2023 秋季高级机器学习
习题二

2023.11.03

一. (30 points) **稀疏学习**

上一次习题中涉及到了 PCA 的矩阵低秩近似角度理解。Robust PCA 在此基础上增加了一个变量和正则项：

$$\begin{aligned} \min_{X',E} \quad & \text{rank}(X') + \lambda \|E\|_0 \\ s.t. \quad & X = X' + E \end{aligned} \tag{1}$$

其中 $\|\cdot\|_0$ 为零范数。λ 为正则化参数。为了解该优化问题，我们考虑它的凸松弛（Convex Relaxation）：

$$\begin{aligned} \min_{X',E} \quad & \|X'\|_* + \lambda \|E\|_1 \\ s.t. \quad & X = X' + E \end{aligned} \tag{2}$$

其中 $\|\cdot\|_*$ 为核范数（Nuclear Norm）。使用增广拉格朗日方法（Augmented Lagrangian Method）处理约束条件，可以得到：

$$\min_{X',E} \quad \|X'\|_* + \lambda \|E\|_1 + \langle Y, X - X' - E \rangle + \frac{\mu}{2} \|X - X' - E\|_F^2 \tag{3}$$

其中 Y 为拉格朗日乘子。此处省略后续的推导过程和收敛性分析，求解该优化问题的交替求解算法中的 python 代码片段如下：

```
1  ...
2  Xk = np.zeros(self.X.shape)
3  Ek = np.zeros(self.X.shape)
4  Yk = np.zeros(self.X.shape)
5  while (err > _tol) and iter_ < max_iter:
6      Xk = self.nuclear_prox(self.X - Ek + self.mu_inv * Yk, self.mu_inv)
7      Ek = self.L1_prox(self.X - Xk + self.mu_inv * Yk, self.mu_inv * self.lmbda)
8      Yk = Yk + self.mu * (self.X - Xk - Ek)
9      err = self.frobenius_norm(self.X - Xk - Ek)
10     iter_ += 1
11     if (iter_ % iter_print) == 0 or iter_ == 1 or iter_ > max_iter or err <= _tol:
12         print('iteration: {0}, error: {1}'.format(iter_, err))
13 ...
```

1. (4 points) Robust PCA 增加的变量 E 和正则项对模型有什么作用？
2. (14 points) 代码片段中第 6 行和第 7 行调用的方法实现了什么优化方法解了哪两个优化问题？（写出优化方法并分别写出优化问题）
3. (12 points) 你认为 Robust PCA 具有哪些实际应用场景？在这些应用场景中有什么优势？（举出三个具体例子，并简要说明在这些场景中的优势，多于三个批改时以前三个为准）

解：

1. [1]$E = X - X'$ 可以看成是真实数据的特征矩阵和对应的低秩近似矩阵在目标函数中加入了对 E 的 0 范数的最小化限制，优化得到的 E 是稀疏的 (噪声)，各元素取值 0 居多，可以认为保持了 X 和 X' 的相似性

   [2] 在目标函数中加入了对 $rank(X')$ 的最小化限制，得到的 X' 是 X 的低秩表示。

   综上，目标函数保证了 X' 是对 X 进行低秩表示，同时由于 E 的存在，可以把噪声或异常值压缩到较少的非零元素，可以保证在降低维度（降秩）的同时，使得 X 和 X' 相差不大，也即矩阵各个特征的含义不会发生很大的改变。

