距离测度学习[1]

高涛[2]

1 前言

1.1 什么叫距离测度学习

距离测度（distance metric learning），顾名思义，就是衡量两样本 (模式) 之间距离大小的量。最常见的就是度量空间中的欧氏距离 (L2 范数)，还有曼哈顿距离 (L_1 范数)。当然不同的距离度量有着自己不同的特点，从拓扑的角度来看，不同的度量刻画距离的粗细可能不同，其他的拓扑性质也可能不同。虽然 L_1 和 L_2 范数实际应用简单方面，但是由于理论上不具有旋转和尺度不变形，因此衡量模式间的距离未必总是有意义。另外，由于提取维度的量纲有差异，若直接采用 L_k 范数则意味着每维特征在距离计算中所占权重相同，这样显然不合理。因此我们需要结合数据的性质，学习有效的距离测度来很衡量模式间的距离大小，换句话说，也是衡量模式间的相近程度。

在多元统计中，由于综合了样本之间的各种信息，Mahalanobis 距离应用十分广泛。距离测度学习多半寻找距离测度 $\mathbf{M}$，计算样本之间的 Mahalanobis 测度意义下的距离。

$$d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)=||\mathbf{x}_i-\mathbf{x}_j||^2_{\mathbf{M}}=(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i-\mathbf{x}_j),\mathbf{M}\geq 0$$

非 Mahalanobis 距离学习的则有局部距离测度学习 (local distance learning) 和 Kernel 方法。这些方法的目的都是为了对样本进行线性或非线性变换而获取另一种更具有类别区分性的形式。距离测度学习大致可以分为两类：有监督距离测度学习和无监督距离测度学习，目前也有些半监督方法出现。无监督距离测度学习的主要思想就是低维映射，同时尽可能地保留观测数据间的几何关系，与降维 (dimension reduction) 有很大关系；有监督距离测度学习与分类联系很大，主要思想就是学习的距离测度能使同类数据点收缩靠近，使不同类数据点分离。

1.2 距离测度学习的用途

距离测度应用已经十分广泛，在人脸识别、物体识别、音乐的相似性、人体姿势估计、信息检索、语音识别、手写体识别等领域都有较好的应用。因为距离测度学习的目的即为了衡量样本之间的相近程度，而这也正是模式识别的核心问题之一。大量的机器学习方法，比如 K 近邻、支持向量机、径向基函数网络等分类方法以及 K-means 聚类方法，还有一些基于图的方法，其性能好坏都主要有样本之间的相似度量方法的选择决定，因此针对不同的问题，学习适合样本的距离测度具有很重要的意义。

2 有监督的距离测度学习

所谓有监督的距离测度学习，通俗来说，就是知道了训练数据的类别 (class labels)，然后依此信息和数据的各维特征来对数据进行变换，不过有监督的距离测度需要成对约束 (pairwise constraints)：同类的等价约束 $S=\{(\mathbf{x}_i,\mathbf{x}_j)|\mathbf{x}_i和\mathbf{x}_j属于同类\}$ 和不同类的不等价

[1] R 包开发: https://github.com/road2stat/sdml

[2] 作者单位: 中南大学统计系.

约束 $D = \{(\mathbf{x}_i, \mathbf{x}_j)|\mathbf{x}_i$和$\mathbf{x}_j$属于不同类$\}$，希望得到的效果多半是同类点间的距离收缩，不同类点间的距离扩张分离。(另外，有监督距离测度学习又可细分为：全局有监督距离测度学习和局部有监督距离测度学习。所谓全局就是同时满足所有成对约束的距离测度学习，而局部只需要满足局部的成对约束即可。对于不同的数据类型，两种方法的效果不同：全局方式简便，但是对数据的可分性要求较高；局部方式适应效果好，但是操作起来可能更为复杂。)

但是不管是全局还是局部距离测度学习，都是优化问题，都由优化目标 (objective)，约束条件 (constraints) 和优化算法 (optimization) 三个部分组成，无非是由于指导思想 (idea) 不同，使得其中一部分不同，于是才有了各种方法的涌现。本文也将从这三个方面对各个方法进行描述讲解。

2.1 全局的有监督距离测度学习

有监督的距离测度学习的想法就是变换后使同类数据点收缩靠近，使不同类数据点分离。因此最直接的想法就是对某一类 (同类或者不同类) 进行条件约束，然后取其相反类的最小或最大距离和。

$$\min_{\mathbf{A}} \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S} ||\mathbf{x}_i - \mathbf{x}_j||^2_{\mathbf{A}}$$

$$s.t. \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D} ||\mathbf{x}_i - \mathbf{x}_j||^2_{\mathbf{A}} \geq 1, \mathbf{A} \geq 0.$$

或者

$$\max_{\mathbf{A}} \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D} ||\mathbf{x}_i - \mathbf{x}_j||^2_{A}$$

$$s.t. \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S} ||\mathbf{x}_i - \mathbf{x}_j||^2_{\mathbf{A}} \leq 1, \mathbf{A} \geq 0.$$

这是一个凸优化问题，但是并不属于某特定的凸优化问题，如二次规划、半正定规划 (semi-difinite programming) 等，而且此问题对特征个数没有做任何处理，因此高维是解起来消耗比较大，需要对其进行转化变形。当然也可以直接对该问题进行求解， $\mathbf{A}$ 为对角阵和普通矩阵时分别由相应的算法解决。

