LxMLS 2021 夏令营材料

内容

1. 数学基础
2. 编程入门
3. 文本特征和分类模型
4. 序列文本机器学习(HMM)
5. 文本结构化预测(CRF)
6. 深度文本处理技术

材料

1. 夏令营：主页
2. 课程视频：Youtube
3. 实验手册：PDF
4. 实验代码：Github
5. 重点提示：PDF

说明

1. 将代码和模型相印证，是学习机器学习和深度学习的最佳方法
2. 结合上课内容，搞懂模型是怎么代码实现的，直到自己能写出来，才真正理解和掌握了这些算法，并具备了机器学习的研究和开发能力
3. 这套代码就是一套帮助你掌握这些算法的最佳工具
4. 这些代码非常清晰，易懂。某种程度上，比看公式推导，更容易懂
5. 目标是自己能从0开始，把算法写出来，并把实验完成
6. 这是真正值得学习的

一、数学基础

本节介绍大数据算法的数学基础

1. 材料
    1. Mario Figueiredo，概率论和线性代数基础教程，PDF，原链接, 本地链接
2. 重点
    1. Binomial分布，Multinomial分布
    2. 高斯分布
    3. 贝叶斯定理
    4. 指数族：重点的重点。第36页–第38页详细推导
    5. 特别是“指数家族”概率分布的Partition函数对\eta求导过程。这对理解后面的所有模型（直到CRF）都非常关键
    6. 向量空间、范数、内积
    7. 高斯–施瓦茨不等式，特别是第59页其精妙的证明
    8. 矩阵的Rank、SVD
    9. Convex函数

二、编程入门

本节介绍大数据算法的编程基础，包括Python入门、梯度下降、线性回归

1. 材料
    1. Luis Pedro Coelho, Python、Numpy、Matplotlib 入门，Github
    2. Andre Martins，文本机器学习简介：线性学习器，PDF，原链接, 本地链接
2. 重点
    1. 第26-34页，线性回归模型
        1. 扩展，支持非线性
        2. 误差函数概念：Squared Loss
        3. 概率理解：基于高斯噪声和最大似然估计，基于高斯先验的L2正则
    2. 第92页-93页，梯度下降优化方法
3. 实验：Lab5：父子身高相关吗？腾讯文档

三、文本特征和分类模型

本节介绍大数据文本处理的特征工程和分类模型基本概念。

材料

1. Andre Martins，文本机器学习简介：线性学习器，，PDF，原链接, 本地链接

重点 1：文本特征、分数的基本概念

1. 第8-9页，示例
    1. 文本分类中 P(类型|单词) 的统计及物理意义
    2. 这些“单词”，通常被称作“特征”
    3. 统计得到的 P(类型T|单词A) 表示了“单词A”对判断文档是否属于“类型T”的“绝对价值”，通常被称为“分数” – Score
2. 文本特征的表示
    1. 特别简单，就是0/1
    2. 出现了特征，比如单词“Love”，就标“1”，否则标“0”
    3. 特征向量：比如一共有5个特征，可能表示为 [0,1,1,0,0]
    4. 即：第1，4，5特征没有出现，2，3特征出现了
3. 文本特征的设计
    1. 在上面用“单词”作为判断“文本类型”的“特征”
    2. 除了单个单词，还可以有别的特征吗?
    3. 可以，如:
        1. 单词 + 上下文：n-gram，如 Jack London
        2. 单词 + 单词属性：如 London （人名），这样就和 London（位置）区分开了
        3. 单词 + 上下文 + 属性：如通过 Jack (人名）London，Go （动作）London，我们可以区分这两个 London
    4. CRF就是采用了“单词 + 上下文 + 属性”的特征
4. 分数
    1. 在上面通过统计得到 P(类型|单词)，作为“单词A”对判断文档是否属于“类型T”的“分数”
    2. 除此之外，还可以有别的方法可以得到这个分数吗?
    3. 可以，机器学习模型，将此作为一个最优化的问题，通过各种算法，得到最优“分数”

