要分析的是　　　？（回忆$l_2$距离定义）

随机变量 $f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j-y_j)^2$ 即 $\|\mathbf{x}-\mathbf{y}\|_2^2$.

$$\mathbf{Var}[f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})]=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j = 1, \ldots, k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_r(\boldsymbol{x}) - f_r(\boldsymbol{y})$相对于$\boldsymbol{x} - \boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{r} \rangle = \sum_{j=1}^{k} x_j r_j.}$$

$$\boxed{f_{\mathbf{r}}(\mathbf{x}) - f_{\mathbf{r}}(\mathbf{y}) = \sum_{j=1}^{k} x_j r_j - \sum_{j=1}^{k} y_j r_j = \sum_{j=1}^{k} \underbrace{(x_j - y_j)}_{\text{看作常数}} r_j.}$$

每个$r_j$都是服从正态分布的随机变量

目标：降维过程中保护数据之间距离

$(1-\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \le \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|_2 \le (1+\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$

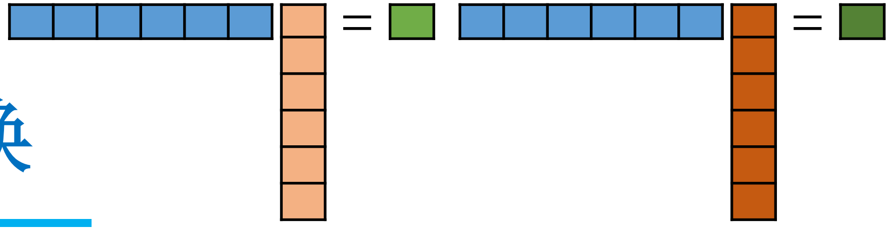
需要分析的是$|f_r(\boldsymbol{x}) - f_r(y)|$（一维数据间距离）

注意绝对值符号

随机变量 $f_{\mathbf{r}}(\mathbf{x}) - f_{\mathbf{r}}(\mathbf{y})$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j - y_j)^2$ 即 $\|\mathbf{x} - \mathbf{y}\|_2^2.$

$$\mathbf{Var}[f_r(\boldsymbol{x}) - f_r(y)] = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

# JL转换



分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\dots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_r(\boldsymbol{x})-f_r(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_jr_j.}$$

$$\boxed{f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_jr_j-\sum_{j=1}^{k}y_jr_j=\sum_{j=1}^{k}(\underbrace{x_j-y_j}_{\text{看作常数}})r_j.}$$

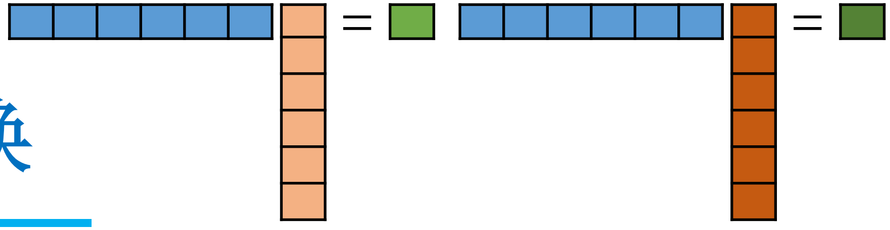
每个$r_j$都是服从正态分布的随机变量

目标：降维过程中保护数据之间距离
$(1-\varepsilon)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2\le\|\tilde{\boldsymbol{x}}_i-\tilde{\boldsymbol{x}}_j\|_2\le(1+\varepsilon)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$

改为分析$\left(f_r(\boldsymbol{x})-f_r(\boldsymbol{y})\right)^2$

随机变量 $f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j-y_j)^2$ 即 $\|\mathbf{x}-\mathbf{y}\|_2^2.$

$$\mathbf{Var}[f_r(\boldsymbol{x})-f_r(\boldsymbol{y})]=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\dots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_r(\boldsymbol{x})-f_r(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

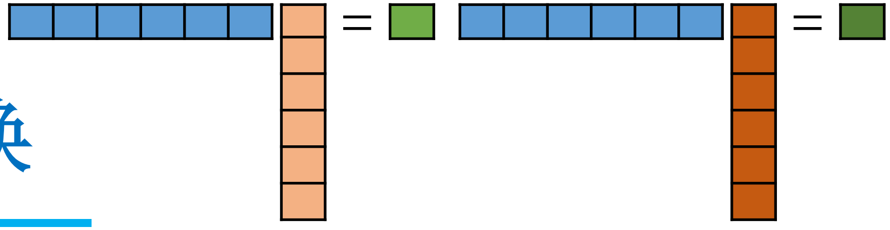
$$\boxed{f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_j r_j-\sum_{j=1}^{k}y_j r_j=\sum_{j=1}^{k}\underbrace{(x_j-y_j)}_{\text{看作常数}}r_j.}$$

