

一、2018-2019

1. 论述什么是多媒体？多媒体分析与理解有哪些应用领域和挑战？（10 分）

答：

多媒体：多媒体是使用不同内容形式的组合的内容，如文本、音频、图像、动画、视频和交互式内容。

应用领域：应用广泛，数据量大，就数字图像视频而言，就广泛应用在安防、教育、通讯、娱乐等产业中，多媒体还应用在图片检索、内容推荐、移动通讯、社交媒体平台、视频网站等。

面对挑战：

- (1) 如何表示多媒体数据（海量、高维、非结构化）
- (2) 如何理解多媒体数据（语义鸿沟）
- (3) 如何挖掘多媒体数据之间的相互关联（关联复杂，关联的协同性与互补性）
- (4) 如何满足用户多样化的信息需求（海量数据、用户偏好与个性化）

2. 针对文本、音频与图像数据的特征表示方法，请分别列举 2 种典型特征，并分析其优缺点。（15 分）

答：

文本

1) 词频 (TF) 表示法

优点：用词在文档中的出现频率可表示出一篇文档中的侧重点，便于统计与分析。

缺点：文中多次出现的介词、系动词等也会被统计出较高的权重。

2) 潜在语义分析法 (LSA)

优点：通过降维，有效解决了一词多义与一义多词的问题。

缺点：仍未解决文档中间的有序性被破坏的问题。

音频

1) 过零率

优点：能侧面反映信号在短时帧内的平均频率。

缺点：只关注了短时窗内的幅值信息，频域信息缺失。

2) 梅尔频率倒谱系数 (MFCC)

优点：对特征进行了去相关和压缩。

缺点：对所有频率段信息进行同等处理，不能突出重要信息。

图像

1) LBP

优点：一定程度上消除了光照变化的问题，具有旋转不变性，计算速度快。

缺点：光照不均时对应的 LBP 算子会发生变化，同时 LBP 也丢失了方向信息。

2) SIFT

优点：具有很好的尺度不变性和鲁棒性。

缺点：实时性不高，具有时特征点较少，对边缘光滑的目标无法准确提取特征点。

3. 卷积神经网络中一个典型层通常包含三种基本操作，请回答每种操作的基本含义或类型，所具备的基本特性或者优缺点。（15 分）

答：

典型层三种基本操作：卷积 — 非线性变换 — 池化

卷积操作：

1) 基本含义或类型：卷积操作又称滤波，卷积核函数又称滤波器。在输入图像上的二维卷积，卷积输出称为 feature map。

2) 基本特征或优缺点：通常会在同一个卷积层使用多个不同的卷积核以学习图像的不同特征。当卷积核输入包含多个通道时，可以把卷积核看成是 3D 的。

非线性变换：

1) 基本含义或类型：先通过 $\Phi(x)$ 将 x 空间的点转换成 z 空间的点，而在 z 空间上得到一个线性的假设，再恢复到原来的 x 空间中得到一个二次的假设。

2) 基本特征或优缺点：优点是有很好的数学性质，缺点是很容易饱和、输出不是 0 均值的会对梯度产生影响。

池化操作：

1) 基本含义或类型：池化函数使用某一位置的相邻位置的总体统计特征来代替网络在该位置的输出。

常用的池化函数有最大池化、平均池化。

2) 基本特征或优缺点：当输出作出少量平移时，池化能够帮助输入的表示近似不变，即平移不变。减小参数规模，提高统计效率。在全连接层之前采用全局池化，可以保持全连接层节点数量不变，不受输入图像大小的影响。

4. 请说明图像语义理解的基本研究内容，及其各项内容的含义。请针对其中任一项内容，从中选取 1 个典型算法或模型对其具体实现过程进行详细说明。（15 分）

答：

1) 图像语义理解①研究图像中有什么目标，目标之间的相互关系，图像是什么场景的一门学科。②利用人工智能技术让机器能像人一样自动“看懂”外部环境，并可帮助其更好地“说、动、想”。③本质上是学习图像底层特征与高层语义的映射关系。主要研究图像分类、图像标注、图像目标检测、图像语义分割、图像语义描述、视觉问答等内容。

2)

图像分类：给每幅图像一个类别标签（根据各自在图像信息中所反映的不同特征，把不同类别的目标区分开来的图像处理方法。它利用计算机对图像进行定量分析，把图像或图像中的每个像元或区域划归为若干个类别中的某一种，以代替人的视觉判读。）

