使用t-SNE进行高维数据可视化

陈天楚(0112922)

May 31, 2016

1 引言

高维数据的可视化是在很多领域中都是一个比较重要的问题。在最近几十年内，一些不同的可视化高维数据不断地被提出来。其中，部分较为有影响力技术类型包括基于图标形式的类型（例如Chenoff face——一种把多维数据用二维的人脸表示出来的方式）、基于像素点形式的类型、把高维数据通过图的顶点的形式表示出来的类型等等。大多数这种技术类型都是简单地提供一个能够显示多维数据的工具并将发现数据中的规律的工作留给人类观察者。这类技术在处理上千维的真实世界中的高维数据时严重地受到限制。

和上文所述的数据可视化方案相比，基于降维的方法可以将高维度的数据转换成二维和三维的数据，这类数据可以用散点图表示。其中，降维的目标是在将高维数据映射到低维度的数据时，尽可能地保存高维数据中明显的架构图。为了解决这个问题提出了很多降维的方法。传统的线性降维方法如PCA 等的目标是让不相似的数据点在低维度的表达下距离较远。与之相对的是，在处理一些分布在低维度附近的高维数据点时，一些非线性的降维方法更倾向于使得一些相似的数据点在低维度的表达下能有较近的距离，这通常是线性降维方法难以做到的。

有许多非线性的降维方法以保留数据的局部结构为目标被提出。尽管有些方法在人工合成的测试数据上有优秀的效果，但这些方法经常在可视化高维的真实数据时不是那么成功。实际上，大多数这样的方法没有能力在一个映射中同时保留数据局部和全局的结构。

论文介绍了一种将高维数据转换成一个相似度矩阵的方法，在此基础上提出了一种新的技术“t-SNE”用于可视化所产生的相似度矩阵。t-SNE 有能力尽可能捕捉到高维数据的局部特征，同时能够反映数据的全局特征（例如：簇的存在）。论文比较了t-SNE和其他集中降维方法在物种数据集上的表现。

作为计算机应用数学的作业，本文主要内容包括介绍SNE作为t-SNE方法的基础、介绍t-SNE和SNE的区别及t-SNE的具体实现方式。除此之外，本文通过Python实现了一个用于演示t-SNE方法的工具，生成了一些高维数据并测试其结果。

2 SNE方法简介

SNE首先将高维数据点之间的欧式距离转换成用于代表相似度的条件概率[2]。数据点x_j到x_i 的相似度为条件概率$p_{j|i}$。$p_{j|i}$指的是如果按照以x_i为中心的高斯分布选择相邻的点，x_i 和x_j 相邻的概率。对于距离比较近的点对，$p_{j|i}$比较高。$p_{j|i}$通过下列方式计算得到：

$$p_{j|i} = \frac{exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} exp(-\left\|x_i - x_k\right\|^2 / 2\sigma_i^2)}$$

σ_i是高斯分布的方差（在下文会解释如何计算）。由于我们只考虑不同点之间的相似度，所以我们设置$p_{i|i}=0$。对于x_i、x_j对应的低维点y_i、y_j，可以用类似的方法计算$q_{j|i}$，由于可以把高斯分布的方差设置为一个固定的值（设置为不同的值只会导致结果的尺度不同），$q_{j|i}$可以写成下列形式：

$$q_{j|i}=\frac{exp(-\left\|y_i-y_j\right\|^2)}{\sum_{k\neq i}exp(-\left\|y_i-y_k\right\|^2)}$$

$$q_{i|i}=0$$

理想的情况下，如果y_i和y_j可以正确表达高维的x_i和x_j的相似度，那么$p_{j|i}$和$q_{j|i}$应该相等。基于这个想法，SNE的目标就是寻找一些低维点使得$p_{j|i}$和$q_{j|i}$间的差距尽可能的小。SNE以减少所有点的K-L距离总和为目标，使用梯度下降法达到这一目的，目标函数C 的定义为：

$$C=\sum_i KL(P_i\|Q_i)=\sum_i\sum_j p_{j|i}log\frac{p_{j|i}}{q_{j|i}}$$

由于K-L距离具有不对称性，不同类型的差距没有被平等地考虑。实际上SNE的目标函数更加重视保留数据的局部特征。

为了决定所有数据点的σ_i，SNE使用一个二分查找的方法，为每个点x_i找到一个能够产生与用户输入的困惑度最为相似的σ_i。困惑度的计算方式如下：

$$Perp(P_i)=2^{H(P_i)}=-\sum_j p_{j|i}log_2 p_{j|i}$$

用户输入的困惑度可以认为是一个控制邻居点的数量的平滑参数。

尽管SNE能够做出看上去合理的数据可视化，但SNE仍然存在目标函数不容易被优化的问题，以及被称作“Crowding Problem”的问题。

在一些情况下，两个高维点之间的距离不能被一个低维的映射所正确表达。例如当数据以一个点为中心均匀地分布在高维空间中的一个“球体”时，试图通过点之间的距离将其映射到二维空间会出现所谓的“Crowding Problem”问题：二维映射中用于表示相互远离的点的面积和用于表达相互靠近的点的面积相比显得不够大，因此如果想要在映射中更准确地表示相互靠近的点，距离适中的点对就必须在二维映射中变得十分远。

3 t-SNE方法概述

为了解决SNE中存在的“Crowding Problem”。t-SNE使用了和SNE不同的函数作为梯度下降法的优化目标。t-SNE的目标函数和SNE有两个区别[1]：