每个$r_j$都是服从正态分布的随机变量

改为分析$\left(f_r(\boldsymbol{x})-f_r(y)\right)^2$

随机变量 $f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j-y_j)^2$ 即 $\|\mathbf{x}-\mathbf{y}\|_2^2$.

$$\mathbf{Var}[f_r(\boldsymbol{x})-f_r(y)]=\mathbb{E}\left[\left(f_r(\boldsymbol{x})-f_r(y)\right)^2\right]-(\mathbb{E}[f_r(\boldsymbol{x})-f_r(y)])^2=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$




# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\dots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.$$

$$f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})=\sum_{j=1}^{k}x_j r_j-\sum_{j=1}^{k}y_j r_j=\sum_{j=1}^{k}\underbrace{(x_j-y_j)}_{}r_j.$$

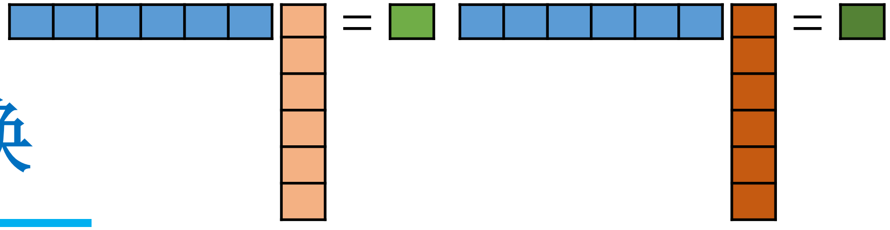
看作常数

每个$r_j$都是服从正态分布的随机变量

改为分析$\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(y)\right)^2$

随机变量 $f_{\mathbf{r}}(\mathbf{x})-f_{\mathbf{r}}(\mathbf{y})$ 服从正态分布，期望为0，方差为 $\sum_{j=1}^{k}(x_j-y_j)^2$ 即 $\|\mathbf{x}-\mathbf{y}\|_2^2$.

$$\mathbb{E}\left[\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(y)\right)^2\right]=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$



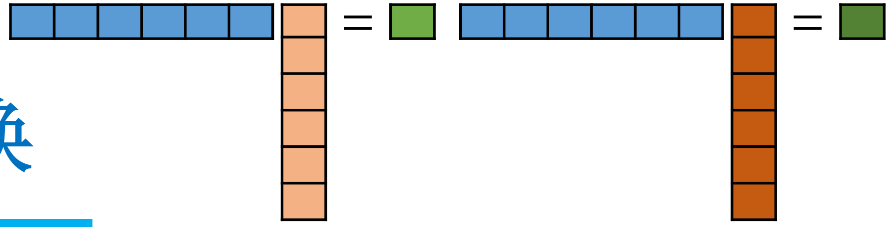
# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\ldots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

$$\mathbb{E}\left[\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(y)\right)^2\right]=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

对两个高维数据$\boldsymbol{x}$和$\boldsymbol{y}$分别用函数$f_{\boldsymbol{r}}(\cdot)$做变换，得到的$\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(y)\right)^2$值可以用于近似$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j = 1, \dots, k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x} - \boldsymbol{y}$的变化

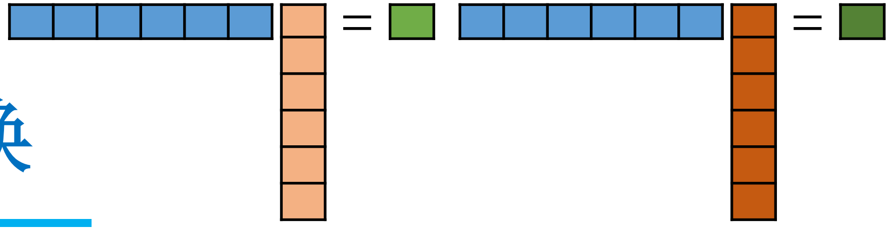
$$\boxed{f_{\mathbf{r}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{r} \rangle = \sum_{j=1}^{k} x_j r_j.}$$

$$\mathbb{E}\left[\left(f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2\right] = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

对两个高维数据$\boldsymbol{x}$和$\boldsymbol{y}$分别用函数$f_{\boldsymbol{r}}(\cdot)$做变换，得到的$\left(f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2$值可以用于近似$\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$

问题一，现在仅是$\left(f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2$的期望等于$\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$，$\left(f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2$的具体取值可能明显偏离$\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$，如何降低随机性?

