Local Affine Multidimensional Projection

蔡睿凡_11421030

1 引言

多维投影技术是可视化领域的基础方法之一。当数据维度大于视觉空间维度时，如何查看数据之间的关系成为一个非常重要的问题。多维投影技术基于数据点之间的相似度，将数据从高位空间变换至视觉空间（2 维或 3 维），并保持在视觉空间中数据投影点之间的距离尽可能与原始高维空间中数据点之间的距离相似。这样在低维可视空间中，使用者可以根据数据在低维空间中投影之间的关系来间接理解数据在原始空间中的分布和相似性关系。

然而，多维投影技术通常存在以下一些问题：性能当前多维投影算法通常计算复杂度较高（大于线性），这使得方法在数据量较大的情况下变得效率低下，甚至无法得出结果； 用户知识输入可视化的重要目的之一是将人类交互融合入可视化过程中。绝大部分多维投影技术通常只构建一个高位数据空间到低维视觉空间的全局 映射，这种方式使得用户难以对映射过程进行调整，或是根据先验知识改 善投影效果。近年来一些局部多维投影方法开始重视用户输入的作用，这些方法通常会定义一些局部锚点，这些锚点由用户指定，其他数据则根据 锚点位置和数据点相似性进行投影。这种方法的好处是允许用户的先验知 识以锚点方式输入算法过程中，并且拥有相对较低的复杂度，其中一些高 效率的局部投影算法已经达到了线性时间复杂度。当前局部投影算法的问 题是锚点的设置对整个投影结果的敏感度较大，当锚点进行简单移动时， 投影结果也会有相当大的差异。同时，控制点的数量也对最终投影结果有 巨大影响。当控制点较多时，最终投影结果会更加实际地反应数据点之间 的相似度关系，然而控制点的获得会消耗用户分析成本，控制点过多反而 会得不偿失。

基于上述挑战，[1] 提出了局部仿射多维投影方法（Local Affine Multidimensional Projection, 缩写为 LAMP）。该方法属于局部投影方法，其具备了局部方法所具有的用户输入支持和较低的算法复杂度，同时在锚点进行小幅度变化时，结果的变化敏感度较低。

2 方法概述

图 1 展示了局部仿射多维投影方法的三个主要步骤。当数据输入后，首先数据中的控制点将以全局多维投影方式先行投影，之后适用于其他数据点的仿射变 换将基于控制点投影结果计算得出，其他数据点将使用此变换进行投影，用户可 以根据投影结果进一步调整控制点位置和相互距离关系，新的仿射变换也会被重 新计算，整个过程构成了一个有用户参与的迭代过程。

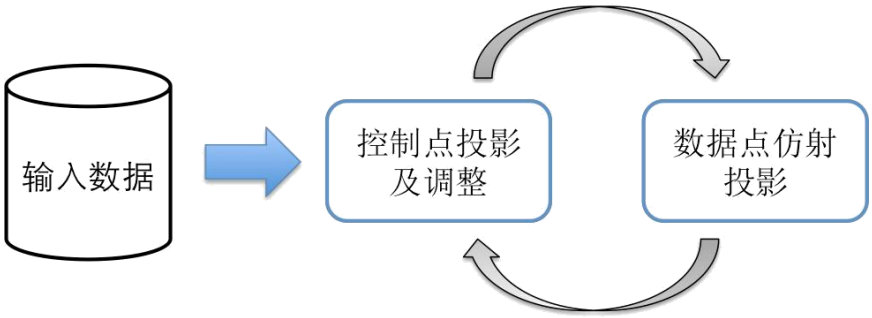

Figure 1: 算法过程

2.1 定义

为了描述算法过程，我们使用以下定义：

x 为数据集 X 中的数据点，个数为 n，$x \in R^m$；

x_i 为从√X 中选择出的控制点集 $X_S = \{x_1, x_2, \ldots, x_k\}$ 中的数据点，其个数一般为 n；

X_S 在视觉空间（R^2）中的映射记为 $Y_S = \{y_1, y_2, \ldots, y_k\}$。

首先，所有控制点使用一种全局多维投影方法先行投影至 R^2 平面上，即先 行求出 Y。对于数据集中的其他数据点 x，局部仿射多维投影方法通过寻找一个 最优放射变换 $f_x(p) = pM + t$ 将其变换为 R^2 空间，该方式变换使得下式最小化：

$$\sum \alpha_i \| f_x(x_i) \|^2, M^T M = I$$

其中矩阵 M 和向量 t 为未知量， I 为单位矩阵， α_i 为权重，定义为：

$$\alpha_i = \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{x}\|^2}$$

对 t 求偏导数并赋值为 0，则 t 可记为：

$$t = \tilde{y} - \tilde{x}M, \tilde{x} = \frac{\sum_i \alpha_i \boldsymbol{x}_i}{\alpha}, \tilde{y} = \frac{\sum_i \alpha_i \boldsymbol{y}_i}{\alpha}$$

其中 $\alpha = \sum_i \alpha_i$。

这样，针对放射变换的最小化问题可重写为：

$$\sum_i \alpha_i \|\hat{\boldsymbol{x}}_i M - \hat{\boldsymbol{y}}_i\|^2, M^T M = I$$

其中 $\hat{x}_i = x_i - \hat{x}$，$\hat{y}_i = y_i - \hat{x}$ 上述最小化问题可被重写为以下形式：

$\|AM - B\|_F, M^T M = I$ 其中， $\| \cdot \|_F$ 为 F 范数，矩阵 A 和 B 为：

$$A = [\sqrt{\alpha_1}x\hat{}_1, \sqrt{\alpha}\bar{}_2 x\hat{}_2, \ldots, \sqrt{\alpha_k}x\hat{}_k]^T$$

$$B = [\sqrt{\alpha_1}y\hat{}_1, \sqrt{\alpha}_2 y\hat{}_2, \ldots, \sqrt{\alpha_k}y\hat{}_k]^T$$

