

一种基于 SA_LDA 模型的文本相似度计算方法

邱先标 陈笑蓉

(贵州大学计算机科学与技术学院 贵阳 550025)

摘 要 计算文本的相似度是许多文本信息处理技术的基础。然而,常用的基于向量空间模型(VSM)的相似度计算方法存在着高维稀疏和语义敏感度较差等问题,因此相似度计算的效果并不理想。在传统的 LDA(Latent Dirichlet Allocation)模型的基础上,针对其需要人工确定主题数目的问题,提出了一种能通过模型自身迭代确定主题个数的自适应 LDA(SA_LDA)模型。然后,将其引入文本的相似度计算中,在一定程度上解决了高维稀疏等问题。通过实验表明,该方法能自动确定模型主题个数,并且利用该模型计算文本相似度时取得了比 VSM 模型更高的准确度。

关键词 文本相似度,SA_LDA 模型,主题模型,文本挖掘

中图分类号 TP391

文献标识码 A

Text Similarity Calculation Algorithm Based on SA_LDA Model

QIU Xian-biao CHEN Xiao-rong

(College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

Abstract Many information processing techniques are based on computing the similarity of text. However, the traditional method of similarity calculation based on vector space model has the problems of high dimension and poor semantic sensitivity, so the performance is not very satisfactory. This paper proposed a self-adaptive LDA (SA_LDA) model based on traditional LDA model. It can manually determine the number of topic. Applying it in text similarity calculation, it can solve the high dimensional and sparse problem. Experiments show that this method improves the accuracy of similarity calculation and the effect of text clustering compared with VSM.

Keywords Text similarity, SA_LDA model, Topic model, Text mining

1 引言

在互联网平台的海量数据中,以文本形式存在的数据占据着很重要的一部分。网络中的文本数据还以各种各样的展现形式存在,例如新闻、网页、微博、论文、评论等,这就使得文本信息的处理变得十分复杂。因此,怎样准确而又高效地从大量的网络文本数据中挖掘出有价值的信息是当前信息处理领域的一个重要研究方向。

文本的相似度计算一直是各种文本信息处理技术的基础,它是文本聚类、文本查重、文本特征提取等技术的重要组成部分^[1]。计算文本相似度是比较文本之间差异性的基础,是很多文本分析过程的前提。例如,在文本聚类中,文本间的相似度直接决定一个文本是否属于该聚簇,对于划分后的每个聚簇,在同一个聚簇内各文本间的相似度较大,在不同聚簇之间的文本相似度较小^[2]。因此,文本相似度的计算也是决定文本聚类处理效果的关键步骤。

在计算文本之间的相似度时,当前使用最多的算法就是基于 TF-IDF 权值的向量空间模型(VSM)^[3]计算方法。在这个模型中,将文本集中的每一个特征项作为向量空间的坐标,将某一个文本中该特征项对应的权值作为该文本在当前维度的坐标值,这样就会将每个文本表示成向量空间中的向量;然

后,两个文本间的相似度就可以通过计算其所对应的向量之间的相似度来表示。但是该方法存在如下问题:1)预处理过后的文本集合所包含的特征项数目十分巨大,生成向量空间模型后会出现维数很高以及语义稀疏的问题;2)由于向量空间模型一般选取文本集中的单词作为特征项,文本中每个词之间的隐含关联信息被忽略了,语义敏感度较差。

本文提出的自适应 LDA(SA_LDA)模型在一定程度上解决了高维稀疏和语义敏感度差的问题。传统 LDA 模型需要事先人为确定主题数目,这样的计算结果会受主观性的影响。主题数目的设定将会直接影响模型的性能^[4]。SA_LDA 模型能根据文本集的信息自动确定要提取的主题个数,提高了模型的自适应性。然后使用该模型对语料库中的文本建模,统计每个文本的隐含主题分布情况,并将其映射到所对应的主题空间中,再对文本相似度进行计算。