问题二，$\mathbb{E}\left[\left(f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2\right] = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$不代表$\mathbb{E}[|f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})|] = \|\boldsymbol{x} - \boldsymbol{y}\|_2$，在问题一存在的基础上，$|f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})|$可能进一步偏离$\|\boldsymbol{x} - \boldsymbol{y}\|_2$



# JL转换

分析当向量$\boldsymbol{r}$中的每个元素$r_j$（$j=1,\ldots,k$）都按标准正态分布$\mathcal{N}(0,1)$随机取值时$f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})$相对于$\boldsymbol{x}-\boldsymbol{y}$的变化

$$\boxed{f_{\mathbf{r}}(\mathbf{x})=\langle\mathbf{x},\mathbf{r}\rangle=\sum_{j=1}^{k}x_j r_j.}$$

$$\mathbb{E}\left[\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2\right]=\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

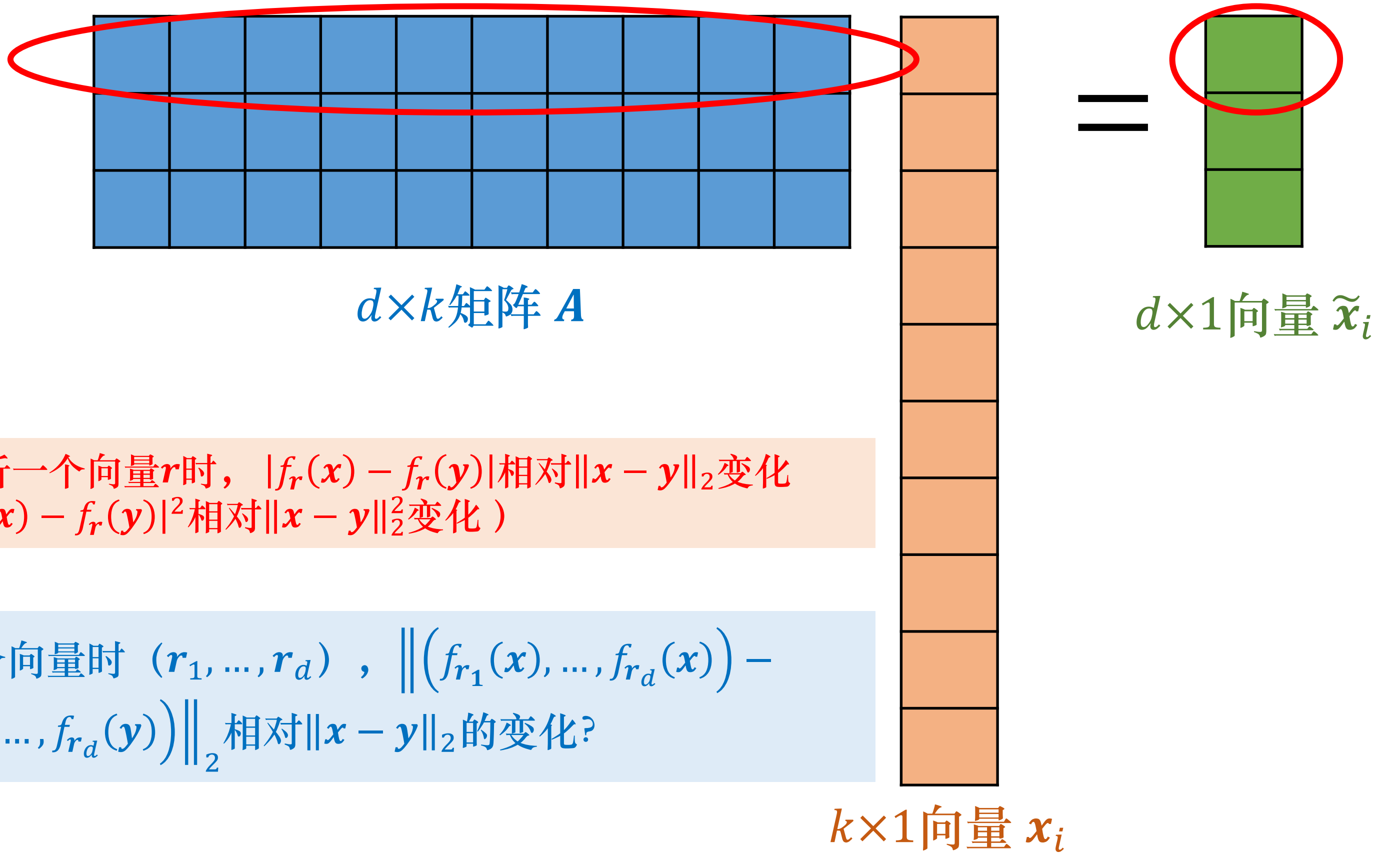
对两个高维数据$\boldsymbol{x}$和$\boldsymbol{y}$分别用函数$f_{\boldsymbol{r}}(\cdot)$做变换，得到的$\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2$值可以用于近似$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$

问题一，$\left(f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})\right)^2$的具体取值可能明显偏离$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$，如何降低随机性?

问题二，$|f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})|$可能进一步偏离$\|\boldsymbol{x}-\boldsymbol{y}\|_2$

解决思路？多个哈希函数?