重写后的最小化问题符合正交 Procrustes 问题的形式，其解为：

$$M = UV, A^T B = UDV$$

其中 UDV 是 $A^T B$ 的 SVD 分解结果。当 M 被计算出来后，数据点 x 的投影 y 可藉由以下变换得到：

$$y = f_x(x) = (x - x\hat{})M + y\hat{}$$

需要注意的是，由于投影目标空间为 R^2，因此 $A^T B$ 实际为 $m \times 2$ 矩阵，计算这种矩阵的 SVD 分解仅需线性时间 $O(k)$。这样整个算法的时间复杂度为 $O(kn)$。

3 实验结果

图 2 展示了 9 种多维投影方法（行方向）在 8 个数据集（列方向）上的测试结果矩阵。其中每一个矩阵单元为一个散点图，横纵坐标分别为原始空间中每对数据点之间的距离，和投影至 R^2 后两点之间的距离。散点越靠近角平分线，说明投影后两点之间的距离更能反映实际高维空间中两者的距离。从第一列 LAMP 方法的结果中可以看出，其大部分结果都处于角平分线附近，效果在同数据集其他方法上都占有优势。

4 小结与讨论

局部仿射多维投影方法作为一种局部多维投影方法，通常会受制于控制点的选择。由于这里的控制点依旧是采用全局投影方法先行投影至平面上，因此该全局方法的性能和复杂度也是影响本方法结果的一个重要因素。

如果控制点位置由用户进行手工设定，那么这种手工设定也会为算法本身引入不确定性。这就需要一个能够很好地配合算法本身的交互界面和完整的分析策略，使得用户能了解当前的控制点情况和投影结果好坏，并不断地优化控制点位置，以达到更好的投影效果。这是本算法在应用方面的一个很好的拓展。

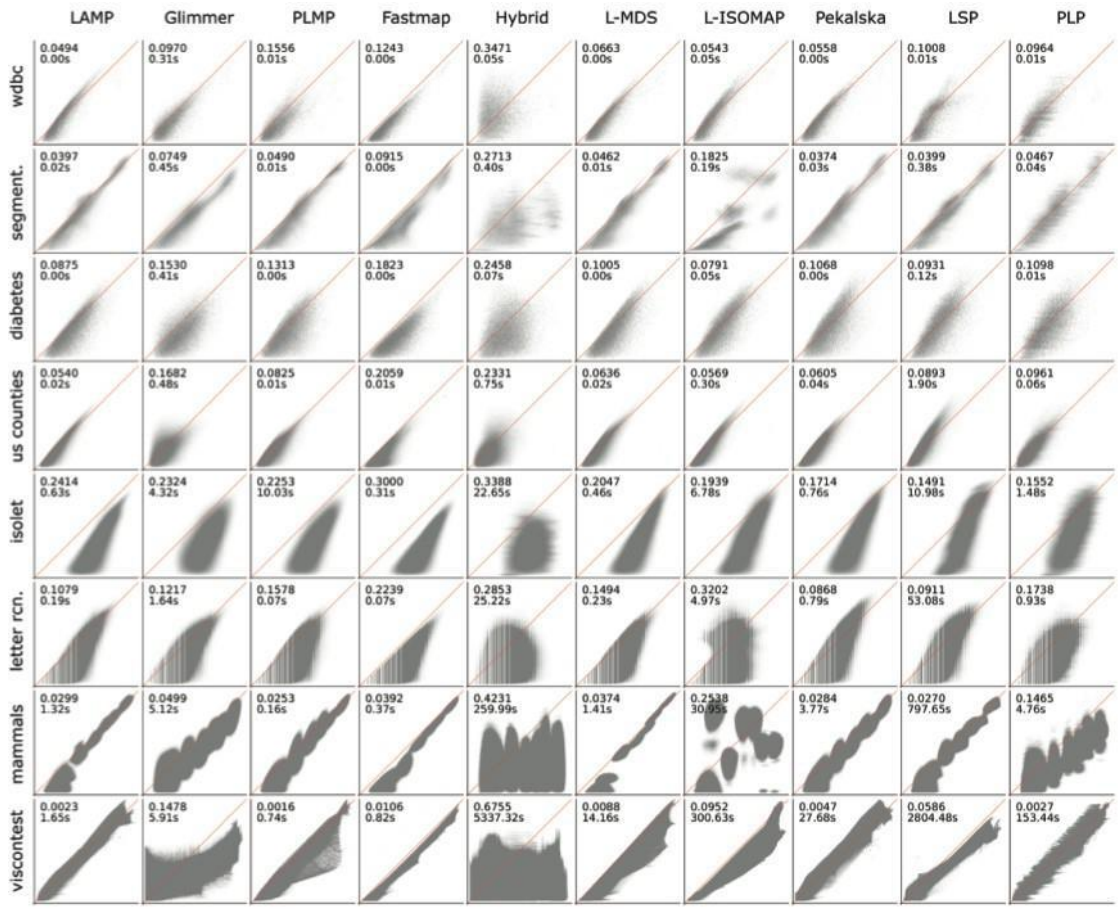

Figure 2:实验结果

参考文献

[1] P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato, and L. G. Nonato, “Local Affine Multidimensional Projection,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2563–2571, 2011.