2 相关工作

针对向量空间模型在计算文本相似度时存在的一些问题,相关的研究者们也提出了改进措施。王刚等利用本体(Ontology)来描述文本^[5],通过本体间的语义相似度衡量文本间的相似程度。HU 等提出利用外部的维基百科知识来创建概念语义知识库,再通过结合基于概率知识库的相似度和

本文受国家自然科学基金(61363028)资助。

邱先标(1993—),男,硕士生,主要研究方向为自然语言处理、数据挖掘,E-mail:qiuxianbiao@163.com;陈笑蓉(1954—),女,教授,主要研究方向为信息检索与信息挖掘、自然语言处理等,E-mail:xrchengz@163.com(通信作者)。

文本中词语的相似度以及文本类别相似度来对文本的相似度进行计算^[6]。HUANG 等提取维基百科中的概念信息和文本的类别信息来计算文本相似度^[7],这样可以扩充文本的外部相似性。还有其他常用的方法利用隐性语义索引(LSI)模型或概率隐性语义索引(PLSI)进行文本相似度的计算^[8]。LSI 考虑了每个词语间的语义关系,利用奇异值分解(SVD)技术达到降维的效果;但其很有可能会过滤掉一些重要但是稀有的特征,从而造成最终效果不佳。本文针对以上缺点,提出基于 LDA 模型的 SA_LDA 模型,所提模型改进了 LSI 模型在特征提取时的不足,并且能自动确定模型需要提取的主题个数。

LDA 模型是由 Blei 等^[9]于 2003 年提出的一种概率语言模型,它是对 PLSI 模型进行了贝叶斯改进,即对主题和主题对应的特征词加上了先验分布。LDA 模型由于引入了文本的主题分布,因此相比常用的 VSM 模型降低了文本表示的维度,并且其参数空间的大小是不变的,与文本集的规模没有关系,因此对更大规模的文本数据也比较适用。目前,LDA 模型已经在文本信息检索、文本摘要以及图像处理等领域得到了广泛应用。

LDA 模型在应用于文本的建模或者文本特征提取时,就是对文本数据的“隐性语义”进行分析,也即以无监督学习的方式从文本中发现隐含在文本中的“主题”。在该模型中的隐性语义分析是指利用各词项共同出现的频率来确立文本的主题结构,当一些词项的共现频率比较高时,就可以假设它们是属于一个主题的,这样就不需要像知网或维基百科这样的外部知识。该方法也能在一定程度上解决“一词多词”和“一词多义”的问题,在“一词多义”的情况下该词会出现在不同的主题中,在“一词多词”的情况下这些词汇会被分配到一个主题之下,从而解决因“一词多义”或“一词多词”引起的文本特征表示问题。

LDA 模型可以表示为一个三层的贝叶斯网络结构,分别为词层、主题层、文档层^[10]。它是基于一个这样的假设:每篇文档都是由一些隐含的主题组成,而每一个主题又由一些出现在文本中的词语组成。根据以上描述,图 1 给出它的拓扑结构。

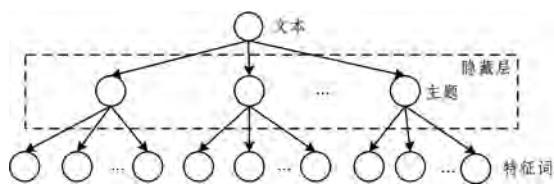


图 1 LDA 模型的拓扑结构

LDA 模型的基本想法就是将文本表示为一些隐含主题的概率分布,这些主题又表示成一些相关词的概率分布^[10]。文本和词是可见的,但是隐藏层的主题是不可见的。

LDA 模型的生成过程如下:对于所有的主题 Z ,由以 β 为参数的狄利克雷分布 $\varphi \sim \text{Dir}(\beta)$ 得到每个主题上单词的概率分布向量 φ ;再根据以 α 为参数的狄利克雷分布 $\theta \sim \text{Dir}(\alpha)$ 得到文本的主题概率的分布向量 θ ;接着根据主题 Z 服从参数为 θ 的多项分布随机选择一个主题 Z_i ;最后从主题 Z_i 的词项分布中选择一个单词 w_i 作为要生成文本中的一个词。循环执行这些步骤,直到生成一个文本。

