

# 模式识别

“稀疏”数据sparse  
未对齐数据misaligned

吴建鑫  
南京大学计算机系，2018

# 目标

- ✓ 对介绍的较前沿的知识 (sparsity) 正确了解其含义
  - 了解 sparsity 适用的范围和可能的应用
- ✓ 掌握动态规划 dynamic programming 的基本算法
- ✓ 掌握动态时间弯曲 dynamic time warping 的概念、算法和适用范围
- ✓ 提高目标
  - 进一步能通过独立阅读、了解 sparse learning, 如其优化算法、在研究中的应用、理论等
  - 进一步能通过独立阅读、了解其他未对齐数据的处理

能识别吗？

加里你熊丢胆白没行字  
沿胆没字还可以简化50%  
糗事百科



我居北海君南海

黄沙白雾昼常昏

# 为什么？

- ✓ 我们靠什么能够从字的半边猜出正确的字来？
- ✓ 我们为什么不能靠什么从字的半边猜出正确的字来？

# 稀疏

---

sparse

# Tutorial: Yi Ma **et al.**

✓ <http://www.eecs.berkeley.edu/~yang/courses/ECCV2012/ECCV12-lecture1.pdf>

# 阅读提示

- ✓ 进一步 Laplace distribution: p9
- ✓ Underdetermined linear system: 观察值（已知量）少于变量（未知量）
- ✓ 进一步 Wavelet 小波: p18, DCT 离散余弦变换: p21
- ✓  $\ell_0$  “norm”: 向量 $\mathbf{x}$ 的 $\ell_0$ 是其中非零元素的个数
  - 不是严格意义上的norm, 很不好优化
- ✓ Relax: 当原来的问题不好解时, 简化
  - 发现一个容易解的问题, 与原问题相似（包括最优解的相似性）
  - 需要做的: 理论或实践确保最优解的近似或相似

# 什么是稀疏？

## ✓ 向量 $\mathbf{x}$

- 计算其非0元素的个数

## ✓ 那么，图像（例如人脸图像）是稀疏的吗？在什么意义上？

- 不是在原来的空间（每一维代表一个像素）
- 而是在某种更有效的（通常高维但稀疏的）表达方式上，如
  - 人脸图像在光照条件变化是存在于一个低维子空间中
  - 视频监控图像中背景是低频（变化慢）而前景是高频（变化快）的
  - 语音信号中的语音和噪声所处的频段，...
- 总之，需要正确理解稀疏sparse的含义