1、t-SNE使用对称的联合概率p_{ij}而不是条件概率$p_{j|i}$表示相似度，使得目标函数更容易被优化。

2、t-SNE在定义两个低维点之间的相似度时使用Student-t分布而不是高斯分布。t-SNE通过在低维空间中使用一个“长尾”分布做到同时解决“Crowding Problem”和SNE原本的优化问题。

根据第一点区别，在t-SNE中，高维空间中两个点的相似度由联合概率p_{ij}表示，很容易写出下列定义：

$$p_{ij}=\frac{exp(-\left\|x_i-x_j\right\|^2/2\sigma_i^2)}{\sum_{k\neq l}exp(-\left\|x_k-x_l\right\|^2/2\sigma_i^2)}$$

但是这样的定义当一个高维点x_i离其它所有点的距离$\|x_i - x_j\|^2$ 都很大时，对于所有的j，p_{ij}的值都很小，因此对应的低维空间中的y_i对目标函数的影响很小。为了避免这个问题，t-SNE 定义p_{ij}为：

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

这保证了对于所有的x_i能够对目标函数有一个明显的影响。

除此之外，为了解决“Crowding Problem”问题，t-SNE在定义两个低维点之间的相似度时使用Student-t 分布：

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}$$

基于和上一节相同的理由，定义$p_{ii} = 0, q_{ii} = 0$，t-SNE需要通过梯度下降法优化的目标函数为：

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

可以通过目标函数推导出对y_i的对应的梯度：

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}$$

综上所述，t-SNE的工作方式如下：

输入参数：高维数据：$x_1, x_2 ... x_n$；

困惑度：$Perp$；

进行梯度下降法时需要的参数：最大迭代次数T，步长η，动量$\alpha(t)$。

输出结果：低维空间对应的数据点：$Y^T = y_1, y_1 ... y_n$

算法步骤：

1、使用上一节所述的和SNE相同的方法，利用$Perp$计算条件概率$p_{j|i}$，令$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$。

2、从正态分布$N(0, 10^{-4}I)$生成初始的Y^0。

3、计算q_{ij}.

4、计算梯度$\frac{\delta C}{\delta Y}$。

5、$Y^t = Y^{t-1} + \eta \frac{\delta C}{\delta Y} + \alpha(t)(Y^{t-1} - Y^{t-2})$。

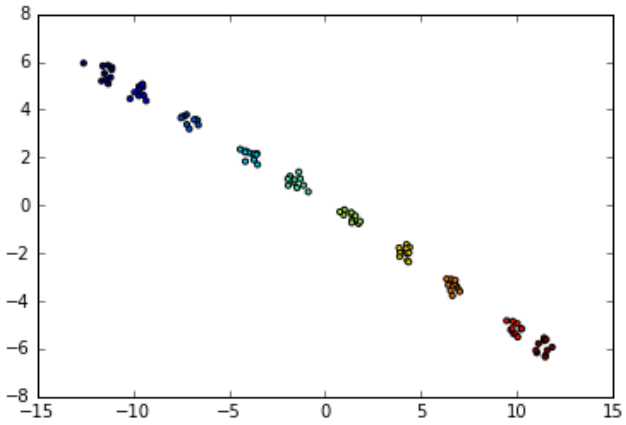
（重复步骤3到5一共T次）

4 实验结果

为了进行使用t-SNE处理高维数据的实验，本文选择Python语言作为程序设计语言实现了从生成高维测试数据到使用t-SNE降维并使用matplotlib完成二维散点图的绘制的过程。实验代码的编写过程中参考了论文作者提供的t-SNE实现代码。

实验代码生成了10个数据集$D_0, D_1 ... D_9$，每个数据集包含10个10维的数据。其中数据集D_i 通过平均值为i、协方差矩阵为$I * 0.1$的多变量高斯分布随机生成。

生成测试数据之后，实验代码尝试用t-SNE将其降至二维，并将二维的结果在散点图中绘制，散点图中各个数据点的颜色和数据点原本属于的数据集相对应。实验结果如下图所示。



在高维中属于不同高斯分布的数据点经过t-SNE降维后，可以在散点图中看到属于相同分布的数据聚集在一起，而属于不同分布的数据彼此相互远离。

5 小结与讨论

本文简要地介绍了SNE方法作为一种高维数据可视化方案的基本原理和优化目标，并分析了SNE方法存在的缺陷。在此基础之上，本文介绍了t-SNE方法和SNE方法的差异并描述了简单的t-SNE降维过程。作为一种高维数据可视化方案，t-SNE通过在高维数据的低维映射中表现数据点之间的相似度信息，实现了在可视化高维数据时能够同时保留数据中的局部结构和重要的全局结构。

本文介绍了一个基本的t-SNE的实现方法。这个方法在迭代时使用了固定的动量和步长。在实际使用t-SNE 时通常会通过对基本的方法进行改进以在较短的时间内获得更好的结果。这些改进包括随着迭代次数的增长修改动量和步长等参数，在使用t-SNE处理数据前对输入数据进行PCA降维等等。

t-SNE存在一些问题，例如目前仍不明确t-SNE 是如何进行降维工作的（因此论文将其作为一种数据可视化方案而不是降维方案提出），以及t-SNE中的优化方法不能保证收敛于全局最优等等。尽管如此，和其它数据可视化技术相比，t-SNE在结果上存在一些优势。作为一种数据可视化技术，t-SNE在处理高维真实数据时有尝试的价值。

参考文献

[1] Visualizing data using t-SNE, Laurens van der Maaten and Geoffrey Hintton, Journel of machine learning research, 2008.

[2] G.E. Hinton and S.T. Roweis. Stochastic Neighbor Embedding. In Advances in Neural Information Processing Systems, volume 15, pages 833 – 840, Cambridge, MA, USA, 2002. The MIT Press.