根据 LDA 模型生成文本中的每一个词的步骤,其层次结

构可以表示为一个有向概率图模型,生成过程如图 2 所示。

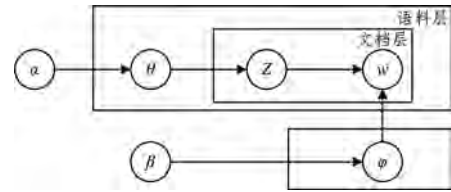


图 2 LDA 的图模型表示

由 LDA 模型的生成过程和以上图模型可知,在 LDA 模型假设下生成一篇文档的概率为:

$$p(w|\alpha,\beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n,\beta) \right) d\theta \quad (1)$$

其中 α, β 为超参数, α 体现了文本集中各主题之间关系的强弱, β 决定主题自身在文本中的分布概率。 θ 表示文档中各主题的概率分布,它是服从于参数为 α 的狄利克雷分布的。 Z 表示文本的主题集, z_n 表示该文本集的第 n 个主题, W 表示文本中所有的特征词, w_n 指模型产生的第 n 个词, φ 表示在某个主题下各词的分布概率,它服从参数为 β 的狄利克雷分布。

在 LDA 模型的众多参数中,每个主题的词概率分布 φ 和每个文本中的主题的概率分布 θ 是两组十分重要的参数。模型的参数估计可被看作是生成过程的一种逆过程:生成过程指根据各给定的参数生成每一篇文本;而参数估计指在已经拥有文本集的条件下,通过训练各个参数,得到各个未知参数的值。LDA 模型的参数估计方法有变分贝叶斯学习推理(Variational Bayesian)、期望传播(Expectation Propagation algorithm)以及 Gibbs 采样等。其中比较流行的 Gibbs 采样先是对每个特征词的主题进行初步采样,在确定各词的主题之后,再统计每个特征词的频次就可以计算出参数。因此,这种对参数的估计就变成了计算特征词序列下的各个主题序列分布的条件概率,在获得每个单词 w 的主题 z 的标号后,可以用如下参数计算公式对模型的参数进行计算:

$$\theta_{m,k} = \frac{n_{m,k}^k + \alpha_k}{\sum_{t=1}^k n_{m,t}^k + \alpha_k} \quad (2)$$

$$\varphi_{k,t} = \frac{n_{k,t}^t + \beta_t}{\sum_{t=1}^v n_{k,t}^t + \beta_t} \quad (3)$$

其中, $\theta_{m,k}$ 指文档 m 中第 k 个主题的概率分布; $\varphi_{k,t}$ 指词项 t 在主题 k 中的分布概率; $n_{m,k}^k$ 表示在文本 m 中出现主题 k 的频次; $n_{k,t}^t$ 表示在第 k 个主题中特征词项 t 出现的频次; α_k 是主题 k 的狄利克雷先验; β_t 是词项 t 的狄利克雷先验。

3 LDA 模型的自适应过程

在传统的 LDA 模型中,主题的个数 k 是需要作为输入参数事先给定的。 K 值的设定会对提取的文本主题结构产生较大影响,若在不知道任何背景知识的前提下指定它的大小,则存在一定的主观性和经验性。最简单的确定最优主题个数的办法是用许多的 k 值进行重复实验,当选用的评价指标(如聚类的准确性)达到最优时,就认为此时的主题个数是最恰当的。Teh 等^[12]针对该问题提出了一种 HDP(Hierarchical Dirichlet Process)方法来自动确定主题个数,该方法通过建立多层的狄利克雷过程来学习主题个数,虽然能自动确定主题数目,但是必须为文本集合同时创建一个 HDP 模型和 LDA 模型,这使得模型的复杂度变得更大,不利于对大规模文本