5. 什么是最优?
    1. 最大熵
    2. 最大似然估计
    3. 最大后验概率
    4. Loss函数最小

重点 2：文本机器学习模型工作原理

1. 需求
    1. 设计一个模型，能够综合一个文档的各个特征 \phi(x_i)，得到其属于某一“类型y_i”的概率 P(y_i|X)
    2. 然后各个类型的概率，选概率最大的，完成分类
2. 多元感知机模型
    1. 最直观、简单、实用的一个模型，也是人工智能、深度学习模型的鼻祖
    2. 模型：第58页
    3. 特点：简单线性模型
    4. 训练方法：第60页
    5. 原理：
        1. 在线算法：来一个样本，就调一次“分数”
        2. 分数在这里用w表示，即特征的权重（weight）
        3. 如果样本类型应该是i，但被模型错误地分类为j，这意味着两个事情：
        4. 对类型i
            1. 这意味着：这个样本的特征，在计算 P(y_i|X) 没有得到足够重视
            2. 该样本的这些特征对应的 w 分数太低，需要增加
            3. 怎么做?
            4. 把文本的特征向量，加到 P(y_i|X) 模型的w上
            5. 达到的效果：如果一个特征出现了（它的x是1），它的分数w就会被加1
            6. 这样，这个样本的各个特征的权重在 P(y_i|X) 模型中是不是被提高了一些?
            7. 这是我们想要的
            8. 为什么只加1，不加100?
            9. 不能着急，慢慢来。看算法，会继续迭代，下次发现错误了，还会加1，慢慢的就加上来了
        5. 对类型j
            1. 与上面对i的w的调整正好相反
            2. 这意味着：这个样本的特征，在计算 P(y_j|X) 时，被错误地得到了太多的重视
            3. 该样本的这些特征对应的 w 分数太高，需要减少
            4. 怎么做?
            5. 把文本的特征向量，在 P(y_j|X) 模型的w上减去
            6. 达到的效果：如果一个特征出现了（它的x是1），它的分数w就会被减1
            7. 这样，这个样本的各个特征在 P(y_j|X) 模型中权重是不是被减少了一些?
            8. 这是我们想要的
    6. 这就是第60页，多元感知机的内在逻辑
    7. 理解这个对理解文本机器学习的特征工程和模型调整非常关键
    8. 后面的所有模型，都是基于这个思路来的

重点 3：Naive Bayes 模型

1. 两个特点：
    1. 它比较的是 P(x,y)，不是上面提到的 P(Y|X)
    2. 它通过条件独立假设，减化 P(X|Y) 的计算
2. 例：第71页–78页

重点 4：逻辑回归

1. 特点
    1. 第87页
    2. 它用一个“指数家族”的形式，模型 P(y|x)
    3. 请回顾数学基础部分“指数家族”概率分布的强大模型能力！
    4. 它还是一个线性模型
2. 优化模型：第90页
3. 注意其中分两项：
    1. 第一部分：Partition函数部分
        1. 请回顾数学基础部分“指数家族”概率分布的Partition函数对\eta求导的公式
    2. 第二部分：文本特征部分
4. 用梯度下降法求解
    1. 第97页：核心的核心。一定要一步步跟下来
    2. 关键是，对最后结果的理解：
        1. 最后得到的梯度的第一部分，指的是：将当前样本的特征向量，乘以：模型得到的各种y的概率，作为这些y的模型的w的梯度
        2. 最后得到的梯度的第二部分，指的是：将当前样本的特征向量，作为该样本的真实y的模型的w的梯度
    3. 它的物理意义非常清楚
    4. 和前面感知机的w的调整算法一脉相乘。非常有趣

实验

1. 类似 Lab 5，完成 Day 1 的实验手册阅读和 linear_classifiers 目录下的1个实验。任务是亚马逊评论文本的情感分类，包括三个模型
    1. Naive Bayes
    2. Perceptron
    3. 最大熵模型，其中又包括 L–BFGS、SGD 两种优化方法

四、序列文本机器学习

本节介绍大数据文本处理的序列模型：HMM 和 CRF

材料

1. Noah Smith，序列模型，PDF，本地链接

重点 1：文本中为什么要”隐状态“?