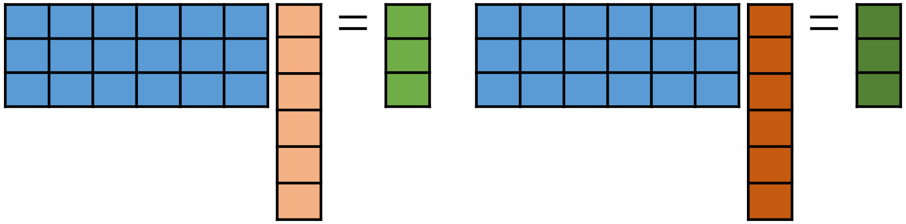
# JL转换




现仅分析一个向量$\boldsymbol{r}$时，$|f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})|$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$变化（或$|f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})|^2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$变化）

当有$d$个向量时（$\boldsymbol{r}_1,\ldots,\boldsymbol{r}_d$），$\left\|\left(f_{\boldsymbol{r}_1}(\boldsymbol{x}),\ldots,f_{\boldsymbol{r}_d}(\boldsymbol{x})\right)-\left(f_{\boldsymbol{r}_1}(\boldsymbol{y}),\ldots,f_{\boldsymbol{r}_d}(\boldsymbol{y})\right)\right\|_2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$的变化?

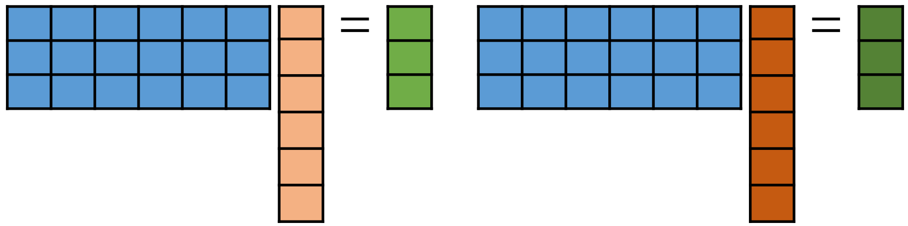
# JL转换



现仅分析一个向量$\boldsymbol{r}$时，$|f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})|$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$变化（或$|f_{\boldsymbol{r}}(\boldsymbol{x}) - f_{\boldsymbol{r}}(\boldsymbol{y})|^2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$变化）

当有$d$个向量时（$\boldsymbol{r}_1,\dots,\boldsymbol{r}_d$），$\left\|\left(f_{\boldsymbol{r}_1}(\boldsymbol{x}),\dots,f_{\boldsymbol{r}_d}(\boldsymbol{x})\right) - \left(f_{\boldsymbol{r}_1}(\boldsymbol{y}),\dots,f_{\boldsymbol{r}_d}(\boldsymbol{y})\right)\right\|_2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$的变化?

# JL转换



现仅分析一个向量$\boldsymbol{r}$时，$|f_r(\boldsymbol{x}) - f_r(\boldsymbol{y})|$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$变化（或$|f_r(\boldsymbol{x}) - f_r(\boldsymbol{y})|^2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$变化）

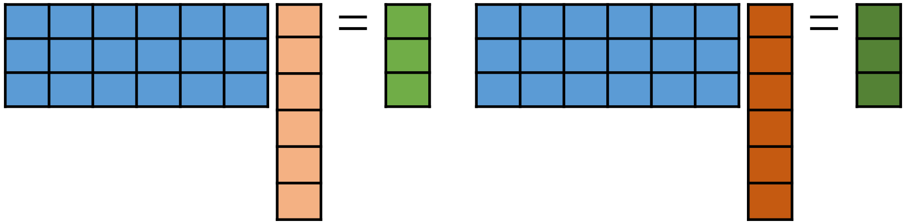
当有$d$个向量时（$\boldsymbol{r}_1, \dots, \boldsymbol{r}_d$），$\left\|\left(f_{r_1}(\boldsymbol{x}), \dots, f_{r_d}(\boldsymbol{x})\right) - \left(f_{r_1}(\boldsymbol{y}), \dots, f_{r_d}(\boldsymbol{y})\right)\right\|_2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$的变化?

直觉上，可以看到如下结果：

$\left\|\frac{1}{\sqrt{d}}\left(f_{r_1}(\boldsymbol{x}), \dots, f_{r_d}(\boldsymbol{x})\right) - \frac{1}{\sqrt{d}}\left(f_{r_1}(\boldsymbol{y}), \dots, f_{r_d}(\boldsymbol{y})\right)\right\|_2$相比于 $|f_r(\boldsymbol{x}) - f_r(y)|$能更好地近似$\|\boldsymbol{x}-\boldsymbol{y}\|_2$

绝对值符号内是一个向量　　　　　　绝对值符号内是一个标量

# JL转换



现仅分析一个向量$\boldsymbol{r}$时，$|f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})|$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$变化
（或$|f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(\boldsymbol{y})|^2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$变化）

当有$d$个向量时（$\boldsymbol{r}_1,\dots,\boldsymbol{r}_d$），$\left\|\left(f_{\boldsymbol{r}_1}(\boldsymbol{x}),\dots,f_{\boldsymbol{r}_d}(\boldsymbol{x})\right)-\left(f_{\boldsymbol{r}_1}(\boldsymbol{y}),\dots,f_{\boldsymbol{r}_d}(\boldsymbol{y})\right)\right\|_2$相对$\|\boldsymbol{x}-\boldsymbol{y}\|_2$的变化?