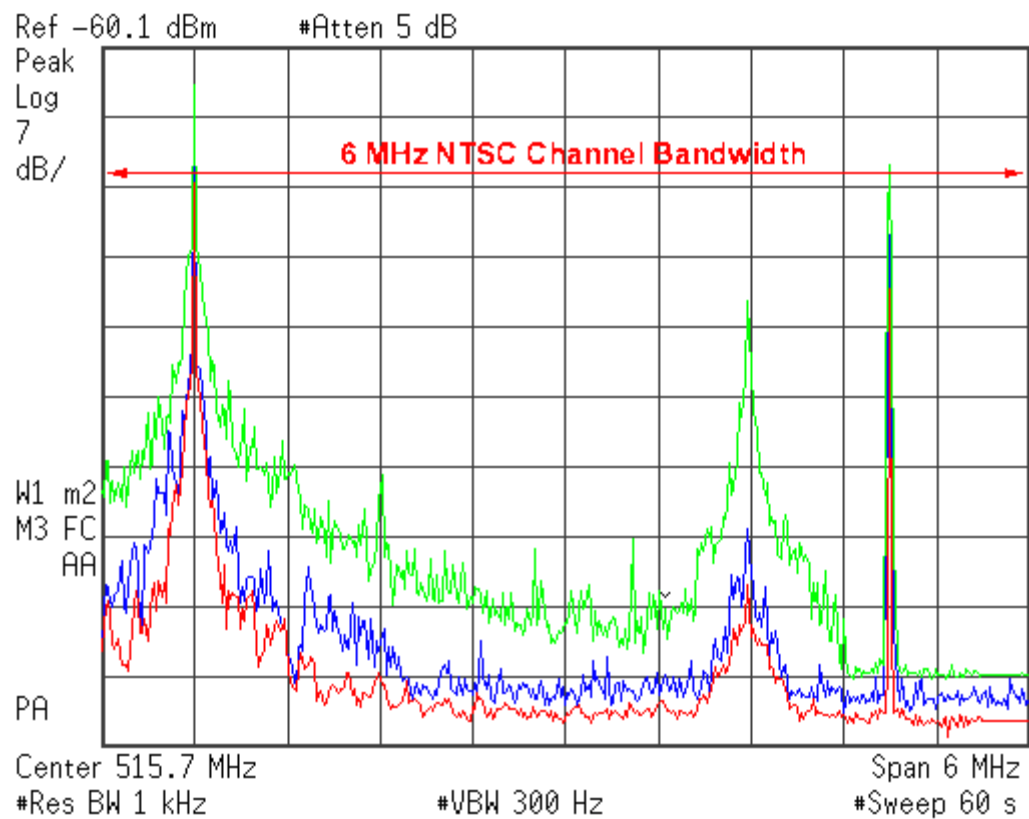


# 动态时间弯曲

---

DTW: Dynamic Time Warping

# 怎样比较相似度 (1) ?

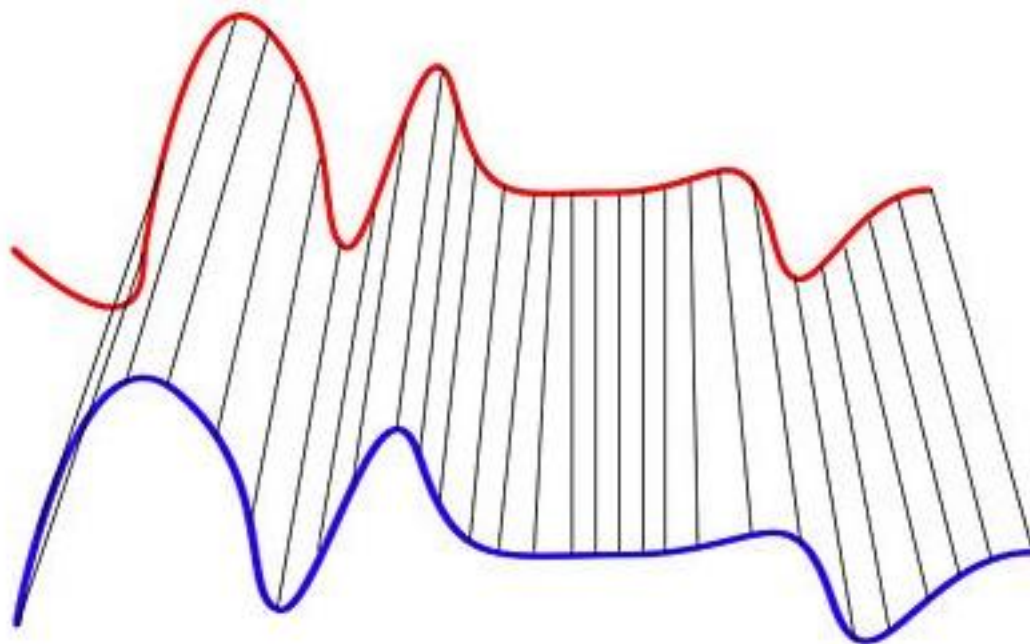


# 怎样比较相似度 (2) ?



视频来自UCF50数据库<http://crcv.ucf.edu/data/UCF50.php>

# Alignment 对齐



Dynamic Time Warping Matching

一个项目是一个时间序列time series, 包含若干顺序的数据

# 要求

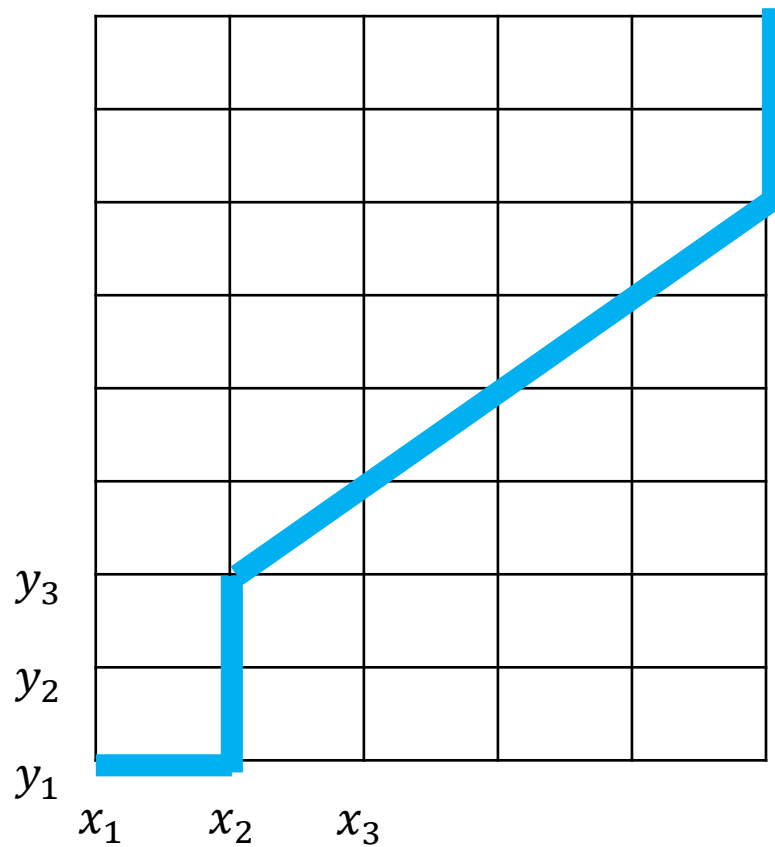
- ✓ 假设两组（顺序的）数据  $\mathbf{x} = (x_1, \dots, x_n)$ ,  $\mathbf{y} = (y_1, \dots, y_m)$ 
  - 很可能  $m \neq n$
  - 对任意的  $x_i, y_j$ , 存在函数  $d(x_i, y_j)$  描述其距离
  - 如果能找到好的匹配, 那么就可以计算  $\mathbf{x}$  和  $\mathbf{y}$  的距离
- ✓ 那么, 好的匹配应该满足什么条件?
  - 若  $x_i, y_j$  匹配, 那么  $d(x_i, y_j)$  越小越好
  - 匹配可以有跳过的数据, 如  $x_1 \leftrightarrow y_2, x_3 \leftrightarrow y_3$ 
    - 可以跳过  $\mathbf{x}$  的, 也可以跳过  $\mathbf{y}$  中的数据
  - 匹配是顺序的, 如果  $x_i \leftrightarrow y_j, x_{i+1} \leftrightarrow y_k$ , 那么  $j \leq k$