1. 我们需要感知单词的状态
2. POS标签（词性）能够给单词加上更多有用的特征

3. 比如：“天”，可以是名词，也可以是感叹词。两者是完全不一样的。加了词性后，特征表示更加准确。
4. NER（命名实体识别）对NLU（自然语言理解）非常重要

重点 2：维特比译码

1. 根据POS标签、NER的有监督数据（有标签），能够很方便地统计出 HMM 的 两个概率
2. 数据量少的时候，加入先验概率，改进模型的泛化能力（第184页）
    1. Transition概率
    2. Emission概率
3. 这样就得到了 HMM 模型
4. 问题：来了一个新的句子，如何感知其单词的状态?
5. 算法：维特比译码
6. 原理
    1. 从前往后，对每个单词的每一种可能状态，都算出，从第一个单词开始，各种可能的状态序列下，到达该状态的，“最大”概率。直到最后一个单词
    2. 最后那个单词的概率最大的状态，就是这个单词的状态的最佳估计
    3. 然后从后往前，依次找这个最佳估计是从哪来的，就可以得到前一个单词的最佳状态
    4. 由此完成所谓的“译码”
7. 观察它的式子，这是属于 max-product 的形式
8. 扩展：很多算法，都属于这种，叫 “动态规划”。它们原理差不多，只是形式不同，比如我们学过的 Dijkstra 最短路径算法

重点 3：句子出现概率

1. 有了HMM模型，如何算出现在的这个句子的出现概率?
2. 类似维特比算法
3. 从前往后，对每个单词的每一种可能状态，都算出，从第一个单词开始，各种可能的状态序列下，到达该状态的，概率的“和”。直到最后一个单词
    1. 和前面维特比算法比较，可以发现，它和维特比的唯一差别是：它计算“和”，维特比是计算“最大”
    2. 这个叫“前向”算法
    3. 最后一个单词的概率，就是这个句子出现的概率
4. 类似的，可以有“后向”算法
    1. 从后往前，算出从这个状态出发，生成后面的单词的各种状态序列的概率的和。直到第一个单词
    2. 第一个单词的概率，就是这个句子出现的概率
5. 观察它的式子，这是属于 add-product 的形式
6. 在抽象代数里，它们都属于“半环”的代数结构
7. 可以扩展到其它计算目标

重点 4：i位置单词的状态分布

1. 精确地算出i位置单词的状态分布。利用它，就可以统计获得一个新的 Emission 概率，因此实现无监督HMM模型训练的一次迭代
    1. 即经典的EM算法
    2. E：算i位置单词的状态分布（是平均的）

3. M：基于状态分布，MLE估计，得到新的 Transition 和 Emission 概率

2. 方法
    1. 固定 i位置单词的状态，为j，此时，生成当前句子的各种状态序列的总概率就是 i位置状态 为 j 的 前向概率 * 后向概率
    2. 算出所有可能状态的这个概率，做归一化，就得到了 i位置单词的状态分布
3. 应用
    1. 既然得到了i位置单词的状态分布，那么选择概率最大的状态，作为这个单词的状态，是不是就可以?
    2. 可以，这就是“最大后验概率”解码
    3. 它和维特比译码比起来，哪个好?
    4. 要看你的评价指标，即成本函数
        1. 如果是按句子级统计错误，即一个句子里错一个字，也算错，那么维特比译码好
        2. 如果是按单词级统计错误，即一个句子里错一个字，就算一个错，那么这种译码好
    5. 所以成本函数就很重要。你可以设计各种成本函数

重点 5：i位置单词的状态从j变为k的概率分布

1. 利用它，就可以统计获得一个新的 Transition 概率，因此实现无监督HMM模型训练的一次迭代
    1. 即经典的EM算法
    2. E：算i位置单词的状态分布（是平均的）
    3. M：基于状态分布，MLE估计，得到新的 Transition 和 Emission 概率
2. 方法
    1. 固定 i位置单词的状态，为j，下一个位置的状态为k，此时，生成当前句子的各种状态序列的总概率就是 i位置状态 为 j 的 前向概率 * j的emission概率 * j到k的transition概率 * k的后向概率
    2. 算出所有可能状态的这个概率，做归一化，就得到了 i位置单词状态 从 j变到k的分布