另一种常用的全局方法是从概率角度来解决问题，使得数据间取得同类或异类的似然概率最大。首先定义两点间同类或者异类的概率

$$P(y_{ij}|\mathbf{x}_i, \mathbf{x}_j) = 1/(1 + exp(-y_{ij}(||\mathbf{x}_i - \mathbf{x}_j||^2_{\mathbf{A}}) - u))$$

其中

$$y_{ij} = \begin{cases} 1 & (\mathbf{x}_i, \mathbf{x}_j) \in S \\ -1 & (\mathbf{x}_i, \mathbf{x}_j) \in D \end{cases}$$

而我们优化的问题便为

$$\min \quad L(\mathbf{A}, u) = logP(S) + logP(D)$$

$$s.t. \quad \mathbf{A} \geq 0, u \geq 0$$

由于半正定 $\mathbf{A}$ 此条件的限制，使得问题解起来比较复杂，因此我们将问题进行转化解决。常用的就是对矩阵 $\mathbf{M} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T$ 进行特征向量求解，取前 K(K< 维数 m) 个特征向

量 $\mathbf{v}_i$, 令 $\mathbf{A}$ 是这些特征向量的线性组合

$$\mathbf{A} = \sum_{i=1}^{K} r_i \mathbf{v}_i \mathbf{v}_i^T, r_i \geq 0, i = 1, \ldots, K$$

这样就可以获得大部分数据的特征信息，通过这些主要信息来构建距离测度。将 $\mathbf{A}$ 带入原问题，就转化为了用 newton 法都可以求解的凸优化问题。

$$\min \quad L({r_i}^{K_{i=1}}, u)$$

$$s.t. \quad u \geq 0, r_i \geq 0, i = 1, \ldots, k$$

求的 $\mathbf{A}$ 和 u 后，便可以得到任意两个点同类或异类的概率，可以结合 KNN 对测试点进行分类。当然，该方法还有个优点就是矩阵 $\mathbf{M}$ 的构建对数据的类别 (class label) 没有要求，对于无标签的点也可以处理，同样可以利用其特征信息，这就使得适用的数据更广泛了 (类似半监督)。

2.2 局部有监督距离测度学习

所谓的局部有监督距离测度学习，只要满足局部的成对就行，这多半都是为了 KNN 分类器做准备。KNN 分类器是基于实例的学习，它使用具体的实例进行预测，而不必建立或维护源自于数据的模型，这是一种消极的学习方法，所谓见招拆招。积极的学习方法包括决策树和基于规则的分类器，很明显，它们都要根据训练集建立模型，然后在对测试集进行测试。这两种方法各有优劣，消极分类器思想简单，但是计算开销大，每次都需要计算测试样例与训练集的相似度，相反积极分类器寻求一个模型比较困难，误差风险也大，但是一旦比较好的模型建立，对测试样例分类就非常快。在图像检索、信息检索、手写体识别等复杂的实际问题面前，通过训练及建立模型通常是很困难的，因此本文根据实际操作简便的需要，主要介绍与 KNN 分类器相关的方法。(这句话对否？)

对于 KNN 分类器，最理想的假设就是测试点在 k 个最近邻的条件下分类概率概率是不变的，这样就可以保证每次分类都能正确，但实际上不可能做到。我们可以放宽假设，取测试点在 K 近邻的条件下概率是光滑或者缓慢变化的函数，然而即使如此，该条件还是难以满足，毕竟在分类的决策边界附近，不同类的点的分类概率变化还是很大的，特别是在高维稀疏的情况下，这种分类的连续性难以满足。为了解决这个问题，我们将从两个角度来改进 KNN 分类器。一个就是高维情况时，降维或者赋予不同维度不同的权重，而且要保证同类点之间距离收缩，不同类点距离扩张；另一个就是增强决策边界的空间解析度 (spatial resolution)，通俗地讲就是使得决策边界处，点分类概率变化舒缓而不太剧烈，这里可以利用 SVM 的思想来思考。

最常见的是近邻分量分析 NCA(Neighborhood Component Analysis)，相关分量分析 RCA(Relevant Component Analysis)，大边界最近邻分类 LMNN(Large Margin Neareast Neighborhood Classification) 等方法。

2.2.1 NCA

现有如下数据集 $L = (\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)$。NCA 定义的邻域并不是像 KNN 那样先选择几个邻近点，而是通过概率 p_{ij} 的方式来定义点 $\mathbf{x}_i$ 的“软”邻域 (soft neighborhood)。

而为了保证矩阵 M 半正定，做如下分解 $\mathbf{M}=\mathbf{A}^T\mathbf{A}$，于是

$$p_{ij}=\frac{exp(-||\mathbf{A}\mathbf{x}_i-\mathbf{A}\mathbf{x}_j||^2)}{\sum_{k\neq i}exp(-||\mathbf{A}\mathbf{x}_i-\mathbf{A}\mathbf{x}_k||^2)}$$