  - 选择总距离最小的匹配

# 有问题吗？

- ✓ 假设距离  $d(x_i, y_j) \geq 0$ 
  - 如果任何一个数据都不选择进入匹配，那么总距离是？
- ✓ 解决的方法
  - 要求  $\forall i, x_i$  必须和一个  $y_j$  匹配，反之亦然
  - 但是，一个  $x_i$  可以和多个  $y_j$  匹配，一个  $y_i$  也可以和多个  $x_j$  匹配
  - 如  $x_1 \leftrightarrow y_2, x_1 \leftrightarrow y_3, x_2 \leftrightarrow y_3$

# 有效的可视化

$y_{1:9}$



$x_{1:6}$

# 形式化formalize

✓ 匹配变成了发现最佳路径，那么，什么是最佳路径？

- $\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_m)$
- 任意一条路径表示为一系列二维坐标： $(r_i, t_i)$ ，上面提到的要求可以翻译为：
  - 路径的长度是多少？
  - Without loss of generality (WLOG)，假设  $n \leq m$ ，
  - 最短的路径是  $\max(n, m)$ ，尽可能走对角线
  - 最长的路径是  $n + m - 1$ （不走任何对角线）
  - 每个  $x_i$  和  $y_j$  都必须有匹配的对象
  - $r_1 = 1, r_K = n, r_{i+1} - r_i = 0/1$
  - $t_1 = 1, t_K = m, t_{i+1} - t_i = 0/1$



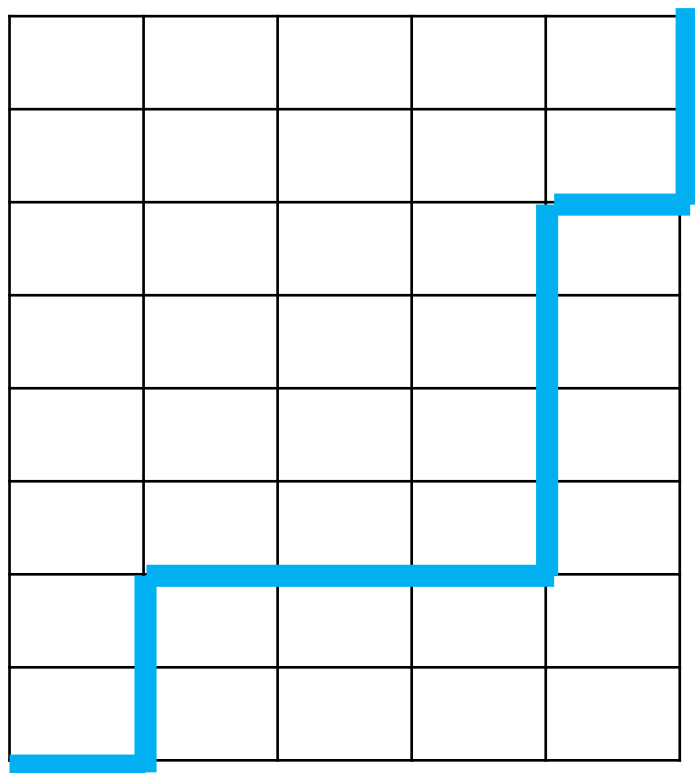
# 形式化formalize (2)