2. [1] **优化方法**

   step1: 总体上，代码第 6 和第 7 行在思想上使用了优化中的多变量交替优化，固定其他变量而着重优化一个。以下使用 L 表示优化问题中的目标函数,m 为当前的迭代轮数

$$\begin{aligned}\min_{X',E} \quad & L(X', E, Y, \mu) \\ (X')^{m+1} &= \arg\min_{X'} L(X', E^m, Y^m, \mu) \\ E^{m+1} &= \arg\min_{E} L((X')^{m+1}, E, Y^m, \mu)\end{aligned}$$

   step2: 每一步，对每一个变量的优化过程中，使用近端梯度下降 PGD 来解决核范数、L1 范数不可微的优化问题。具体解释在下面优化问题的推导给出

   [2] **优化问题**

   step1: 对 X' 的优化

$$(X')^{m+1} = \arg\min_{X'} L(X', E^m, Y^m, \mu)$$

   以下假设当前已经迭代了 m 轮，各参数已经更新成 X_k^m, E_k^m, Y_k^m，以下推导针对正在进行第 m+1 轮的优化给出近端梯度下降 PGD 的具体表达式（也可以使用软阈值函数进行解释）

$$\begin{aligned}X_k^{m+1} &= \arg\min_{X'} L(X', E_k^m, Y_k^m, \mu) \\ &= \arg\min_{X'} \|X'\|_* + < Y_k^m, X - X' - E_k^m > + \frac{1}{2}\|X - X' - E_k^m\| \\ &= \arg\min_{X'} \|X'\|_* + tr((Y_k^m)^T \cdot (-X')) + \frac{1}{2}\|X - X' - E_k^m\| \\ &= \arg\min_{X'} \|X'\|_* + tr((Y_k^m)^T \cdot (-X')) + \frac{1}{2}\|X - X' - E_k^m\| \\ &= \arg\min_{X'} \|X'\|_* + tr((Y_k^m)^T \cdot (-X')) + \frac{1}{2} tr\left(I^T \cdot (X - X' - E_k^m) \cdot (X - X' - E_k^m)^T \cdot I\right) \\ &= \arg\min_{X'} \|X'\|_* + \frac{\mu}{2} \cdot tr\left(-\frac{2Y_k^m X'}{\mu} + (X - X' - E_k^m)(X - X' - E_k^m)^T\right) \\ &= \arg\min_{X'} \|X'\|_* + \frac{\mu}{2} \cdot \\ & tr\left((X - X' - E_k^m)(X - X' - E_k^m)^T + (X - X' - E_k^m) \cdot \frac{(Y_k^m)^T}{\mu}(X - X' - E_k^m)^T \cdot \frac{(Y_k^m)}{\mu} + \frac{Y_k^m (Y_k^m)^T}{\mu^2}\right) \\ &= \arg\min_{X'} \|X'\|_* + \frac{\mu}{2} \cdot \left\|X - X' - E_k^m + \frac{Y_k^m}{\mu}\right\|_F^2\end{aligned}$$

   综上，由于优化函数后项（F 范数的平方）二阶可导，交替优化中使用 PGD 优化 X_k 的最终优化问题为

$$X_k^{m+1} = \arg\min \|X'\|_* + \frac{\mu}{2} \cdot \left\|X - X' - E_k^m + \frac{Y_k^m}{\mu}\right\|_F^2$$

所以本代码第六行可以调用近端梯度下降 PGD 的库函数

$$X_k^{m+1} = Prox_{\|\cdot\|_1,\frac{1}{\mu}}(X - E_k^m + \frac{Y_k^m}{\mu}) = \arg\min \|X'\|_* + \frac{\mu}{2} \cdot \left\| X - X' - E_k^m + \frac{Y_k^m}{\mu} \right\|_F^2$$

step2: 对 E 的优化

同对 X' 的优化，可以得到对 E 的 PGD 优化问题为

$$\begin{aligned} E_k^{m+1} &= Prox_{\|\cdot\|_1,\frac{\lambda}{\mu}}(X - X_k^{m+1} + \frac{Y_k^m}{\mu}) \\ &= \arg\min_E \lambda \cdot \|E\|_1 + \frac{\mu}{2} \left\| X - X_k^m - E - \frac{Y_k^m}{\mu} \right\|_F^2 \end{aligned}$$