图像标注：给图像多个语义标签

图像目标检测：对图像中的目标标注一个类别同时用一个紧致框定位目标（输出一系列带有类别标签的目标框）

图像语义分割：给图像多个语义标签且像素级定位

图像语义描述：用自然语言来描述图像内容，即看图说话。

视觉问答：给定图片和图片相关的问题，回答出正确的答案。

3) 基于深度学习的目标检测中的 R-CNN 模型：

1. 首先输入测试图像；

2 利用 selective search 算法在图像中提取 2000 个左右的目标候选区域；

3. 将每个候选区域缩放成 227×227 的大小输入到 CNN，采用 fc7 层特征输出；

4. 基于 CNN 的候选区域特征表示，输入到 SVM 进行分类。

5. 详细说明基于 SVD 和 RBM 推荐方法的基本原理，并比较它们的优劣。（15 分）

答：

SVD 推荐方法的基本原理：对于所有用户对所有商品打分，可以建一个矩阵来表示，而这个矩阵是极其稀疏的。而对于任意矩阵都有它的满秩分解。

$R_{U \times I} = P_{U \times K} \cdot Q_{K \times I}$ （ U 为用户数， I 为商品数）

然后用 R 中已知数据去训练 P 和 Q ，使得 P 和 Q 相乘的结果最好的拟合已知的评分。

预测用户 U 对商品的评分：

$$\hat{r}_{ui} = p_u^T q_i$$

假设已知的评分 r ，则预测误差为

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

继而可以计算出总的误差平方和：

$$SSE = \sum_{ui} e_{ui}^2 = \sum_{ui} \left(r_{ui} - \sum_{k=1}^K p_{uk} q_{ki} \right)^2$$

接下来训练的是 SSE 最小的模型即可。

RBM 推荐方法的基本原理：将某一用户对某一商品的评分视作一个 softmax 神经元，softmax 神经元是个长度为 k 的向量，只有一个分量为 1，其余分量为 0。而未评分的部分就可以用全 0 的 softmax 神经元表示。这样某个用户的评分可以用矩阵 V 来表示，在给定可见单元的状态下其激活概率为：

$$P(h_j = 1|V) = \frac{1}{1 + \exp\left(-bj - \sum_{i=1}^M \sum_{k=1}^K V_i^K W_{ij}^K\right)}$$

同理，在给定隐单元状态下，可见单元的激活率为：

$$P(V_i^K = 1|h) = \frac{\exp\left(a_i^K + \sum_{j=1}^F w_{ij}^K h_j\right)}{\sum_{l=1}^K \exp\left(a_i^l + \sum_{j=1}^F w_{ij}^l h_j\right)}$$

算每一个评分 K 所对应的激活概率，取所有概率的期望作为预测值。

比较：SVD 计算过程虽更简单，但由于训练目标单一，容易造成过拟合，而 RBM 则可以防止梯度爆炸和梯度消失，但求期望的过程会比较复杂，学习效率过慢。

6. 简单描述迭代量化哈希方法 (Iterative Quantization, ITQ) 的基本思想, 并比较 ITQ 方法与局部敏感哈希 (LSH) 方法的优劣。（15 分）

答：

迭代量化哈希方法的基本思想：先对数据集进行 PCA 降维，然后寻找量化误差最小的旋转矩阵即可得到对应该最优旋转矩阵下的特征向量的二进制编码。（查找 PCA 嵌入式数据；对于 c_bit 代码，采用顶部 c PCA 方向和零阈值）

比较 ITQ 方法与局部敏感哈希 (LSH) 方法的优劣：

1. ITQ

优点：相比局部敏感哈希 (LSH) 方法多了一步操作，即数据降维后使用矩阵旋转优化，可以降低量化误差

劣势：由于 PCA 不同维度的方差不平衡，旋转 PCA 投影数据以尽量减少量化误差时需不断控制旋转角度，即找出最优的旋转矩阵和与之对应的编码，相对麻烦

2. LSH

优点：通过 hash function 映射变换操作，将原始数据集分成了多个子集合，而每个子集合中的数据间是相邻的且该子集合中的元素个数较小，因此将一个在超大集合内查找相邻元素的问题转化为了在一个很小的集合内查找相邻元素的问题，计算量下降了很多，改善近似检索的计算表现。

劣势：局部敏感哈希 (LSH) 方法并不能保证一定能够查找到与 query data point 最相邻的数据

7. 运动目标检测的困难有哪些？并简述目前常用方法的优缺点。（15 分）

答：

困难：光照的变化、动态背景、伪装、相机抖动、相机自动对焦、间断的物体运动、阴影

目前常用方法：帧间差分法、背景差分法、统计平均、高斯模型、非参数核密度概率估计、基于码本的运动目标检测、背景建模 (ViBe、SuBSense)