集进行处理。

在 LDA 模型中,最合适的主题个数 k 不但与文本集中的文本数量和长度有关,还受各文本间的关联程度的影响。曹娟等^[11]通过理论推导以及实验证明:当 LDA 模型提取的各个主题之间相似度的平均值为最小时, k 值是最合理的。当主题之间的相似度最小时,说明各个主题之间具有较强的不相关性,每个独立的主题能更好地代表文本所表达的含义。根据该原理,本文在 LDA 模型的基础上提出了 SA_LDA 模型。我们将在模型训练过程中计算每个主题之间的平均相似度,找到造成主题之间的平均相似度较大的主题数目,通过该参数更新下一次提取的主题个数,反复迭代,直到主题之间的平均相似度收敛到某个值,确定此时模型应该提取的主题个数,以解决最优 k 值的设置问题。

为了便于算法运算,先给出如下定义。

定义 1(两个主题之间的相似度 $\text{corr}(Z_i, Z_j)$) 指主题 Z_i 和 Z_j 对应的词概率分布向量 φ_i 和 φ_j 之间的余弦相似度,其中 M 指的是词向量的维度。

$$\text{corr}(z_i, z_j) = \text{sim}(\varphi_i, \varphi_j) = \frac{\sum_{m=0}^M \varphi_{im} \cdot \varphi_{jm}}{\sqrt{\sum_{m=0}^M (\varphi_{im})^2 \sum_{m=0}^M (\varphi_{jm})^2}} \quad (4)$$

从而得出模型中的全部 k 个主题之间的平均相似度为:

$$\text{Avecorr}(\text{topic}) = \frac{\sum_{i=1}^k \sum_{j=i+1}^k \text{corr}(z_i, z_j)}{k \times (k-1) / 2} \quad (5)$$

定义 2(主题协调值 $\text{Harm}(Z_i, R)$) 利用定义 1 的公式分别计算主题 Z_i 与其他 $k-1$ 个主题之间的相似度,其中相似度的值小于或等于 R 的主题个数记为主题 Z_i 基于 R 的协调值。某个主题与其他主题之间的相似度越小,则该主题的协调值越大。

定义 3(振荡系数 $\text{Oscil}(M, n)$) 对于一个 LDA 模型 M 和一个正整数 n ,在该模型中主题协调值小于或等于 n 的主题个数被称为该模型的振荡系数。振荡系数就是指导致模型中主题相似度较大的主题个数,它决定模型主题个数在下次迭代时变化的大小。

本文提出的 LDA 模型自适应算法的主要思想是:先给定一个随机主题个数,初步得到模型参数;再根据定义 1 分别计算每个主题与其他主题间的相似度值,求出所有主题的平均相似度;然后根据定义 2 计算每个主题的主题协调值,找出协调值较小的主题个数,即模型的振荡系数,通过振荡系数修正模型的主题个数,这样就可以保证每次迭代的主题平均相似度都会减小,从而使模型更优。

根据以上定义和算法思想,可以得出 LDA 模型的自适应过程算法步骤。

输入:文本集数据, LDA 模型超参数 α, β

输出:最佳主题个数 K 及此时的模型参数

Step1 随机给定初始主题的个数 K_0 (K_0 固定在程序中,不需要用户输入),并对文本集进行 Gibbs 采样,得到模型的初始化参数 θ 和 φ ,获得一个初始的模型。

Step2 根据模型参数训练得出的主题-词概率矩阵 φ ,利用定义 1 中的公式计算主题之间的相似度以及所有主题的平均相似度 $p = \text{Avecorr}(\varphi)$,再将 p 作为参数,得到每个主题的主题协调值 $\text{Harm}(Z_i, p)$,利用主题协调值计算出此时的模型振荡系数 $S = \text{Oscil}(M, n)$,此时 n 取值为 0。

Step3 根据上一步的值,采用更新公式 $K_{n+1} = f(p) \times (K_n - S_n) + K_n$ 重新估算主题的个数,并对模型重新采样得

到此时 LDA 模型的参数。

上述公式中的 $f(p)$ 是平均相似度 p 的变化方向。当主题之间的平均相似度变化与上次相反时, $f_{n+1}(p) = -1 \times f_n(p)$,当平均相似度变化与上次一样时, $f_{n+1}(p) = f_n(p)$,其中 $f_0(p) = -1$ 。