3. [1] **图像去噪（去除 E 的角度）**

   图片中大部分内容相似（如背景信息等），图片对应的特征矩阵对应图片中包含的信息丰富程度。我们认为高质量的图片应该是低秩的。如果秩很高，则认为图像中含有较多的噪声，同时噪声也可以认为是稀疏的。

   优势：

   RPCA 在保持特征矩阵的含义不发生很大的改变的情况下（X'），可以去除稀疏的噪声矩阵（E）。这是传统 PCA 在存在噪声中所做不到的

   [2] **文件压缩（保留 X' 的角度）**

   文件压缩任务通常需要做文件中重要信息（特征）的抽取，而不需要一些相关性低的信息。

   优势：

   此时，RPCA 也继续保持在图像去噪任务中的优势。此外，在压缩任务中，由于压缩的程度不同，RPCA 中 E 的稀疏程度可以自定义调整。同时，RPCA 加入了 L2 范数的乘法项，加入后当前优化问题结果和只有拉格朗日函数的最优解相同，而且优化过程更加平滑，使得最终求解（降维后的 X'）更加精准。尽量保持原始文件语义在压缩任务中是至关重要的

   [3] **异常检测、特征预测（保留 E 和 X' 的角度）**

   异常检测任务需要首先识别到正常样本的特征（RPCA 中的 X'），同时需要提取出异常样本（RPCA 中的 E）

   优势：

   相比较于 PCA，RPCA 可以显示提取出异常点特征矩阵（噪声矩阵）。E 的存在可以分析异常值点 X' 的存在可以认为是去除掉无关信息后的重要特征，可用于分析数据中关系和潜在的变化趋势

二. (30 points) **图半监督学习**

多标记图半监督学习算法的正则化框架如下 (另见西瓜书 p303)。

$$\mathcal{Q}(F) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{d_i}} F_i - \frac{1}{\sqrt{d_j}} F_j \right\|^2\right) + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2 \tag{4}$$

1. (15 points) 求正则化框架的最优解 F^*。
2. (15 points) 试说明该正则化框架与书中 p303 页多分类标记传播算法之间的关系。

解：

1. 本题的两小问关系紧密，二者的部分关系会在第一问正则化框架最优解求解的过程中涉及。正则化框架最优值求解可以分为两步进行，首先先推导出迭代解析解（通项），再假定可以无限迭代进而得到最优解

   正则化框架 Q（F）是关于 F 的凸函数，而且是无约束优化。故可以直接求梯度求解。此外，由于 Q（F）的呈现形式是以 F_i 为单位表示，故一下求解也以 F_i 为单位进行优化

   以下先进行符号约定

$$M = \frac{1}{2} \cdot \left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{d_i}} F_i - \frac{1}{\sqrt{d_j}} F_j \right\|^2 \right)$$

$$N = \mu \cdot \sum_{i=1}^{n} \|F_i - Y_i\|^2$$

   所以可以进行优化问题的转化：

$$\frac{\partial Q(F)}{\partial F_k} = 0 \Leftrightarrow \frac{\partial M}{\partial F_k} + \frac{\partial N}{\partial F_k} = 0$$