令与 $\mathbf{x}_i$ 属于同类的样本组成集合为 $C_i=j|c_i=c_j$，则 $\mathbf{x}_i$ 被正确分类的概率可以表示 $p_i=\sum_{j\in C_i}p_{ij}$，那么总的分类正确率就可以表示为 $f(\mathbf{A})=\sum_{i=1}^{n}p_i$。那么我们的目标就是最大化 $f(\mathbf{A})$。这里对 A 求导整理可得

$$\frac{\partial f}{\partial \mathbf{A}}=2\mathbf{A}\sum_i(p_i\sum_k p_{ik}\mathbf{x}_{ik}\mathbf{x}_{ik}^T-\sum_{j\in C_i}p_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^T)$$

A 的求解可以通过采用基于梯度下降的最优化方法求得。另外，如果令 $\mathbf{A}$ 不是方阵，而是 $d\times m$，其中 $d\leq m$，NCA 也可以做线性降维 ($\mathbf{y}_n=\mathbf{A}\mathbf{x}_n$，变换后的 $\mathbf{y}_n$ 点维度变低，这对 KNN 分类器来说应该是件好事)。也就是说，我们可以对 A 的秩进行控制，在低维度下进行优化运算，而且这是无约束优化，接打起来比较简单。唯一的问题不能保证得到全局最优解。

当然 NCA 也有衍生品 MCML，基本定义是相同的，不同的是优化的目标。在概率论中，我们要衡量两个概率分布的非对称性多有大，通常用 KL 散度 (Kullback–Leibler divergence，简称 KLD)，也被称为相对熵，对于连续性的随机变量，计算公式如下：

$$KL(P||Q)=\int_{-\infty}^{\infty}p(x)ln\frac{p(x)}{q(x)}dx$$

因此对于 NCA，我们需要再引入新的概率分布来度量 (注意，KL 散度不具有对称性，通常 $KL(P||Q)\neq KL(Q||P)$)。引入“完美”概率分布 p_{ij}^0:

$$p_{ij}^0\propto\begin{cases}1 & 若\mathbf{x_i}与\mathbf{x_j}同类\\ 0 & 其他\end{cases}$$

我们需要优化的过程就变成了

$$min_{\mathbf{A}}\quad KL(p^0||p)$$

$$s.t.\quad \mathbf{A}\geq 0$$

之所以这样做，就是问题就成了凸优化问题，就有很多现成的方法去解决，而且该优化方法下全局最优解也可以得到 (NCA 无法保证，多得到局部最优解)。

2.2.2 RCA

RCA 可以说是所有方法中最为简单的方法，它的计算与样本协方差矩阵计算非常类似，先简述整个计算过程。

1. 训练样本中属于已知的某类的子集称为“团簇”(chunklet)，未知的构建 k 个“团簇”。
2. 计算 RCA 的“团簇”协差阵 (类似类内协差阵)$\hat{\mathbf{C}}=\frac{1}{n}\sum_j^k\sum_{i=1}^{n_j}(\mathbf{x}_{ji}-\mathbf{m}_j)(\mathbf{x}_{ji}-\mathbf{m}_j)^T$。其中 $\mathbf{m}_j$ 是团簇 j 的均值， n_j 是团簇 j 内点的个数，n 是总的数据点个数。
3. $\hat{\mathbf{C}}^{-1}$ 即可当做 Mahalanobis 距离使用，对数据点做如下新的变换 $\mathbf{x}_{new}=\hat{\mathbf{C}}^{-1/2}\mathbf{x}$。对变换后的数据点进行分类。

RCA 乍一看觉得简单的不可思议，与协方差矩阵很像，但是是否能这样用，还需要理论支持，事实上，结合信息最大化 (Information Maximization) 理论可以证明结果是正确的。令变换后的点集为 $\mathbf{Y}$,$I(\mathbf{X},\mathbf{Y})$ 为变换前后两种点集的互信息 (信息熵的内容)，k 为一个阈值， $\mathbf{Y}=f(\mathbf{X})$ 则优化的问题即

$$\max_{f\in F} \quad I(\mathbf{X},\mathbf{Y})$$

$$s.t. \quad \frac{1}{p}\sum_{j}^{k}\sum_{i=1}^{n_j}||\mathbf{y}_{ji}-\mathbf{m}_j^y||^2\le k$$

在经过一系列的变换，由于 $p_y=\frac{p_x}{|J(x)|}$，则

$$H(\mathbf{Y})=H(\mathbf{X})+<\log|J(x)|>$$

其中雅克比项为常数，如为线性变换 $\mathbf{Y}=\mathbf{AX}$，则互信息只依赖于变换项 $\mathbf{A}$, 而 Mahalanobis 距离矩阵 $\mathbf{B}=\mathbf{A}^T\mathbf{A}$，原优化问题就变为了

$$\max_{\mathbf{B}}|\mathbf{B}|$$

$$s.t. \quad \frac{1}{p}\sum_{j}^{k}\sum_{i=1}^{n_j}||\mathbf{x}_{ji}-\mathbf{m}_j^x||_{\mathbf{B}}^2\le k$$

通过拉格朗日数乘法可以得到解为 $\frac{k}{n}C^{-1}$，而常数项 $\frac{k}{n}$ 对线性变换没有影响，因此 Mahalanobis 距离矩阵 $\mathbf{B}=\mathbf{C}^{-1}$。