优缺点：

1. 帧间差分法

优点：算法简单、容易实现、检测速度较快。而且一般相邻两帧的时间间隔比较短，因此对场景的光线变化不是特别敏感。

缺点：对噪声非常敏感且检测出的物体位置不精确，其次帧差法的检测结果与目标运动速度和相邻两帧间隔大小有关。目标运动过快会被分成两个目标；目标运动过慢会被视作背景。

2. 背景差分法

优点：算法比较简单；一定程度上克服了环境光线的影响；

缺点：不能用于运动的摄像头；对背景图像实时更新困难；

3. 统计平均法

优点：选择适当的参数能够很好地修正背景图像，从而得到一个较为真实的背景估计图像。

缺点：对于频繁出现的运动目标或者在场景内停留的时间较长，该模型不能够很好的提取运动目标。在复杂场景中会将一些伪目标（如摇摆的树枝等）检测为运动目标。这是由于在场景中树枝的摇摆会引起像素值的变化。

4. 混合高斯模型

优点：能适应背景随时间的缓慢变化，能描述背景中的如显示器屏幕闪烁，树枝晃动等一些周期性扰动等。

缺点：对于缓慢运动的目标不能精确的检测提取，不能够很好的适应阴影、噪声造成的误检或漏检，且不能适应场景的突然变化。

5. 非参数核密度概率估计

优点：可以渐进收敛于任意形状的概率密度，而且对动态的场景还具有一定的适应性。

缺点：运算量非常大，很难实现对视频图像的实时检测。对内存要求比较高

6. 基于码本的运动目标检测

优点：鲁棒性强，计算效率高，速度快，运算量少，精确度高。

缺点：当训练帧存在较大运动前景时，码本模型的建立会非常不准确，更新参数的调整非常复杂，不能广泛应用于实际情况；由于对于视频每个像素都要建立一个甚至多个码本模型，训练模型比较耗时，若需要背景重构，将大大降低运算速度。

7. ViBe

优点：思想简单，易于实现。（用有限样本近似无限时间窗口）计算量小。运算效率高。（样本少；优化了相似度匹配算法）样本衰减策略。（随机更新策略使得样本生命周期指数衰减，不同于其他方法的先进先出）

缺点：有鬼影、静止目标、阴影前景和运动目标不完整等问题。

8. SuBSENSE

优点：使用了反馈机制，更好适应不同场景，对噪声更为鲁棒。

缺点：（PPT 没有）

二、2019-2020

1、论述什么是多媒体？多媒体分析与理解有哪些应用领域和挑战？（10 分）

PPT2 前面

2、简述特征表示学习的必要性，及其主要解决方法与基本原理。（15 分）

PPT2 后面

3、简述反向误差传播算法的基本原理，并说明该算法在神经网络训练过程中可能遇到的问题及其应对措施。（15 分）

PPT3 前面

4、简述图像语义理解的基本含义及其面临的主要挑战，选取 1 个用于图像语义理解任务的典型算法或模型对其具体实现过程进行详细说明。（15 分）

PPT4 后面 PPT5 前面

5、在 NetFlix 比赛中，单模型性能最好前两种算法是什么？分别阐述其基本原理及它们之间的优劣。（15 分）

PPT11 后面，上面第 5 题

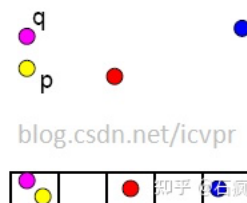
6、简单描述迭代量化哈希方法(Iterative Quantization, ITQ) 的基本思想，并比较 ITQ 方法与局部敏感哈希(LSH) 方法的优劣。（15 分）

PPT10 前面，上面第 6 题

局部敏感哈希：

一般的 hash，原内容发生微小变化后，hash 值的变化是无法预估的（字符串改一个字母后，整个字符串 md5 变得完全不一样；图片改一个像素后 hash 值也变得完全不一样）局部敏感 hash 的改进在于，原内容发生微小变化后，其 hash 值也只发生微小变化。从而满足原内容相近 hash 值也相近的良好性质。这种性质的好处在于，可以在 hash 空间进行近邻检索。

我们知道，通过建立 Hash Table 的方式我们能够得到 $O(1)$ 的查找时间性能，其中关键在于选取一个 hash function，将原始数据映射到相对应的桶内（bucket, hash bin），例如对数据求模： $h = x \bmod w$ ， w 通常为一个素数。在对数据集进行 hash 的过程中，会发生不同的数据被映射到了同一个桶中（即发生了冲突 collision），这一般通过再次哈希将数据映射到其他空桶内来解决。这是普通 Hash 方法或者叫传统 Hash 方法，其与 LSH 有些不同之处。