Step4 循环执行 Step2 和 Step3,直到主题个数 K 值收敛。

4 应用 SA_LDA 模型进行文本相似度的计算

SA_LDA 模型从统计的角度发掘出文档隐含的主题信息,并能利用主题之间的平均相似度大小来确定主题的个数,将文本集中的每个文本都映射到基于统计的潜在主题特征空间之上,然后再得到每个文本的潜在主题概率分布向量。由于 JS (Jensen-Shannon) 距离针对基于概率分布的相似度计算有很好的效果,因此再采用 JS 距离来计算各文本主题分布的相似度,并以此作为文本的相似度。利用 JS 距离公式计算文档 $p = p_1, p_2, \dots, p_T$ 与 $q = q_1, q_2, \dots, q_T$ 的相似度 $\text{sim}(p, q)$ 如下:

$$\text{sim}(p, q) = D_{JS}(p, q) = \frac{1}{2} [D_{KL}(p, \frac{p+q}{2}) + D_{KL}(q, \frac{p+q}{2})] \quad (6)$$

其中, p 和 q 为两个文本的主题概率向量, $D_{KL}(p, q) = \sum_{j=1}^T p_j \ln \frac{p_j}{q_j}$

为 KL (Kullback-Leibler) 距离。由于其计算距离时不满足相似度对称性,因此一般采用 JS 距离计算相似度。

最后再利用比较基础的聚类算法即 K-means 算法的聚类效果对结果进行评价。该相似度计算算法的具体步骤如下:

输入:文本集数据

输出:文本集的相似度矩阵,聚类的结果

Step1 使用开源工具对文本集中的文本进行预处理,包括分词以及去除停用词等。

Step2 将预处理后的每个文本进行向量化处理,构造成词项-文档矩阵,并以此作为模型的输入。

Step3 利用词项-文档矩阵对文本集进行 SA_LDA 建模,利用其自适应过程得到文本集的主题个数 k ,并得到文本的主题概率分布。

Step4 利用文本隐含主题的概率分布来计算每两个文本之间的相似度值,并得到文本集的相似度矩阵。

Step5 在得到文本相似度矩阵后,使用聚类算法对其进行聚类处理,并用聚类结果的准确性来评估文本相似度的准确性。

根据以上步骤,本文采用的相似度计算方法和文本建模没有依赖任何的外部语义知识库,因此该方法计算文本相似度具有较好的鲁棒性。LDA 建模时对文本向量的维度压缩较大,因此理论上其相对于 VSM 模型会提升文本相似度的计算效率。本文相似度计算方法的效果将在下一节进行实验说明。

5 实验设计及结果分析

本文实验使用搜狗实验室提供的新闻文本分类语料库 (Sogou-C),该语料库由人工标注好的新闻文本组成。其中的数据分为 10 类:汽车类 (C07)、财经类 (C08)、IT 类 (C10)、健

康类(C13)、体育类(C14)、旅游类(C16)、教育类(C20)、招聘类(C22)、文化和军事类(C24)。这些类别各含有 8000 篇文章,本实验从每一类中随机选出 200 篇文章作为测试样本,共 2000 篇文章。

本实验采用中科院的自然语言处理工具 ICTCLAS 对数据集进行预处理,将文本向量化后再进行 SA_LDA 建模。按照上节描述的步骤,计算测试数据集文本的相似度,并通过文本聚类结果评价相似度计算的准确性。在本次实验中,模型参数的初始值选择如下:自适应算法初始的主题个数 $k_0 = 40$,LDA 模型的狄利克雷先验参数 $\alpha = 40/k, \beta = 0.01$ 。