又由于 $\hat{\mathbf{C}}$ 可能是奇异的，因此可以引入正则因子 ϵI 使得 $\hat{C}$ 非奇异。令 $\mathbf{X}$ 为整个点集矩阵， $\mathbf{X}=[\mathbf{x}_{11},\mathbf{x}_{12},\ldots,\mathbf{x}_{1n_1},\mathbf{x}_{21},\ldots,\mathbf{x}_{2n_2},\ldots,\mathbf{x}_{jn_j}]$， $\mathbf{1}_j$ 是一个 n 维的向量，表示 j 团簇内的点对应的位置为 1，其余的为 0. 而 $\mathbf{I}_j$ 为 $n\times n$ 的以 $diag(i_j)$ 的对角矩阵。则

$$\hat{\mathbf{C}}=\frac{1}{n}\sum_{j}^{k}\sum_{i=1}^{n_j}(\mathbf{x}_{ji}-\frac{1}{n_j}(\mathbf{X}\mathbf{I}_j))(\mathbf{x}_{ji}-\frac{1}{n_j}(\mathbf{X}\mathbf{I}_j))^T=\frac{1}{n}\mathbf{X}\mathbf{H}\mathbf{X}$$

其中 $H=\sum_{j=1}^{k}(\mathbf{I}_j-\frac{1}{n_j}\mathbf{1}_j\mathbf{1}_j^T)$ 而更新的 $\hat{\mathbf{C}}$ 即为 $\mathbf{C}=\epsilon I+\hat{\mathbf{C}}$。这样的方式更新后，可以求得 $\mathbf{C}$ 的逆矩阵。(这样求的 $\mathbf{C}$ 后可以 Kernel 化)

另外，RCA 也可以做降维。假设数据为高斯型时，可以降维。

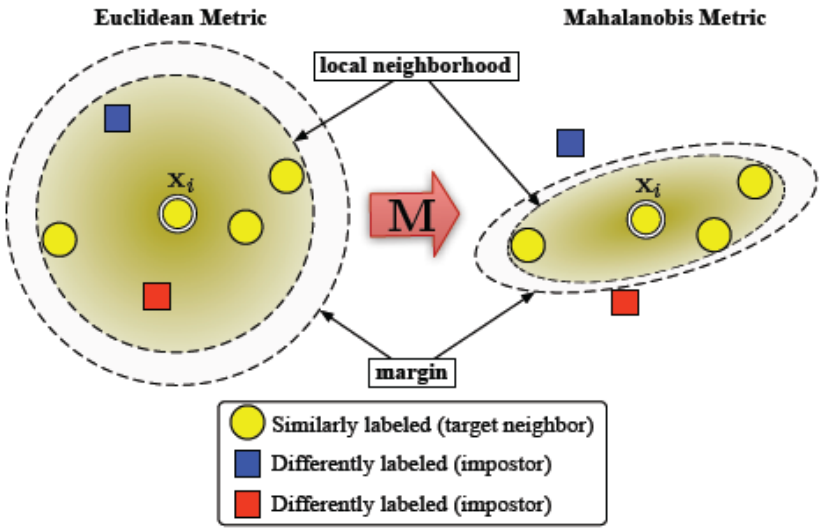
2.2.3 LMNN

LMNN 的思想类似 SVM。SVM 虽然被称作支持向量机，名字看起来很神秘，但是真正做的不过是寻求一个分类的超平面 (hyper plane classifier)，使得该超平面与最靠近的点 (分类有效，或者说正确)，它们之间的边界 (margin) 最大化。这种思想对解决 KNN 分类器的第二问题非常有好处，而 LMNN 的思维就是为了使得变换后的邻域内的点同类之间紧缩，不同类点扩展，并且之间间隔 (margin) 能尽可能大，这里有副图很好的揭示了 LMNN 的思路：

1. 缺乏先验信息时，我们先可以使用 Euclid 距离做 KNN，选择测试点 $\mathbf{x}_i$ 同类标签 y_i 的 k 个“目标”邻居 (target neighbors)，其余几个邻居不是同类的就是“非目标”邻居；

2. 对各点进行线性变换 L，计算各点之间的 Mahalanobis 距离 $d(\mathbf{x}_i,\mathbf{x}_j)=(\mathbf{x}_i-\mathbf{x}_j)^TM(\mathbf{x}_i-\mathbf{x}_j)$，其中 $\mathbf{M}=\mathbf{L}^T\mathbf{L}$；

3. 变换后，限制目标邻居与非目标邻居之间的 Mahalanobis 距离之差至少比一常量大，同时最小化目标邻居与测试点的 Mahalanobis 距离之和。



因此按上面的思路，即可以得到如下的优化问题：

$$\min_{\mathbf{M}} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S} d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)$$

$$s.t. d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k) - d_{\mathbf{M}}(\mathbf{x}_i-\mathbf{x}_j) \geq 1, \forall(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\in R, A\geq 0$$

和 SVM 对待离群点 (outliers) 一样的，为了能保证总有最优解，引入松弛变量 (slack variables)ξ_{ijk}，于是对原问题稍加改动即可：

$$\min_{\mathbf{M},\xi} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S} d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) + c \sum_{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\in R} \xi_{ijk}$$

$$s.t. d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k) - d_{\mathbf{M}}(\mathbf{x}_i-\mathbf{x}_j) \geq 1-\xi_{ijk}, \forall(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\in R, \mathbf{M}\geq 0, \xi_{ijk}\geq 0$$