局部敏感哈希示意图（from: Piotr Indyk）

LSH 的基本思想是：将原始数据空间中的两个相邻数据点通过相同的映射或投影变换（projection）后，这两个数据点在新的数据空间中仍然相邻的概率很大，而不相邻的数据点被映射到同一个桶的概率很小。也就是说，如果我们对原始数据进行一些 hash 映射后，我们希望原先相邻的两个数据能够被 hash 到相同的桶内，具有相同的桶号。对原始数据集中所有的数据都进行 hash 映射后，我们就得到了一个 hash table，这些原始数据集被分散到了 hash table 的桶内，每个桶会落入一些原始数据，属于同一个桶内的数据就有很大的可能是相邻的，当然也存在不相邻的数据被 hash 到了同一个桶内。因此，如果我们能够找到这样一些 hash functions，使得经过它们的哈希映射变换后，原始空间中相邻的数据落入相同的桶内的话，那么我们在该数据集中进行近邻查找就变得容易了，我们只需要将查询数据进行哈希映射得到其桶号，然后取出该桶号对应桶内的所有数据，再进行线性匹配即可查找到与查询数据相邻的数据。换句话说，我们通过 hash function 映射变换操作，将原始数据集分成了多个子集合，而每个子集合中的数据间是相邻的且该子集合中的元素个数较小，因此将一个在超大集合内查找相邻元素的问题转化为了在一个很小的集合内查找相邻元素的问题，显然计算量下降了很多。

那具有怎样特点的 hash functions 才能够使得原本相邻的两个数据点经过 hash 变换后会落入相同的桶内？这些 hash function 需要满足以下两个条件：

1) 如果 $d(x, y) \leq d_1$ ，则 $h(x) = h(y)$ 的概率至少为 p_1 ；

2) 如果 $d(x, y) \geq d_2$ ，则 $h(x) = h(y)$ 的概率至多为 p_2 ；

其中 $d(x, y)$ 表示 x 和 y 之间的距离， $d_1 < d_2$ ， $h(x)$ 和 $h(y)$ 分别表示对 x 和 y 进行 hash 变换。

满足以上两个条件的 hash functions 称为 (d_1, d_2, p_1, p_2) -sensitive。而通过一个或多个 (d_1, d_2, p_1, p_2) -sensitive 的 hash function 对原始数据集进行 hashing 生成一个或多个 hash table 的过程称为 Locality-sensitive Hashing。

使用 LSH 进行对海量数据建立索引 (Hash table) 并通过索引来进行近似最近邻查找的过程如下:

1. 离线建立索引

- (1) 选取满足 $(d1, d2, p1, p2)$ -sensitive 的 LSH hash functions;
- (2) 根据对查找结果的准确率 (即相邻的数据被查找到的概率) 确定 hash table 的个数 L , 每个 table 内的 hash functions 的个数 K , 以及跟 LSH hash function 自身有关的参数;
- (3) 将所有数据经过 LSH hash function 哈希到相应的桶内, 构成了一个或多个 hash table;

2. 在线查找

- (1) 将查询数据经过 LSH hash function 哈希得到相应的桶号;
- (2) 将桶号中对应的数据取出; (为了保证查找速度, 通常只需要取出前 $2L$ 个数据即可);
- (3) 计算查询数据与这 $2L$ 个数据之间的相似度或距离, 返回最近邻的数据;

LSH 在线查找时间由两个部分组成: (1) 通过 LSH hash functions 计算 hash 值 (桶号) 的时间; (2) 将查询数据与桶内的数据进行比较计算的时间。因此, LSH 的查找时间至少是一个 sublinear 时间。为什么是“至少”? 因为我们可以通过对桶内的属于建立索引来加快匹配速度, 这时第 (2) 部分的耗时就从 $O(N)$ 变成了 $O(\log N)$ 或 $O(1)$ (取决于采用的索引方法)。

LSH 为我们提供了一种在海量的高维数据集中查找与查询数据点 (query data point) 近似最相邻的某个或某些数据点。需要注意的是, LSH 并不能保证一定能够查找到与 query data point 最相邻的数据, 而是减少需要匹配的数据点个数的同时保证查找到最近邻的数据点的概率很大。

7、阐述 PageRank 的基本原理, 并试着阐述还可能有哪些改进的方式。(15 分)

PageRank 算法是由斯坦福大学 (Stanford University) 的 Sergey 和 Lawrence 在 1998 年提出的, 是目前 Google 用来表示网页重要性的综合性指标, 而且该指标不会受到各种检索 (引擎) 的影响。

PageRank 算法借鉴了传统的情报检索理论中的引文分析方法: 当网页 A 有一个链接指向网页 B 时, 就认为网页 B 获得了一定的分数, 该分值的多少取决于网页 A 的重要程度, 即网页 A 的重要性越大, 网页 B 获得的分数就越高。我们通过图 1 来具体地看一下刚才所阐述的方法。具体的算法是, 将某个页面的 PageRank 除以存在于这个页面的“正向链接”, 由此得到的值分别和正向链接所指向的页面的 PageRank 值相加, 即得到了被链接的页面的 PageRank 值。