本文采用的评价指标主要是 F 度量值。在评价文本聚类效果时,两个相对基础的指标分别是准确率和召回率,5.2 节实验也会参考这两个指标。然而通常情况下,召回率和准确率的值在某些时刻是会相互影响的,当一个值变大时,另一个就很可能变小。F 值是结合准确率以及召回率的一种综合度量,它更能综合体现出算法的性能。通常情况下,F 值越大,表示文本聚类算法具有更好的效果。针对聚类结果 j 和预定义类别 i 的准确率 $P(i, j)$ 、召回率 $R(i, j)$ 、F 值的公式如下:

$$P(i, j) = \frac{n_{ij}}{n_j} \quad (7)$$

$$R(i, j) = \frac{n_{ij}}{n_i} \quad (8)$$

$$F = \frac{2 \times P(i, j) \times R(i, j)}{P(i, j) + R(i, j)} \quad (9)$$

其中, n_j 是簇 j 中文本的数量, n_i 是类 i 中文本的数量, n_{ij} 表示簇 j 包含类 i 中文本的数量。

5.1 自适应过程选取主题个数

使用 LDA 模型能达到的效果与主题数目的选取有很大的关系,本文提出的 SA_LDA 模型能根据主题平均相似度确定主题数目。本次实验的设计就是为了检验自适应算法对主题个数选取的准确性。图 3 表示模型主题个数以 40 为单位递增时主题之间的平均相似度的变化,由图可知,当主题数目为 280 时,各主题之间具有最小的平均相似度。根据本文第 3 节的理论,此时的主题数目在该文本集下是最优的。当我们使用未改进前的 LDA 模型进行实验验证时,主题数目从 40 开始,以 40 为单位递增,测得各个主题个数下使用 K-means 算法聚类后的 F 度量值,结果如图 4 所示,当主题数目为 280 时 F 度量值达到最大,此时的模型确实是最好的。由此可知,本文所采取的自适应算法的原理是可信的。

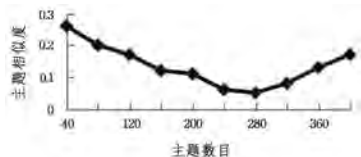


图 3 不同数目下主题的平均相似度

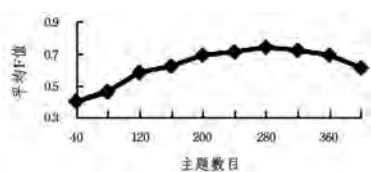


图 4 不同主题数目的聚类结果

使用第 3 节提出的自适应算法,测试在给定不同的初始值 k_0 时算法的迭代次数以及迭代后选取的主题数目,结果如

表 1 所列。当初始值 k_0 越接近于最优主题数时,达到最优值时所需的迭代次数就越少,迭代后最终的主题数目在 280 左右徘徊。由此可知,所提自适应算法可以找出模型的最佳主题个数。

表 1 不同初始值下自适应过程自动获取的主题数

初始主题数目 k_0	迭代次数	最终的主题数
40	21	276
150	18	279
250	4	278
350	9	280
450	28	283

5.2 使用 SA_LDA 计算文本相似度的效果

本文提出的 SA_LDA 模型仅仅是对 LDA 模型需要事先确定主题数目问题的一种自适应性改进。理论上来说,当 LDA 模型的主题个数选取恰当时,两者能达到的效果是一样的,LDA 模型需要人为设置主题数,但需要具有一定的经验性;SA_LDA 模型的优势在于能自动确定主题数。本次实验将使用本文介绍的相似度算法与当前比较流行的基于 TF-IDF 的相似度计算方法进行比较。分别统计基于这两种方法计算相似度的情况下 K-means 聚类的准确率、召回率、F 值,总测试文档篇数为 2000,结果如图 5 所示。由图可知,利用本文提出的 SA_LDA 模型计算文本相似度比基于 TF-IDF 方法来计算文本相似度后的聚类结果更优,因此本文方法的相似度计算更加准确。