- 匹配可以有跳过的数据
- 匹配是顺序的
- 自动满足
- 若 $x_i, y_j$ 匹配, 那么 $d(x_i, y_j)$ 越小越好
- 选择总距离最小的匹配

$$D(n, m) = \min_{(\mathbf{r}, \mathbf{t})} \sum_{i=1}^{l(\mathbf{r}, \mathbf{t})} d(x_{r_i}, y_{t_i})$$

其中 $(\mathbf{r}, \mathbf{t}) = \{(r_i, t_i)\}$ ,  $i = 1, 2, \dots$  是任意一条满足以上条件的路径, 而 $l(\mathbf{r}, \mathbf{t})$ 是该路径的长度。

# 那么，如何求解呢？



✓ 如果只允许横竖路径

- 那么一共有  $\binom{n+m-2}{n-1} = \binom{n+m-2}{m-1}$

- 读  $(n+m-2)$  choose  $(n-1)$

- 指数级增长，没有办法列举

✓ 如果还允许对角线路径

- 那么可能的路径数目更多

✓ 怎样求解最佳路径？

# 拆成若干小规模的子问题

(6, 9) ✓ 若有先知oracle告知 $(i, j)$ 一定存在于最佳路径，怎么求？

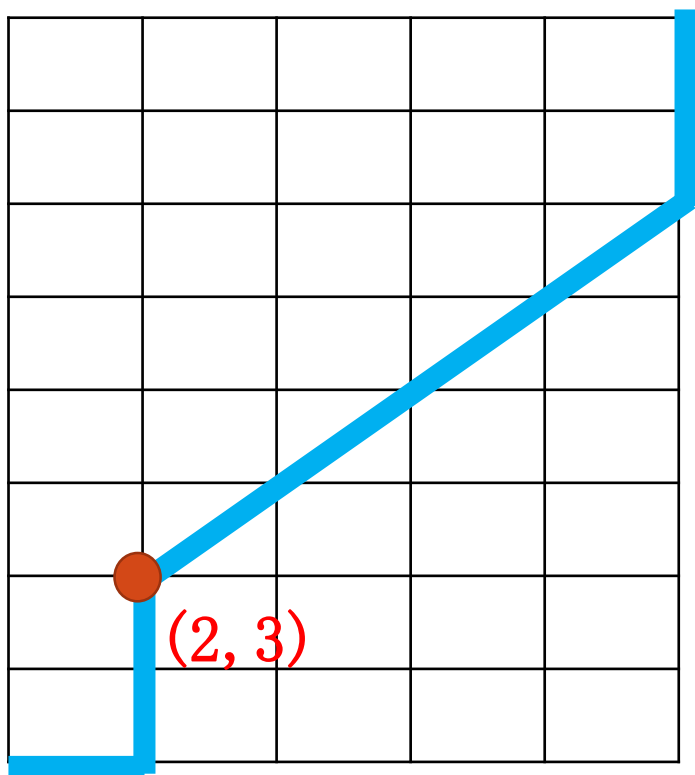
✓ 为什么要拆成小问题？

✓ 什么样的小问题很容易解？

- 只走一步！

✓ 没有先知，怎么办？

- 尝试所有可能的 $(i, j)$ ！



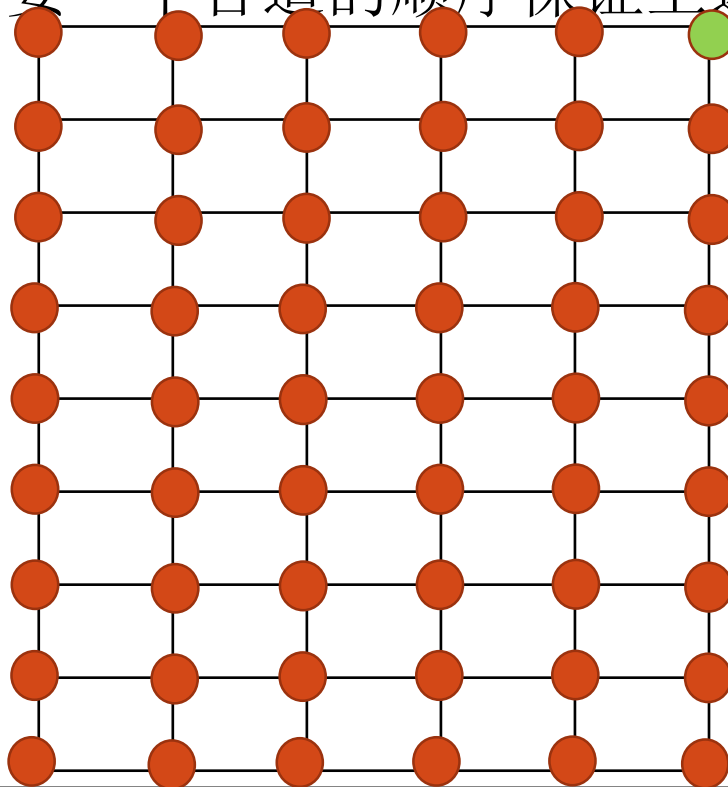
(1, 1)

# 只走一步

- ✓ 路径的起点总是  $(1, 1)$
- ✓ 记到  $(i, j)$  的最短距离为  $D(i, j)$ ，要求的是  $D(n, m)$
- ✓ 如果只走一步，那么怎样达到  $(n, m)$ ?
  - 横:  $D(n - 1, m) + d(n, m)$
  - 竖:  $D(n, m - 1) + d(n, m)$
  - 对角:  $D(n - 1, m - 1) + d(n, m)$
- ✓ 那么，  
 $D(n, m)$   
 $= d(n, m) + \min(D(n - 1, m), D(n, m - 1), D(n - 1, m - 1))$

# 保证顺序

- ✓ 也就是说，求解 $(i, j)$ 时需要保证到达 $(i - 1, j)$ 、 $(i, j - 1)$ 、 $(i - 1, j - 1)$ 的最佳路径已知！
  - 以此类推，求解 $D(n, m)$ 需要整个网格已知
  - 所以需要有一个合适的顺序保证上述要求