对这类优化问题，也有很多现成的算法，类似于像 SVM 那样转化对偶问题后再用梯度下降解决。由于对非邻居点没有约束，算法的速度还挺快，据在双核的 2GHZ 的台式机上测试，32 亿个约束花了不到 4 个小时即可完成，而且效果还很不错，仅有的麻烦就是邻近点数量的选择，以及是否使用其他常量代替 1 来加减松弛变量 ξ_{ijk}，这些都需要一定的尝试才能确定好的效果。

另外，LMNN 也有衍生品——局部化，不再是只得到一个全局的距离测度 $\mathbf{M}$，而是得到多个距离测度 $\mathbf{M}_i$。方法是首先将训练集聚类划分为 k 个子集，然后对每个子集做 LMNN，这样就得到了各个子集的距离测度 $\mathbf{M}_i$，这样的好处是更关注数据的局部特征，使得 KNN 能更细腻准确的分类，当然这也加重了计算的负担，而且未变换前子集的划分对后续的计算有多大影响个人觉得还需要大量验证。

2.2.4 LFDA

FDA 可谓年代久远，于 1936 年提出，从名字看，应该是 Fisher 他天才老人家发明的。虽然方法很老，也有一些问题，比如大 p 小 n 问题处理不好[3]，但是却依然活力无限，

[3]但是我在查阅相关资料时，发现 Tibshirani 大神 2011 年出了个带惩罚的 FFFDA，可以解决大 p 小 n 问题。

稍微对其改进后，对于不同数据的分类效果也是相当不错。主要原因是 FDA 的分类思想有的放矢，非常明确，而且还能有降维效果，同时对多分类问题也是没有任何问题，算法也简单，效率也高，实在是线性分类器的一大利器。下面介绍两种基于 FDA 的方法。

(1) DANN

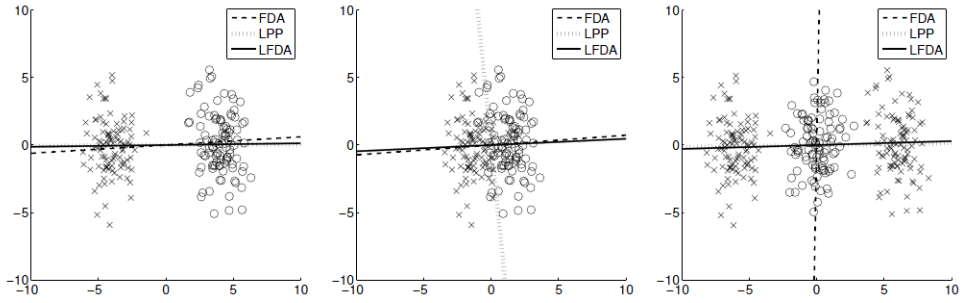
早在 1995 年时，Hastie 和 Tibshirani 就提出了“适应性的最近邻判别分类法”(Discriminant Adaptive Neareast Neighbor Classification，简称 DANN)，是局部距离的学习。既然是局部的适应性方法，思路和上面的 LMNN 差不太多。先局部化————对测试点选择 K 近邻个数 K_m，并对近邻点根据离测试点距离远近 (某种映射) 赋予不同权重 w_{ij}，然后根据权重计算这些近邻的类间离差阵 $\mathbf{B}$ 和类内离差阵 $\mathbf{W}$，然后以该公式更新矩阵 $\sum$ 直至收敛:

$$\sum = \mathbf{W}^{-1/2}[\mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2} + \epsilon\mathbf{I}]\mathbf{W}^{-1/2}$$

然后用收敛的 $\sum$ 重新度量近邻点距离 $(x - x_0)^T \sum (x - x_0)$，用 KNN 法对测试点进行分类。(至于为何用这样的形式更新矩阵，可以结合 FDA 的结果和病态矩阵两处思考。) 另外，该法对有 n 个点的训练集可以得到 n 个局部类间离差阵，经过适当转变可以用于降维，可以得到判决边界。具体过程此处不详述，比较繁琐，可以参考两位大神的论文。

(2) 另一种基于 FDA 的方法

首先看一幅图，看完这幅图基本就可以看到 LFDA 要解决的问题了。



其中 LPP 指的是局部保留投影 (Locality-Preserving Projection)。图 (a) 表示在两类数据分离时三种方法表现的都很好；图 (b) 表示当两类数据混为一堆时，FDA 和 LFDA 都能表现好，而 LPP 由于非监督的学习思维————忽略类别，最大程度的保留数据结构 (竖直方向降维显然信息量最大)；图 (c) 表示同类数据也分离，同时距离小于异类数据时，FDA 会出现异常，而 LPP 和 LFDA 表现正常，这是因为 FDA 目标就是在类内离差阵小的情况下，最大化类间离差阵，面对此种数据，目标出现矛盾，结果当然也容易异常。相反，LPP 能够较好的保留数据结构信息。因此我们可以看到，LFDA 思想就是要融合 FDA 的目标明确的有监督性，和 LPP 的保留数据结构的无监督性，从而对各种有较好线性可分的数据集 (若数据集非线性可分，需要用到非线性的分类方法，如 Kernel 方法) 做分类。虽然定性描述 LFDA 看起来还挺复杂，似乎要结合两个方法的精华，但是实际上 LFDA 只是 FDA 一个系数上的一丁点的改进罢了，整个过程非常简单。