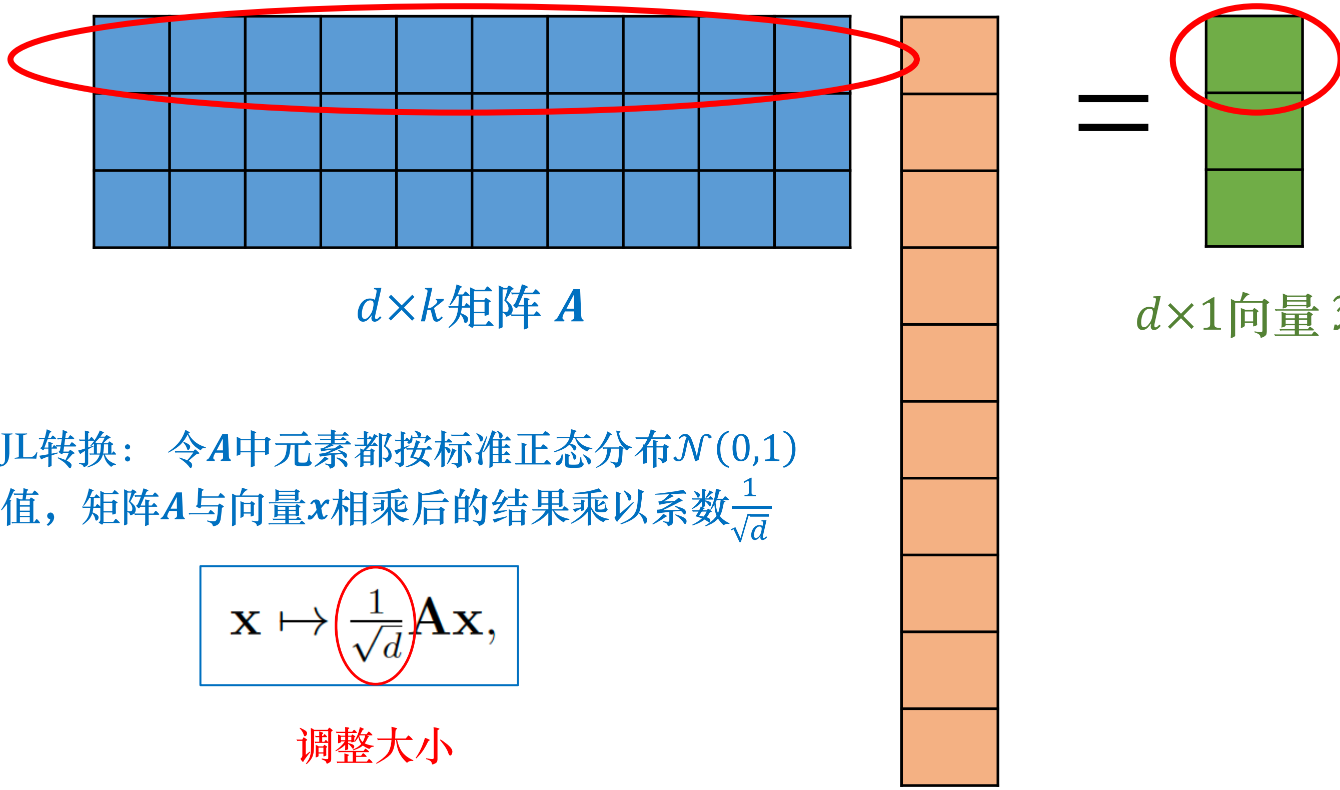
例 $\left\|\frac{1}{\sqrt{2}}(2,2)-\frac{1}{\sqrt{2}}(1,1)\right\|_2=|2-1|$

直觉上，可以看到如下结果:

$\left\|\frac{1}{\sqrt{d}}\left(f_{\boldsymbol{r}_1}(\boldsymbol{x}),\dots,f_{\boldsymbol{r}_d}(\boldsymbol{x})\right)-\frac{1}{\sqrt{d}}\left(f_{\boldsymbol{r}_1}(\boldsymbol{y}),\dots,f_{\boldsymbol{r}_d}(\boldsymbol{y})\right)\right\|_2$相比于$|f_{\boldsymbol{r}}(\boldsymbol{x})-f_{\boldsymbol{r}}(y)|$能更好地近似$\|\boldsymbol{x}-\boldsymbol{y}\|_2$