# 实现中的问题

## ✓ 上页的顺序相当于

- for  $i=1$  to  $n$ 
  - for  $j=1$  to  $m$ 
    - ◆ do one step

## ✓ 如果两个循环颠倒顺序

- 答案对吗?
- 计算复杂度一样嘛? 是多少?
- 哪种更好?

## ✓ 如何处理边界问题?

- $i=1$ 的时候  $i-1=0$ , 没有定义

# 小结

- ✓ 比较两组顺序数据的，若数据不对齐
  - 可以使用DTW
- ✓ 形式化在解法中没有使用
  - 但是对理解问题非常有帮助
- ✓ 合适的可视化visualization可以发现解法
- ✓ 当问题可以拆解成两个或多个较小问题时，若
  - 小问题容易求解
  - 原问题可以通过小问题的解变换得到
  - 通常可以把很高复杂度的问题用较低复杂度解决
  - 称为dynamic programing动态规划

# 动态规划

- ✓ 把规模大的问题分解成小问题求解
  - 进一步阅读：使用动态规划降低矩阵乘法的复杂度、快速傅里叶变换fast Fourier transform
- ✓ 动态规划通常解决比原问题更多的问题
  - 例如， $D(n, m)$ 只要求一个最佳路径和距离；但是动态规划解法求解了 $nm$ 个最佳路径和距离
  - 所以，有时候主动解决更general的问题反而比解决其中的一个有效
  - 动态规划需要存储子问题的解，有时候存储空间会成为问题



# 进一步的阅读

✓ 如果对本章的内容感兴趣，可以参考如下文献

- Sparse:

- <http://www.eecs.berkeley.edu/~yang/courses/ECCV2012/index.htm>

- Another way to induce sparsity (Bayesian, *cf.* tutorial p9): [http://cs.nju.edu.cn/wujx/paper/PR\\_GGIG.pdf](http://cs.nju.edu.cn/wujx/paper/PR_GGIG.pdf)

- Dynamic Programming:

- [http://en.wikipedia.org/wiki/Dynamic\\_programming](http://en.wikipedia.org/wiki/Dynamic_programming)

- DTW: Fundamentals of speech recognition, L. R. Rabiner and B. Juang

- <http://www.amazon.cn/Fundamentals-of-Speech-Recognition-Rabiner-Lawrence-R/dp/0130151572>