首先，传统的 FDA 类内离差阵 $\mathbf{S}^{(w)}$ 和类间离差阵分别为 $\mathbf{S}^{(b)}$ 分别为

$$\mathbf{S}^{(w)} = \sum_{i=1}^{l} \sum_{j:y_j=i} (\mathbf{x}_j - \mathbf{u}_i)(\mathbf{x}_j - \mathbf{u}_i)^T, \mathbf{u}_i = \frac{1}{n_i} \sum_{j:y_j=i} \mathbf{x}_j$$

$$\mathbf{S}^{(b)} = \sum_{i=1}^{l} n_i(\mathbf{u}_i - \mathbf{u})(\mathbf{u}_i - \mathbf{u})^T, \mathbf{u} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

将上面的公式展开、重新整理即可得到如下的新的表达形式：

$$\mathbf{S}^{(w)} = \frac{1}{2}\sum_{i,j=1}^{n}\mathbf{A}_{ij}^{(w)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$$

$$\mathbf{S}^{(b)} = \frac{1}{2}\sum_{i,j=1}^{n}\mathbf{A}_{ij}^{(b)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$$

其中

$$\mathbf{A}_{ij}^{(w)} = \begin{cases} \frac{1}{n_c} & \text{如果} y_i = y_j = c; \\ 0 & \text{如果} y_i \neq y_j \end{cases}$$

$$\mathbf{A}_{ij}^{(b)} = \begin{cases} \frac{1}{n} - \frac{1}{n_c} & \text{如果} y_i = y_j = c; \\ \frac{1}{n} & \text{如果} y_i \neq y_j \end{cases}$$

以上过程都没有任何变化，只是正常的整理变形，下面将要结合局部的思想————LPP 中依据最初点之间的远近来赋予变换后的距离权重 (关于 LPP 详细介绍可参考相应论文)，因此引入一个“相似度矩阵”(affinity matrix)$\mathbf{A}$(元素取值属于 [0, 1])，其中的元素值越大，表明两点靠的越近。常用的 $\mathbf{A}$ 有二值型 0,1，表示最近邻点 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 对应的 $\mathbf{A}_{ij} = 1$，不是最近邻的点则 $\mathbf{A}_{ij} = 0$，还有高斯型 $\mathbf{A}_{ij} = exp(\frac{-d(\mathbf{x}_i,\mathbf{x}_j)}{\sigma^2})(i \neq j)$，而 $\mathbf{A}_{ii} = 0$。对上述的 $\mathbf{A}_{ij}^{(w)}$ 和 $\mathbf{A}_{ij}^{(b)}$ 进行改变，随之的 $\mathbf{S}^{(w)}$ 和 $\mathbf{S}^{(b)}$ 也进行更新：

$$\hat{\mathbf{A}}_{ij}^{(w)} = \begin{cases} \frac{\mathbf{A}_{ij}}{n_c} & \text{如果} y_i = y_j = c; \\ 0 & \text{如果} y_i \neq y_j \end{cases}$$

$$\hat{\mathbf{A}}_{ij}^{(b)} = \begin{cases} \mathbf{A}_{ij}(\frac{1}{n} - \frac{1}{n_c}) & \text{如果} y_i = y_j = c; \\ \frac{1}{n} & \text{如果} y_i \neq y_j \end{cases}$$

很明显，当类内的点如果都同等的足够靠近，那么 $\mathbf{A}_{ij}^{(w)}$ 和 $\mathbf{A}_{ij}^{(b)}$ 就没有改变。但是事实上，这种假设无法满足，因此这种改进关注了局部特征。更明确的说，相似度矩阵 A 的引入，导致了不管是类间还是类内中同类点的权重下降，这样正好符合图 (c) 的描述，LFDA 法能够吸收局部特征，忍受了“同类远离之苦”，从而不会导致分类器异常。后续的方法与 FDA 一致，通过求广义特征向量获得转换矩阵 $\mathbf{L}$，从而既得到分类边界，也得到了相应的距离测度 $\mathbf{M} = \mathbf{L}^T\mathbf{L}$。

另外，基于 FDA 的方法还有一个 DCA（Discriminative Component Analysis）。它与 FDA 整个过程非常类似，只是原来的类内和类间离差阵，变成了基于约束条件的类内和类间协方差阵。因此计算结果会有差别，算法也有差别。不过整体来说，核心思想其实就是 FDA。

2.2.5 ITML

ITML 全称为信息论测度学习 (Information-Theoretic Metric Learning)。既然被称为信息式学习，那么肯定牵涉到先验信息，同时也会有熵的概念出现。类似于 NCA 的衍生品，我们要求的目标就是一个初始化的矩阵 $\mathbf{A}_0$(比较理想) 与训练集所得的矩阵 A 之间的

KL 散度最小。同时我们还有相似点在 $\mathbf{A}$ 矩阵下的距离信息限制 u，以及非相似点在 $\mathbf{A}$ 矩阵下的距离信息限制 l。因此这个优化过程就顺理成章了：

$$\min_{\mathbf{A}} KL(p(x;\mathbf{A}_0||p(x;\mathbf{A}))$$

$$s.t. \begin{array}{ll} d_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_j) \le u & (\mathbf{x}_i,\mathbf{x}_j) \in S \\ d_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_j) \ge u & (\mathbf{x}_i,\mathbf{x}_j) \in D \end{array}$$