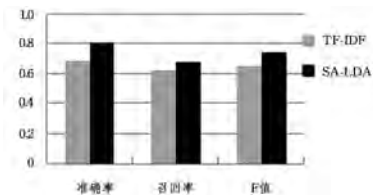


图 5 两种方法聚类结果的比较

结束语 本文首先分析了当前较为流行的基于 TF-IDF 的 SVM 文本相似度计算方法。该方法具有高维稀疏和语义敏感度较差的问题,因此利用其进行文本相似度计算的效果不太理想,LDA 模型可以使这些问题在一定程度上得到解决。但是,传统 LDA 模型需要事先确定文本集中隐含的主题数目,该过程具有一定的主观性,并且会直接影响模型的性能。因此,本文提出了一种自适应 LDA(SA_LDA)模型,其能根据主题之间的相关程度通过迭代确定模型的主题数目,减少了人为参与,提高了模型的自适应性。实验表明,SA_LDA 模型确实能自动获取最优主题数目。将 SA_LDA 模型应用于文本相似度的计算中,相比于基于 TF-IDF 的向量空间模型方法,其显著地提升了文本聚类以及文本相似度计算的效果。由于 LDA 模型易于扩展,为了提升模型训练时的时间效率,下一步将考虑使用快速 Gibbs 采样在迭代时基于上一步的结果更新每个文本的主题分布概率以及每个主题的词项分布概率等参数,以避免每次都重新估算参数,从而提升算法的效率。

参考文献

- [1] XU L, SUN S, WANG Q. Text similarity algorithm based on semantic vector space model[C]// 15th IEEE/ACIS International Conference on Computer and Information Science. 2016.

结束语 本文讨论了基于消息传递的稀疏贝叶斯学习 (SBL), 提出了基于拉伸因子图的低复杂度 BP-MF SBL 算法, 此拉伸因子图是通过在传统因子图添加额外的硬约束因子而得到的。仿真结果显示, 在 MSE 性能和复杂度方面, BP-MF SBL 算法的性能优于当前较好的算法 MF SBL。