通过以上对 PageRank 算法原理的分析, 最终推导出了 PageRank 值的计算公式。

定义 1 令 $PR(A)$ 表示给定页面 A 的 PageRank 值; d 为阻尼因子, 一般设为 0.85, $PA(T_i)$ 表示一个指向 A 页的网站其本身的 PageRank 值, $C(T_i)$ 表示该页面所拥有的导出链接数量; $PR(T_n)/C(T_n)$ 表示为每一个指向 A 页的页面重复相同的操作步骤, 则页面 A 的 PageRank 值计算公式见公式 1。

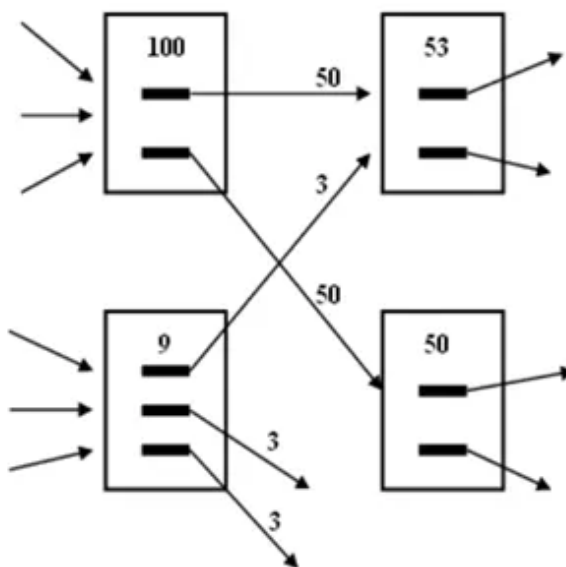


图 1 PageRank 概念图

$$PR(A) = (1 - d) + d \left(\frac{PR(T_1)}{C(T_1)} + \dots + \frac{PR(T_n)}{C(T_n)} \right) \quad [1]$$

的缺点: (1)比如 PageRank 算法偏重旧网页, 因为旧网页被其它网页链接到的可能性更高, 而事实上新的网页可能会具有更好的信息价值; (2)PageRank 算法偏重以.com 结尾的网站, 因为这类网站往往是综合性网站, 自然可以比其它类型的网站获得更多链接, 而事实上某些专业网站对问题的阐述更具有权威性; (3)PageRank 算法无法区分网页中的超链接是和网页主题相关还是无关, 即无法判断网页内容上的相似性, 这样就容易出现主题漂移问题。比如, Google, Yahoo 是互联网上最受欢迎的网页, 拥有很高的 PageRank 值。这样, 如果用户输入一个查询关键字时, 这些网页往往也会出现在该查询的结果集中, 并会占据很靠前的位置, 而事实上这个网页与用户的查询主题有时并不太相关。

3 PageRank 的改进算法

继 Lawrence Page 提出 PageRank 算法以后, 很多研究者对 PageRank 进行了改进, 主要有以下几个方面。

3.1 加速评估的 PageRank 改进算法

针对缺点(1), 上海交通大学的张玲博士^[8]提出了一个加速评估算法, 该算法使得网络上有价值的内容以更快的速度传播; 相反, 一些已经陈旧的数据的页面评估值也将加速下滑。算法的核心思想是通过分析基于时间序列的 PageRank 值变化情况, 预测某个 URL 在未来一段时期内的期望值并把它作为搜索引擎提供检索服务的有效参数。

算法定义了一个 url 的加速因子 AR 为

$$AR = PR \times \text{sizeof}(D)$$

其中, $\text{sizeof}(D)$ 为整个页面集合的文档数。

加速后的 PageRank 表达式为

$$PR_{\text{accelerate}} = \frac{AR_{\text{last}} + BD}{M_{\text{last}}}$$

其中, AR_{last} 是 url 最近一次的 AR 值, B 是该 url 一段时间内 PageRank 值的二次拟合曲线的斜率, D 为离最近一次页面下载的时间间隔天数, M_{last} 是最近一次下载的文档集内的文档数目。

用户检索时, 搜索引擎将按照预测的 PageRank 值的高低决定一个 URL 在检索结果中的位置。

3.2 主题敏感的 PageRank 算法

斯坦福大学计算机科学系 Taher Haveliwala^[6]提出了一种主题敏感(Topic-sensitive)的 PageRank 算法解决了上文提到的缺点(3)。该算法考虑到有些页面在某些领域被认为是重要的, 但并不表示它在其它领域也是重要的。所以, 算法先根据 Open Directory^[7]列出 16 个基本主题向量, 并对每个网页离线计算出对这些基本主题向量的 PageRank 值。在用户查询时, 算法根据用户输入的查询主题或查询的上下文, 计算出该主题与已知的基本主题的相似度, 在基本主题中选择一个最接近的主题代替用户的查询主题。算法的形式化表示如下:

$$R(u) = M^T \times R(u) = cM^T \times R(u) + (1-c)p_u$$

其中 P_u 是网页 u 的主题敏感向量。

该算法可以有效地避免一些明显的主题漂移现象, 比如在查询“美洲虎”时, 有上下文的指引, 算法能明确地区分用户是在查询的是: (1)美洲虎汽车; (2)美洲虎橄榄球队; (3)美洲虎产品; (4)哺乳动物美洲虎, 提供高质量的推荐结果集。