这样似乎就行了，但是计算还是不方便。刚好，这方法可以与一种叫做 Bregman divergence 方法进行转变：$D_{ld}(\mathbf{A},\mathbf{A}_0) = tr(\mathbf{A}\mathbf{A}_0^{-1}) - logDet(\mathbf{A}\mathbf{A}_0^{-1}) - d, \mathbf{A}, \mathbf{A}_0$均为$d \times d$的矩阵而

$$KL(p(x;\mathbf{A}_0||p(x;\mathbf{A}))) = \frac{1}{2} D_{ld}(\mathbf{A}_0^{-1},\mathbf{A}^{-1}) = \frac{1}{2} D_{ld}(\mathbf{A},\mathbf{A}_0)$$

同时为了能够保证一定有解，引入松弛变量 $\xi_{c(i,j)}$, 初始值 ξ_0 可以如下设置：当 $(i,j) \in S$ 时， $\xi_{c(i,j)} = u$；当 $(i,j) \in D$ 时， $\xi_{c(i,j)} = l$，整个优化问题就便成了如下形式：

$$\min_{\mathbf{A} \ge 0, \xi} \quad D_{ld}(\mathbf{A},\mathbf{A}_0) + cD_{ld}(diag(\xi), diag(\xi_0))$$

$$s.t. \begin{array}{ll} tr(A(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T) \le \xi_{c(i,j)} & (i,j) \in S \\ tr(A(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T) \ge \xi_{c(i,j)} & (i,j) \in D \end{array}$$

另外，更令人惊喜的是，由于不少人对 Bregman 散度 (Bregman divergence)[4]进行了研究，有良好的算好对此问题进行解决。一个迭代式再通过相应参数的判别循环，既可以得到最后的距离测度

$$\mathbf{A}_{t+1} = \mathbf{A}_t + \beta \mathbf{A}_t (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}_t$$

更令人惊奇的是该方法与 Low-Rank Kernel 学习 ($\mathbf{K} = \mathbf{G}^T\mathbf{G}, rank(\mathbf{G}) \le rank(\mathbf{K})$) 可以联系起来，上面这种形式是很容易 Kernel 化，随即就可以转入低秩 Kernel 学习问题，这个同样已有比较好的方法解决。更为重要的是，由于最初的优化问题所针对的约束本身就很宽泛、灵活，而且可以用迭代的方式求得最优距离测度，因此该方法可以提供在线学习（Online Metric Learning）。

OML 思想简单地说就是每一次用更新前的矩阵 $\mathbf{A}_t$ 去预测点距离 $\hat{d}_t$，然后评估“真实”（原来的距离）点距离 d_t 与更新前矩阵下点距离之差 $l_t(\mathbf{A}) = (d_t - \hat{d}_t)^2$，每一次都最小化这种差的和，得到最优的新的 A。这种思想类似于不断学习，不断进步的贝叶斯思想。根据以前的在线学习算法思想

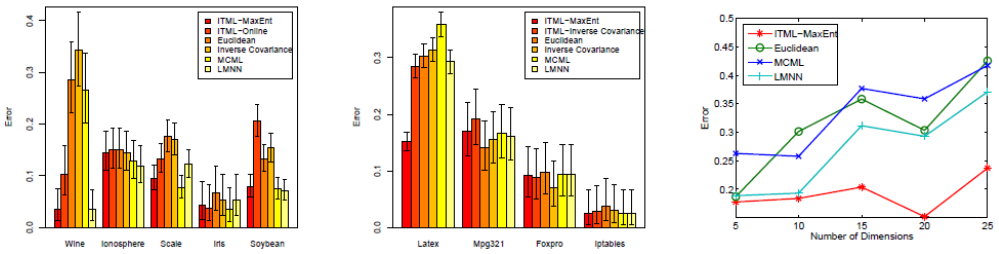
$$\min_{\mathbf{A} \ge 0} \quad f(\mathbf{A}) = \overbrace{D(\mathbf{A},\mathbf{A}_t)}^{\text{正则项}} + \overbrace{\delta_t l_t(\mathbf{A})}^{\text{损失项}}$$

再根据上面的算法，那么每次更新的矩阵就成为

$$\mathbf{A}_{t+1} = \text{argmin}_{\mathbf{A}} D_{ld}(\mathbf{A},\mathbf{A}_t) + \delta_t (d_t - \hat{d}_t)^2$$

但是从已有的测试结果来看，ITML 的 OML 的效果并不是非常好，比不上之前的 ITML。个人觉得这应该是由于“真实”距离的引入导致的，使得最后新的距离与理想距离尽可能符合的目标有所减弱。下面一幅效果比较图。

[4]Bregman 散度表示的意义就是一个函数 $\phi(x)$ 与该函数的线性近似值 $L(x)$ 间的差 $D(x,y) = \phi(x) - L(x), L(x) = \phi(y) + < \nabla\phi(y), (x-y) >$