参 考 文 献

- [1] DONOHO D L. Compressed sensing[J]. IEEE Trans. Inform. Theory, 2016, 52(4): 1289-1306.
- [2] WIPF D P, RAO B D. Sparse Bayesian learning for basis selection[J]. IEEE Trans. Signal Processing, 2014, 52(8): 2153-2164.
- [3] CHEN S, DONOHO D L, SAUNDERS M A. Atomic decomposition by basic pursuit[J]. SIAM Journal on Scientific Computing, 2008(1): 33-61.
- [4] TROPP J A. Greed is good: algorithmic results for sparse approximation[J]. IEEE Trans. Information Theory, 2013: 2231-2242.
- [5] TIPPING M E. Sparse Bayesian learning and the relevance vector machine[J]. Journal of Machine Learning Research, 2011, 1: 211-244.
- [6] TIPPING M E, FAUL A C. Fast marginal likelihood maximisation for sparse Bayesian models[C] // Proc. 9th International Workshop on Artificial Intelligence and Statistics. 2003: 3-6.
- [7] SHUTIN D, BUCHGRABER T, KULKARNI S R, et al. Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals[J]. IEEE Trans. Signal Processing, 2011, 59(12): 6257-6261.
- [8] PEDERSEN N L, MANCHÓN C N, BADIU M A, et al. Sparse estimation using Bayesian hierarchical prior modeling for real and complex linear models[J]. Signal Processing, 2015, 115: 94-109.
- [9] TAN X, LI J. Computationally efficient sparse Bayesian learning via belief propagation[C] // 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, 2009: 1566-1570.
- [10] BARON D, SARVOTHAM S, BARANIUK R. Bayesian compressive sensing via belief propagation[J]. IEEE Trans. Signal Processing, 2010, 58(1): 269-280.
- [11] SOM S, SCHNITER P. Compressive imaging using approximate message passing and a Markov-tree prior[J]. IEEE Trans. Signal Processing, 2012, 60(7): 3439-3448.
- [12] AL-SHOUKAIRI M, RAO B. Sparse Bayesian learning using approximate message passing[C] // 2014 48th Asilomar Conference on Signals, Systems and Computers. 2014: 1957-1961.
- [13] XING E P, JORDAN M I, RUSSELL S A. A generalized mean field algorithm for variational inference in exponential families[C] // Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI'03). San Francisco, CA, USA, 2013: 583-591.
- [14] BISHOP C M, WINN J. Structured variational distributions in VIBES[C] // Proceedings of Artificial Intelligence and Statistics. 2003: 3-6.
- [15] DAUWELS J. On variational message passing on factor graphs[C] // Proc. IEEE International Symposium on Information Theory (ISIT 2007). 2007: 2546-2550.
- [16] PEDERSEN N L, MANCHÓN C N, SHUTIN D, et al. Application of Bayesian hierarchical prior modeling to sparse channel estimation[C] // 2007 IEEE International Conference on Communications (ICC 2012). 2012: 3487-3492.
- [17] HANSEN T L, JØRGENSEN P B, BADIU M, et al. Joint sparse channel estimation and decoding: Continuous and discrete domain sparsity[J]. arXiv: 1507. 02954.
- [18] RIEGLER E, KIRKELUND G E, MANCHÓN C N, et al. Merging belief propagation and the mean field approximation: A free energy approach[J]. IEEE Trans. Inform. Theory, 2013, 59(1): 588-602.
- [19] RANGAN S. Generalized approximate message passing for estimation with random linear mixing[C] // Proc. IEEE Int. Symp. on Inform. Theory (ISIT 2011). 2011: 2168-2172.
- [20] PEDERSEN N L, MANCHÓN C N, FLEURY B H. Low complexity sparse Bayesian learning for channel estimation using generalized mean field[C] // 20th European Wireless Conference. 2014: 838-843.
- [21] KSCHISCHANG F, FREY B, LOELIGER H A. Factor graphs and the sum-product algorithm[J]. IEEE Trans. Inform. Theory, 2001, 47(2): 498-519.
- (上接第 109 页)
- [2] FAN Z X, CHEN S Y, ZHA L, et al. A Text Clustering Approach of Chinese News Based on Neural Network Language Model[J]. International Journal of Parallel Programming, 2016, 44(1): 198-206.
- [3] CAO Q M, GUO Q, WANG Y L, et al. Text clustering using VSM with feature clusters[J]. Neural Computing & Applications, 2015, 26(4): 995-1003.
- [4] GUO L T, LI Y, MU D J, et al. A LDA model based topic detection method[J]. Journal of Northwestern Polytechnical University, 2016, 34(4): 98-102.
- [5] 王刚, 钟国祥. 一种基于本体相似度计算的文本聚类研究[J]. 计算机科学, 2010, 37(9): 222-224.
- [6] HU X H, ZHANG X, et al. Exploiting Wikipedia as External Knowledge for Document Clustering[C] // ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 389-396.
- [7] HUANG A, MILNE D, FRANK E, et al. Clustering Documents using a Wikipedia Based Concept Representation[M] // Advanced in Knowledge Discovery and Data Mining. Spring Berlin Heidelberg, 2009: 628-636.
- [8] HOFMANN T. Probabilistic latent semantic indexing[C] // 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley, CA, USA, 1999: 50-57.
- [9] BLEI D, NG A, JORDAN M. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003(3): 993-1022.
- [10] 徐戈, 黄厚峰. 自然语言处理中主题模型的发展[J]. 计算机学报, 2011, 34(8): 1423-1437.
- [11] 曹娟, 张勇东. 一种基于密度的自适应最优 LDA 模型选择方法[J]. 计算机学报, 2008, 31(10): 1780-1788.
- [12] TEH Y, JORDAN M, BEAL M, et al. Hierarchical dirieght processes[J]. Journal of the American Statistical Association, 2007, 101(476): 1566-1581.
- [13] 张超, 陈利, 李琼. 一种 PST_LDA 中文文本相似度计算方法[J]. 计算机应用研究, 2016, 33(2): 375-377.
- [14] 黄承慧, 印鉴, 侯昉. 一种结合词项语义信息和 TF-IDF 方法的文本相似度度量方法[J]. 计算机学报, 2011, 34(5): 856-864.