3.3 结合链接分析和文本内容的 PageRank 算法

华盛顿大学计算机科学与工程系的 Matthew Richardson 和 Pedro Domingos^[8]认为用户从一个网页跳到另一个网页是受到当前网页内容和正在查询的主题的影响的, 所以提出

了一个结合链接和内容信息的 PageRank 算法。该算法的计算形式为

$$P_q(j) = (1-\beta)P_q(j) + \beta \sum_{i \in B_j} P_q(i)P_q(i \rightarrow j)$$

这里 $P_q(i \rightarrow j)$ 指用户在查询主题 q 下从网页 i 跳转到 j 的可能性。 $P_q(j)$ 表示用户在网页中没有链出链接时, 跳转到 j 的可能性。而 $P_q(j)$ 就是在查询 q 下的网页 j 的 PageRank 值。虽然 $P_q(i \rightarrow j)$ 和 $P_q(j)$ 是任意分布的, 为了统一这两个值, 作者采用一个查询 q 和网页 j 的相关函数 $P_q(j)$ 分别计算这两个值:

$$P_q(j) = \frac{R_q(j)}{\sum_{k \in W} R_q(k)}, \quad P_q(i \rightarrow j) = \frac{R_q(j)}{\sum_{k \in B_i} R_q(k)}$$

其中, W 是整个网络的网页集合, B_i 是网页 i 的链出网页集合。有关函数 $P_q(j)$ 是任意的, 一般取在网页 j 的文本中查询 q 的关键字出现的次数。

3.4 其它的改进算法

由于网页采用了半结构化的 HTML 语言, 其包含有丰富的结构信息, 上海交通大学的宋聚博士^[9]认为在抽取网页的主题内容时应加以利用。对位于 $\langle \text{head} \rangle$ 、 $\langle \text{title} \rangle$ 、 $\langle \text{meta} \rangle$ 以及 $\langle \text{a href} \rangle$ 等标记之内的关键词无疑应该重视, 计算时应赋予较大的权重系数。并对权重系数作出如下定义:

$$W_q = t f_q \lg \left(\frac{N}{t f_q} + 0.5 \right) \cdot \text{func}(t_j)$$

$$\text{func}(t_j) = \begin{cases} 3.0 & \text{关键字在链接文字中} \\ 2.0 & \text{关键字在 head / title / H}_1 / \text{H}_2 \text{ 标记中} \\ 1.8 & \text{关键字在 meta 标记中} \\ 1.0 & \text{其它} \end{cases}$$

北京大学计算机系^[10]也提出了利用 HTTP 协议, 记录每个页面最近一次的修改时间, 在运行分析算法的时候把页面修改时间作为控制参数, 给予新修改的页面以较高的权重, 而给予老页面以较低权重。

3.5 本文的改进

本文仔细分析了 PageRank 算法的随机冲浪模型, 认为主题漂移现象发生主要在于传统的 PageRank 对 PageRank 值是平均分配的, 这就导致主题无关网页获得它本不该获得的 PageRank 值, 使 PageRank 值在无效网页扩散。所以本文吸收了 Matthew Richardson 和 Pedro Domingos^[8]的思想, 即认为用户从一个网页跳到另一个网页是受到当前网页内容和正在查询的主题的影响, 提出一个对两个网页进行相似度描述的二阶相似度概念。由于网页的相似性仅仅表现在两个网页之间, 因此二阶相似度概念就定义为某个网页对在网络连接结构中出现的次数 t , 并利用这个二阶相似度形成网络的相似度矩阵 S , $S_{i,j} = t$ 如果 i 有到 j 的链接, 否则 $S_{i,j} = 0$ 。新算法描述为

$$PR(p) = (1-d) + d \times \sum_{i=1}^n \frac{PR(T_i) \times S_{p,T_i}}{S(T_i)}$$

其中 S_{p,T_i} 是在相似度矩阵 S 中网页 p 对 T_i 的相似度值,

$$S(T_i) = \sum_{u \in B_{T_i}} S_{T_i,u}, \quad B_{T_i} \text{ 是网页 } T_i \text{ 的链出连接集合。}$$