(a) UCI Datasets (b) Clarify Datasets (c) Latex

3 无监督的距离测度学习

对于 K 均值聚类中，“簇”（即所分的类别）的最佳个数选择有多种标准可作参考，比如“簇”内距离平方和的拐点，平均轮廓系数最大点。但是这些方法对于相互缠绕的簇区分并不是那么好，因此仅能作为参考，实际“簇”个数选择还是要谨慎小心。

无监督距离测度学习也被称为流形[5]学习，思想精要即从高维流形中学习得到一个保留大部分观测点的几何关系的低维流形，通俗来说就是降维。噪音和信息损失是必然的，所以要尽可能保证大部分主要的观测点的信息保存下来，同时对于原来数据之间的几何关系要能够在低维流形上保留。比如传统的 PCA 方法，获得主成分向量 $\mathbf{u}_i$，构建距离测度 $\mathbf{A} = \sum_i \mathbf{u}_i \mathbf{u}_i^T$，即可得到点 $\mathbf{x}$ 与点 $\mathbf{y}$ 间的距离 $d = (\mathbf{x}-\mathbf{y})^T \mathbf{A} (\mathbf{x}-\mathbf{y})$。PCA 只是一种线性的变换，对于内部结构更复杂的空间无法处理，因此其他好的处理方式也不断涌现出来，比如 ISOMAP，能进行非线性的处理。

无监督距离测度学习与降维紧密相关，聚类也与降维紧密相连，而降维方法[6]已经比较普及，因此本文只是对各个降维方法进行概述，并不涉及具体的式子。

3.1 PCA

PCA 是非常传统的降维算法，目标是寻找一个正交基 $\mathbf{U}$，使得变换后的 $\mathbf{y}_i = \mathbf{U}^T \mathbf{x}_i$ 方差最大。方法就是对原来 $\mathbf{X}$ 的协方差阵求特征值和特征向量，取前 m 最大分量即可。后来的 kernel 法也不过是借助 kernel 的思想将这种线性的变换转变为非线性，思想类似。

3.2 MDS

MDS 是经典的降维方法。目标就是给出初始数据的相似度矩阵（距离矩阵）后，使得降维后的数据能够保持这种相似度。所用方法与 PCA 很相似，都用到特征根分解。

3.3 ISOMAP

ISOMAP 其实就是扩展了 MDS，使得能够处理非线性流形。Isomap 的主要目标是对于给定的高维流形，欲找到其对应的低维嵌入，使得高维流形上数据点间的近邻结构在低维嵌入中得以保持。它的不同之处在于不用欧氏距离表示高维流形点间距离，而是用曲线距离（也成为测地线距离）。步骤是先用欧式距离构建一个新的权重图，然后在这个图上寻找最短路径代替原来的距离，作为曲线距离的近似。虽然再根据 MDS 法对新的距离矩阵进行求解。

[5]所谓流形就是局部具有欧几里得空间性质的空间，比如三维空间的球面是二维流形

[6]可参考 http://blog.pluskid.org/?p=290

3.4 LLE

LLE(Locally Linear Embedding) 思想也很简单。降维后的点的局部性质与降维前的局部性质相同。先对第 i 点用它的 K 个邻近点的线性组合来重构表示，线性组合的系数为权重矩阵，然后求解这些重构误差和最小时的权重矩阵。求得局部线性重构权重矩阵后，对新的降维点用重构矩阵再进行重构，并求解新的误差最小。

3.5 Laplacian Eigenmaps

LE 与 LLE 很类似。LE 先需要一个相似度矩阵，然后构造一个降维后点与相似度矩阵的目标函数，最小化能使得原来相近的点在映射降维后也不会彼此之间相差太远。求解方法中涉及到 Laplace 矩阵和特征值求解问题，整体过程也比较简单。它的假设就是数据分布在一个高维空间的低维流形，然后寻找这个流形，此处 Laplace 矩阵是该流形的离散近似。上面的 LLE 方法同样假定了平滑流形的局部线性性质。

4 半监督的距离测度学习

TODO

参考文献

[1] Bar-Hillel, A., Ertz, T. H, Shental, N., & Weinshall D.. Learning distance functions using equivalence relations. in *Proc. ICML*, 2003.

[2] D. M. Witten, R. Tibshirani. Penalized classification using Fisher’s linear discriminant. *J. R. Statist. Soc. B* (2011), 73, Part 5, pp. 753–772.

[3] J. Goldberger, S. Roweis, G. Hinton, & R. Salakhutdinov. Neighbourhood components analysis. in *Proc. NIPS*, 2005.

[4] J. V. Davis, B. Kulis, P. Jain, S. Sra, & I. S. Dhillon. Information-Theoretic Metric Learning. in *Proc. ICML*, 2007.

[5] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. in *Proc. NIPS*, 2006.

[6] Sugiyama, M.. Local Fisher discriminant analysis for supervised dimensionality reduction. in *Proc. ICML*, 2006.

[7] T. Hastie, R. Tibshirani. Discriminant adaptive nearest neighbor classification. in *Proc. KDD*, 2003.

[8] Xing, E., Ng, A., Jordan, M., & Russell, S.. Distance metric learning with application to clustering with side-information. *NIPS*, 2003.

[9] Yang, L., & Jin, R.(2006). Distance metric learning: A comprehensive survey. http://www.cs.cmu.edu/~liuy/frame_survey_v2.pdf