绝对值符号内是一个向量　　　　绝对值符号内是一个标量

# JL转换




完整的JL转换： 令$\boldsymbol{A}$中元素都按标准正态分布$\mathcal{N}(0,1)$随机取值，矩阵$\boldsymbol{A}$与向量$\boldsymbol{x}$相乘后的结果乘以系数$\frac{1}{\sqrt{d}}$

$$\boxed{\mathbf{x} \mapsto \frac{1}{\sqrt{d}}\mathbf{A}\mathbf{x},}$$

调整大小

（等价于：令$\boldsymbol{A}$中元素都按正态分布$\mathcal{N}\left(0,\frac{1}{d}\right)$随机取值，矩阵$\boldsymbol{A}$与向量$\boldsymbol{x}$相乘后不用调整系数）

# JL转换

**欧几里得低失真嵌入（Euclidean Low Distortion Embedding）**

给定$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^k$及误差$\varepsilon \geq 0$，求满足以下条件的$\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_n \in \mathbb{R}^d$（$d \ll k$）：

$$(1-\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \leq \left\|\widetilde{\boldsymbol{x}}_i - \widetilde{\boldsymbol{x}}_j\right\|_2 \leq (1+\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2, \forall i, j = 1, \dots, n.$$

**JL引理（Johnson - Lindenstrauss Lemma）**

对任意$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^k$及误差$\varepsilon \geq 0$，存在一个$d \times k$矩阵$\boldsymbol{A}$（$d = O\left(\frac{\log n}{\varepsilon^2}\right)$）使得当$\widetilde{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i$时有

$$(1-\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \leq \left\|\widetilde{\boldsymbol{x}}_i - \widetilde{\boldsymbol{x}}_j\right\|_2 \leq (1+\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2, \forall i, j = 1, \dots, n.$$

若矩阵$\boldsymbol{A}$的每一个元素都依据$\mathcal{N}\left(0, \frac{1}{d}\right)$随机取值，则上式能以较大概率被满足。

# JL转换

**欧几里得低失真嵌入（Euclidean Low Distortion Embedding）**

给定$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^k$及误差$\varepsilon \geq 0$，求满足以下条件的$\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_n \in \mathbb{R}^d$（$d \ll k$）：

$$(1-\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \leq \left\|\widetilde{\boldsymbol{x}}_i - \widetilde{\boldsymbol{x}}_j\right\|_2 \leq (1+\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2, \forall i, j = 1, \dots, n.$$

**JL引理（Johnson - Lindenstrauss Lemma）**

对任意$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^k$及误差$\varepsilon \geq 0$，存在一个$d \times k$矩阵$\boldsymbol{A}$（$d = O\left(\frac{\log n}{\varepsilon^2}\right)$）使得当$\widetilde{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i$时有

存在令不等式严格成立的矩阵$\boldsymbol{A}$

$$(1-\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 \leq \left\|\widetilde{\boldsymbol{x}}_i - \widetilde{\boldsymbol{x}}_j\right\|_2 \leq (1+\varepsilon)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2, \forall i, j = 1, \dots, n.$$

若矩阵$\boldsymbol{A}$的每一个元素都依据$\mathcal{N}\left(0, \frac{1}{d}\right)$随机取值，则上式能以较大概率被满足。

依据正态分布随机生成的矩阵$\boldsymbol{A}$较可能令不等式成立（具体概率值的分析略）

# JL转换

**欧几里得低失真嵌入（Euclidean Low Distortion Embedding）**

给定$\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\in\mathbb{R}^k$及误差$\varepsilon\geq 0$，求满足以下条件的$\widetilde{\boldsymbol{x}}_1,\ldots,\widetilde{\boldsymbol{x}}_n\in\mathbb{R}^d$（$d\ll k$）：

$$(1-\varepsilon)\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2\leq\left\|\widetilde{\boldsymbol{x}}_i-\widetilde{\boldsymbol{x}}_j\right\|_2\leq(1+\varepsilon)\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2,\forall i,j=1,\ldots,n.$$

**JL引理（Johnson - Lindenstrauss Lemma）**

对任意$\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\in\mathbb{R}^k$及误差$\varepsilon\geq 0$，存在一个$d\times k$矩阵$\boldsymbol{A}$（$d=O\left(\frac{\log n}{\varepsilon^2}\right)$）使得当$\widetilde{\boldsymbol{x}}_i=\boldsymbol{A}\boldsymbol{x}_i$时有

$$(1-\varepsilon)\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2\leq\left\|\widetilde{\boldsymbol{x}}_i-\widetilde{\boldsymbol{x}}_j\right\|_2\leq(1+\varepsilon)\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2,\forall i,j=1,\ldots,n.$$

若矩阵$\boldsymbol{A}$的每一个元素都依据$\mathcal{N}\left(0,\frac{1}{d}\right)$随机取值，则上式能以较大概率被满足。