通过新算法就可以使网页的 PageRank 值在具有相似主题的网页上传播, 减少主题无关网页对 PageRank 值的扩散。

为验证算法的有效性, 我们对 <http://ent.sina.com.cn> 网站进行爬行, 获得 10 万张有效网页, 分别利用传统的 PageRank 算法和本文改进的算法进行 PageRank 值计算。并模拟查询 10 个不同的主题, 每次取获得结果集的前 100 项。同时, 为取得标准的结果集, 我们利用 Google 搜索引擎的高级搜索功

(下转第 162 页)

三、2020-2021

1、论述什么是多媒体?多媒体分析与理解有哪些应用领域和挑战?(15分)

看往年卷

2、多媒体内容描述有哪几类代表性的特征表示学习方法,对比说明其异同点,并每类列举2-3种典型算法。(20分)

PPT2 后面

3、简述语言模型的基本概念与应用领域,并给出3种典型的语言模型的基本解决思路。(15分)

PPT4

4、在Netflix比赛中,单模型性能最好前两种算法是什么?分别阐述其基本原理及它们之间的优劣。(15分)

看往年卷 推荐系统那一章 ppt

5、简单描述迭代量化哈希方法(Iterative Quantization, ITQ)的基本思想,并比较ITQ方法与局部敏感哈希(LSH)方法的优劣。(20分)

Iterative Quantization: A Procrustean Approach to Learning Binary Codes

CVPR 2011

Yunchao Gong and Svetlana Lazebnik

目的:

学习保持相似度的二值编码,用于大规模图像采集的高效检索。

贡献:

提出了一种有效的最小化方案来寻找一个旋转的零中心数据,以最小化量化误差。这种方法被称为迭代量化(ITQ)。它既可以用于无监督数据嵌入(如PCA),也可以用于有监督数据嵌入(如典型相关分析(CCA))。

学习二进制码的有效方案:

- 1、代码应该很短,这样我们就可以在内存中存储大量的图像。
- 2、这些代码应该将与具有低汉明距离的二进制字符串相似(无论是感知上还是语义上)的图像映射出来。
- 3、学习二进制码参数和编码新的测试图像的算法应该是非常有效的。

方法:

1. 降维

产生一个有效的代码,其中每个比特的方差是最大的。这个目标函数与主成分分析(PCA)完全相同。

$$\mathcal{I}(W) = \sum_k \text{var}(h_k(\mathbf{x})) = \sum_k \text{var}(\text{sgn}(\mathbf{x}\mathbf{w}_k)),$$
$$\frac{1}{n}B^TB = I.$$

$$\begin{aligned}\tilde{\mathcal{I}}(W) &= \sum_k \mathbb{E}(\|\mathbf{x}\mathbf{w}_k\|_2^2) = \frac{1}{n} \sum_k \mathbf{w}_k^T X^T X \mathbf{w}_k \\ &= \frac{1}{n} \text{tr}(W^T X^T X W), \quad W^T W = I. \quad (1)\end{aligned}$$