重点 6：EM算法

1. 有两种
    1. 软EM：如上
    2. 硬EM：维特比译码后，基于译码的状态，计算 Transition 和 Emission 概率
2. 硬EM也挺好的
3. EM是一种强大的通用方法。值得掌握

实验

1. 类似 Lab 5，完成 Day 3 的实验手册阅读和 sequence_models 目录下的1个实验。任务是Conll的POS标签预测

五、文本结构化预测

本节介绍结构化预测模型，特别是 CRF

材料

1. Xavier Carreras，学习结构化预测器，PDF，本地链接

重点 1：P(Y|X) 形式的序列模型

1. 模型：第42页
2. 像 HMM，研究的是关于“序列”的模型
3. 和 HMM 不同之处是
    1. HMM 模型 P(X,Y)，是 “生成模型”
    2. 它是“辨别模型”
4. 我们学过的“辨别模型”包括：感知机，最大熵模型
5. 对分类问题，如 POS， NER，模型 P(Y|X) 更直接，效果可能更好

重点 2：结构化感知机

1. 模型：第50页
2. 维特比译码，如果错误，像感知机那样，调整W
3. 通过平均，增加稳定性，能够大大提高模型性能
4. 具有感知机的各项优点
    1. 在线算法
    2. 几个回归就能得到好的结果
    3. 效果接近更复杂的CRF算法
    4. 可以通过Beam Search增加视界，改进性能

重点 3：Log-Linear序列预测模型（包括CRF）

1. 模型：第56页
2. 扩展“指数家族”模型到序列预测
3. Factored模型
    1. “特征”中包括 y_i 和 y_{i-1}
    2. 就可以像HMM那样，用前向、后向、维特比算法（后面会看到）
4. 就变成一个优化问题，可以用梯度下降，优化求解w
5. 两种 Loss 函数
    1. MEMM：最大熵马尔科夫模型
        1. Log 似然，只考虑本地y_i就可以了（第60页梯度公式）
        2. “局部 Loss”
    2. CRF：
        1. Log 似然，考虑全部y序列（第61页）
        2. “全局 Loss“
        3. 所以，要计算各种y序列的情况，然后求和（第62页梯度公式）
        4. 因为 f 只取决于 i-1 和 i，所以可以把公式变为对 i-1 和 i 的各种情况求和（第63页）
        5. 类似 HMM，用前向、后向算法，计算 y_{i-1} = a, y_{i} = b 的各种y序列的概率和
        6. 然后再对所有的 i 求和，这就得到了全局的”期望“特征向量（就是梯度里的第二项）
        7. 梯度的第一项是将样本数据代入后的特征向量
6. 预测
    1. 因为“特征”中包括 y_i 和 y_{i-1}，就可以用维特比译码方法，进行解码
7. 结合深度模型
    1. 用深度模型得到深度表征，再进CRF（第77页）

2. 在CRF层还是这么调w，但同时也调下面的深度表征

实验

1. 类似 Lab 5，完成 Day 4 的实验手册阅读和 learning_structured_predictors 目录下的4个实验。任务依然是ConlI的POS标签预测，但包括如下算法
    1. CRF：ID特征
    2. CRF：扩展特征
    3. 结构化感知机：ID特征
    4. 结构化感知机：扩展特征

六、深度文本处理技术

本节介绍深度文本处理技术

内容

1. 用 Log-Linear，MLP，RNN等模型进行文本分类和结构化预测
2. 首先用Numpy编写完成上述模型，因此真正懂得这些模型的原理和实现。这非常关键
3. 最后学习PyTorch框架中这些模型的使用

实验

1. 类似 Lab 5，完成 Day 2 的实验手册阅读和 non_linear_classifiers 目录下的4个实验。任务是Amazon评论情感分类，包括
    1. Log-Linear模型：Numpy版本
    2. Log-Linear模型：PyTorch版本
    3. MLP模型：Numpy版本
    4. MLP模型：PyTorch版本
2. 类似 Lab 5，完成 Day 5 的实验手册阅读和 non_linear_sequence_classifiers 目录下的4个实验。任务是WSJ的POS标签预测，包括
    1. RNN模型：Numpy版本
    2. RNN模型：PyTorch版本