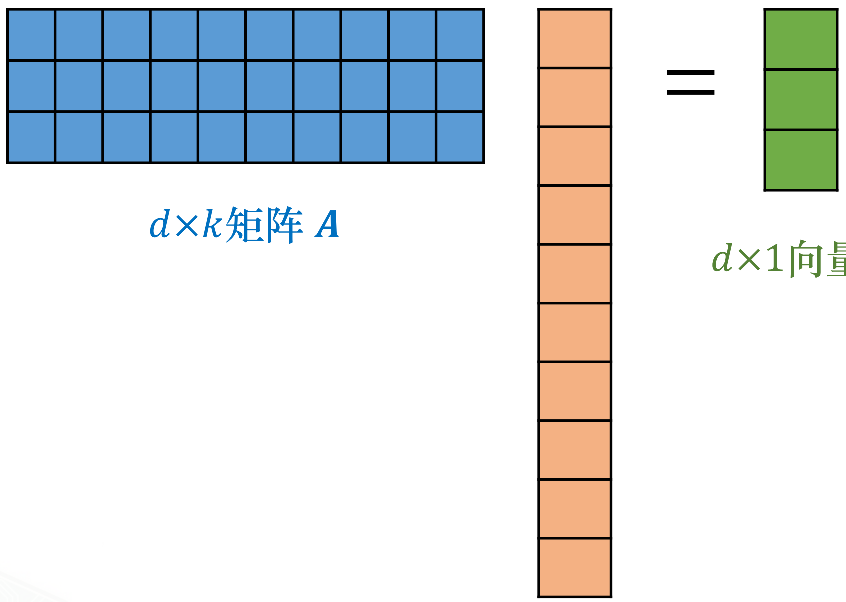
注意：为确保低失真，$d$值有下限（即JL转换不能在低失真约束下无限制地压缩数据）

例：$\varepsilon=0.05, n=10^5$时，$d$值约为6600，不受$k$的影响（如$k=10^{12}$依然可以低失真压缩至$d$维）

# JL转换

JL转换的优点：数据无关（data oblivious）

- 降维映射函数（即矩阵$\boldsymbol{A}$的选择）与数据$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$的具体取值无关
- 数据$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$可以以数据流的形式输入，仅需很小的存储空间即可依次得到$\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_n$
- 可以在多个服务器并行




$d \times k$矩阵 $\boldsymbol{A}$

$d \times 1$向量 $\tilde{\boldsymbol{x}}_i$

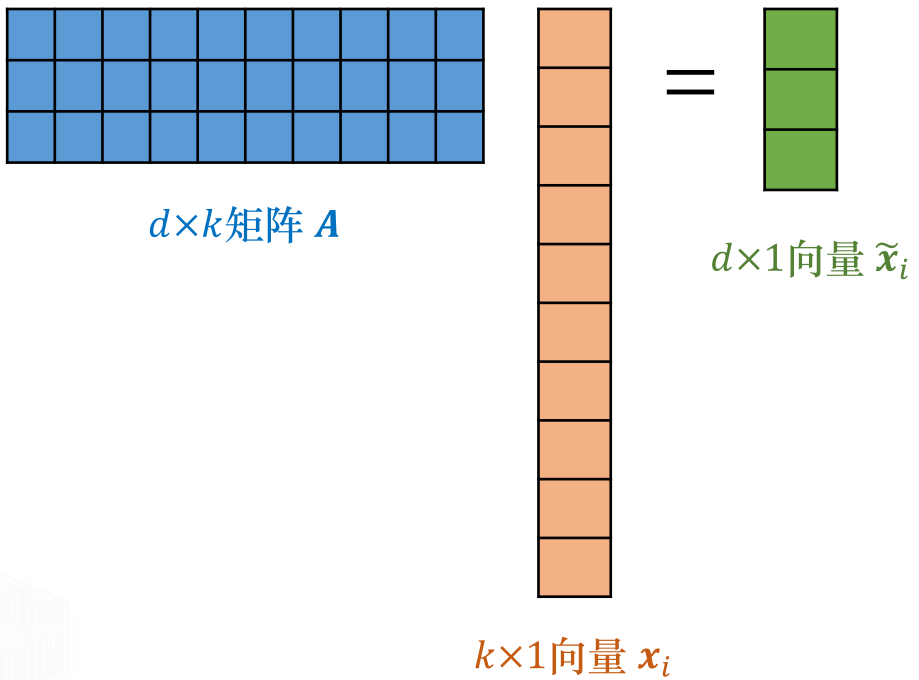
$k \times 1$向量 $\boldsymbol{x}_i$

# JL转换

另一种降维方法（属于谱分析的）主成分分析（PCA）不具备数据无关的特性

JL转换的优点：数据无关（data oblivious）

- 降维映射函数（即矩阵$\boldsymbol{A}$的选择）与数据$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$的具体取值无关
- 数据$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$可以以数据流的形式输入，仅需很小的存储空间即可依次得到$\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n$
- 可以在多个服务器并行