2. 二进制量化

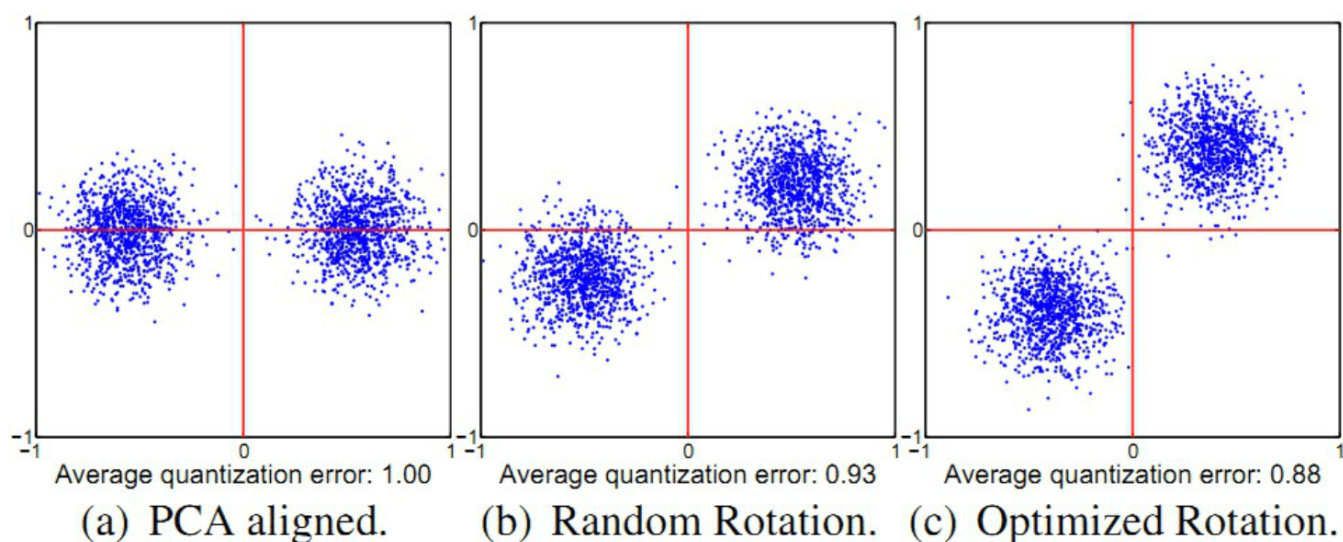
将量化损失最小化,如下所示。

$$Q(B, R) = \|B - VR\|_F^2$$

根据量化损失，迭代“修复 R 更新 B”和“修复 B 更新 R”，直到达到局部最小值。

这篇文章的主要思路是先对原始空间的数据集 $X \in R^{n \times d}$ 用PCA进行降维处理，设经过PCA降维后的数据集为 $V \in R^{n \times c}$ ，该问题就可以转化为将该数据集中的数据点映射到一个二进制超立方体的顶点上，使得对应的量化误差最小，从而而已得到对应数据集优良的二进制编码。

对于PCA降维部分，不做详解，具体可以参阅该文。设 $v \in R^c$ 为原特征空间中某一数据点经过PCA降维后的表示形式，对应在超立方体中的顶点用 $sgn(v) \in \{-1, 1\}^c$ 来表示，要使量化误差最小，即 $v \in R^c$ 与 $sgn(v) \in \{-1, 1\}^c$ 的欧式距离最小，即 $\min \|sgn(v) - v\|^2$ ，对于所有的数据点进行二进制编码后用 B 表示，PCA降维后 $V = X \times W$ ，对整个数据集为 $\min \|B - V\|^2$ 。由于对矩阵进行旋转可以降低量化误差，如下图示：



从图1可以看出，对投影后的矩阵V进行随机旋转后，量化误差降低至0.93，对于找到的最优的旋转矩阵，量化误差降低至0.88(矩阵与正交矩阵相乘实际上就是对矩阵做旋转)。基于这样一个事实，考虑将投影后的数据集V进行旋转变换， $\min \|B - V\|^2$ 便变换为 $\min \|B - VR\|^2$ ，R为旋转矩阵。整个问题域就变成了 $\min \|B - VR\|^2$ 的优化问题，即找出最优的旋转矩阵R和与之对应的编码B。该式的优化可以采用交替跌倒的求解方法：先生成随机矩阵并对其进行SVD分解得到对应的正交矩阵作为R的初始值，然后固定R求B， $B = sgn(V \times D)$ （注意这里截距 $b = 0$ ，因为在原空间已对数据中心化，非常重要），B求出来再通过对 $B \times V$ 进行SVD更新R，交替迭代若干次即可，文中选用的是50次。

6、阐述 PageRank 的基本原理，并试着阐述还可能有哪些改进的方式。（15 分）

看往年卷

四、PPT 目录:

第 01 讲-多媒体内容概述

第 02 讲-多媒体特征表示与提取

1. 多媒体技术概述
2. 传统多媒体特征
3. 多媒体特征表示学习

第 03 讲-基于深度神经网络的特征学习

1. 深度学习概述
2. 深度学习基础知识 (BatchPerceptron, BackPropagation)
3. 典型深度学习模型介绍 (自编码, CNN, RNN, Transformer)

第 04 讲-自然语言理解

1. 自然语言概述
2. 自然语言理解技术
3. 几个典型语言模型 (N-gram, NNLM, Word2Vec, ELMo, GPT, Bert)

第 05 讲-图像视频语义理解

1. 图像语义理解 (上一章最后图像分类。图像标注, 目标检测, 图像语义分割, 图像语义描述, 视觉问答)
2. 视频语义理解: 行为分析

第 06 讲-cross media analysis

1. cross-modality, cross-domain/collection, cross-space
(cross-media①Retrieval②Ranking③Hashing④topic modeling⑤other application)

第 07 讲-社交媒体分析与理解

1. 社交媒体综述
2. 多模态网络表示学习
3. 模型安全与数据隐私

第 08 讲-基于内容的图像视频检索

1. 困难与挑战
2. 各种图像检索与技术分析

第 09 讲-视频检索

困难, 内容, 镜头检测原理方法, 镜头拼接

第 10 讲-Rank and Hashing

1. 索引与排序 (PageRank, LSH vs ITQ)
2. 近似近邻搜索
3. 总结展望

第 11 讲-多媒体内容推荐-2020. pdf

1. 基于协同过滤的推荐方法 CF
2. 基于内容的推荐方法
3. 推荐系统的评价 (SVD/MF vs RBM)