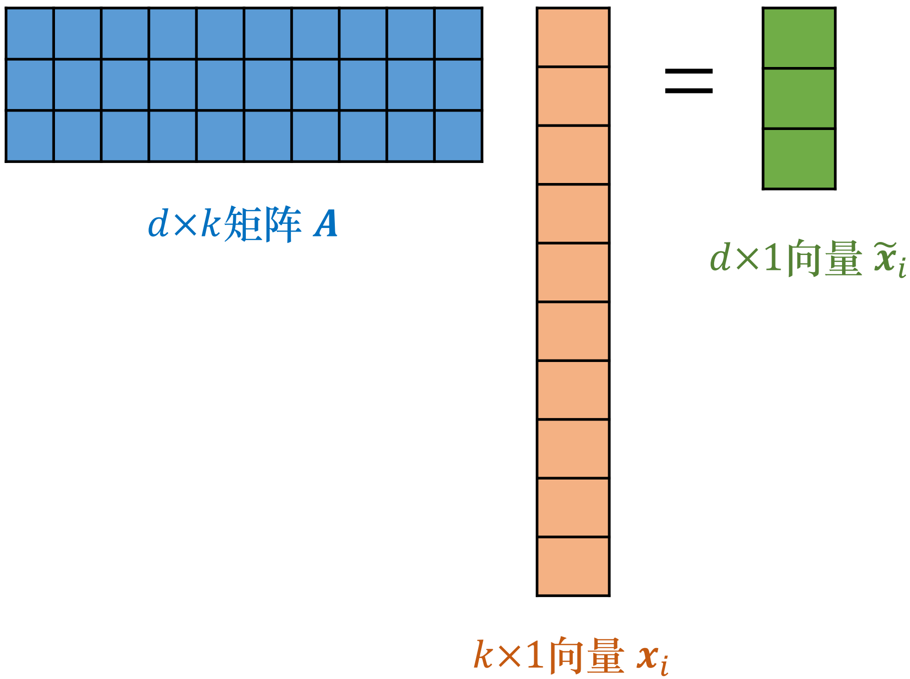


# JL转换

JL转换的改进：当$k$值非常大时，$\boldsymbol{Ax}$运算耗时

- 将在实数范围内随机取值的$A_{ij}$改为在集合$\{-1,0,1\}$中随机取值

可以避免乘法运算，只进行加减运算



# JL转换

JL转换的改进：当$k$值非常大时，$\boldsymbol{Ax}$运算耗时

- 将在实数范围内随机取值的$A_{ij}$改为在集合$\{-1,0,1\}$中随机取值

可以避免乘法运算，只进行加减运算

[HTML] Database-friendly random projections: Johnson-Lindenstrauss with binary coins

D Achlioptas - Journal of computer and System Sciences, 2003 - Elsevier

A classic result of Johnson and Lindenstrauss asserts that any set of n points in d-dimensional Euclidean space can be embedded into k-dimensional Euclidean space—where k is logarithmic in n and independent of d—so that all pairwise distances are maintained within an arbitrarily small factor. All known constructions of such embeddings involve projecting the n points onto a spherically random k-dimensional hyperplane through the origin. We give two constructions of such embeddings with the property that all elements …

☆ Save 99 Cite Cited by 1465 Related articles All 11 versions

**Theorem 1.1.** *Let $P$ be an arbitrary set of $n$ points in $\mathbb{R}^d$, represented as an $n \times d$ matrix $A$. Given $\varepsilon, \beta > 0$ let*

$$k_0 = \frac{4+2\beta}{\varepsilon^2/2 - \varepsilon^3/3} \log n.$$

*For integer $k \geqslant k_0$, let $R$ be a $d \times k$ random matrix with $R(i,j) = r_{ij}$, where $\{r_{ij}\}$ are independent random variables from either one of the following two probability distributions:*

$$r_{ij} = \begin{cases} +1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2, \end{cases} \tag{1}$$

$$r_{ij} = \sqrt{3} \times \begin{cases} +1 & \text{with probability } 1/6, \\ 0 & \text{with probability } 2/3, \\ -1 & \text{with probability } 1/6. \end{cases} \tag{2}$$

*Let*

$$E = \frac{1}{\sqrt{k}} AR$$

*and let $f : \mathbb{R}^d \to \mathbb{R}^k$ map the ith row of $A$ to the ith row of $E$.*
*With probability at least $1 - n^{-\beta}$, for all $u, v \in P$*

$$(1-\varepsilon)||u-v||^2 \leqslant ||f(u)-f(v)||^2 \leqslant (1+\varepsilon)||u-v||^2.$$

论文中符号定义与本课件相反：要将$d$维数据压缩至$k$维数据

# 本讲小结

- 相似搜索问题及相似度衡量
- k维树方法
- 数据降维及JL转换

# 主要参考资料


Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox> Lecture Notes

Cameron Musco <COMPSCI 514 - Algorithms for Data Science> Slides

Ioannis Emiris <Computational Geometry Search in High dimension and kd-trees> Slides

Sham Kakade < CSE547/STAT548 - Machine Learning for Big Data> Slides

Kamesh Munagala <CPS290 - Algorithmic Foundations of Data Science – Similarity Search> Lecture Notes

# 谢谢！