   step1: 计算 $\frac{\partial M}{\partial F_k}$

$$\begin{aligned}
\frac{\partial M}{\partial F_k} &= \frac{\partial}{\partial F_k} \frac{1}{2} \left(\sum_{j \neq k}^{n} W_{kj} \cdot \left\| \frac{1}{\sqrt{d_k}} F_k - \frac{1}{\sqrt{d_j}} F_j \right\|^2 + \sum_{i \neq k}^{n} W_{ik} \left\| \frac{1}{\sqrt{d_i}} F_i - \frac{1}{\sqrt{d_k}} F_k \right\|^2 \right) \\
&= \frac{\partial}{\partial F_k} \frac{1}{2} \left(\sum_{j \neq k}^{n} W_{kj} \cdot \left\| \frac{1}{\sqrt{d_k}} F_k - \frac{1}{\sqrt{d_j}} F_j \right\|^2 + \sum_{j \neq k}^{n} W_{kj} \left\| \frac{1}{\sqrt{d_k}} F_k - \frac{1}{\sqrt{d_j}} F_j \right\|^2 \right) \\
&= \frac{\partial}{\partial F_k} \left(\sum_{j \neq k}^{n} W_{kj} \cdot \left\| \frac{1}{\sqrt{d_k}} F_k - \frac{1}{\sqrt{d_j}} F_j \right\|^2 \right) \\
&= \sum_{j=1, j \neq k}^{n} W_{kj} \frac{\partial}{\partial F_k} \left((\frac{1}{\sqrt{d_k}} F_k - \frac{1}{d_j} F_j)^T (\frac{1}{\sqrt{d_k}} F_k - \frac{1}{d_j} F_j) \right) \\
&= \sum_{j=1, j \neq k}^{n} W_{kj} \left(2 \frac{1}{d_k} F_k - 2 \frac{1}{\sqrt{d_k}} \frac{1}{\sqrt{d_j}} F_j \right) \\
&= \sum_{j=1}^{n} W_{kj} \left(2 \frac{1}{d_k} F_k - 2 \frac{1}{\sqrt{d_k}} \frac{1}{\sqrt{d_j}} F_j \right) \\
&= 2 \sum_{j=1}^{n} W_{kj} \frac{F_k}{d_k} - 2 \sum_{j=1}^{n} W_{kj} \frac{1}{\sqrt{d_k d_j}} F_j \\
&= 2F_k - 2 \sum_{j=1}^{n} W_{kj} \frac{F_j}{\sqrt{d_k d_j}}
\end{aligned}$$

step2: 计算 $\frac{\partial N}{\partial F_k}$

$$\begin{aligned}\frac{\partial N}{\partial F_K} &= \frac{\partial}{\partial F_K}\left(\mu\sum_{i=1}^{n}\|F_i - Y_i\|^2\right)\\ &= \frac{\partial}{\partial F_K}\left(\mu\sum_{i=1}^{n}(F_i - Y_i)^T(F_i - Y_i)\right)\\ &= \frac{\partial}{\partial F_K}\left(\mu(F_k - Y_k)^T(F_k - Y_k)\right)\\ &= 2\mu(F_k - Y_k)\end{aligned}$$

step3: 综合以上两步

$$\frac{\partial Q(F)}{\partial F_k} = 2F_k - 2\sum_{j=1}^{n} W_{kj}\frac{F_j}{\sqrt{d_k d_j}} + 2\mu(F_k - Y_k) = 0$$

经过化简可以得到迭代形式，此时等号左侧可以看成是 m+1 次迭代后结果，右侧可以看成是第 m 次迭代后结果

$$F_k = \frac{1}{\mu+1}\left(\sum_{j=1}^{n} W_{kj}\frac{F_j}{\sqrt{d_k d_j}} + \mu Y_k\right)$$

由于 F_k, F_j, Y_k 均为 n*1 的向量，其中 $j \in [n]$

可以令 $S_k^j = \frac{1}{\sqrt{d_k}W_{kj}\sqrt{d_j}}$，是“标记传播矩阵 S”中的其中一个元素

此时迭代式可以化成

$$F_k = \frac{1}{\mu+1}\left(\sum_{j=1}^{n} S_k^j F_j + \mu Y_k\right)$$

进一步定义完整的“标记传播矩阵”S,

$$\begin{aligned}S = \begin{pmatrix} S_1^1 & \cdots & S_1^n \\ \vdots & & \vdots \\ S_n^1 & \cdots & S_n^n \end{pmatrix} &= \begin{pmatrix} \frac{1}{\sqrt{d_1}}W_{1,1}\frac{1}{\sqrt{d_1}} & & \frac{1}{\sqrt{d_1}}W_{1,n}\frac{n}{\sqrt{d_n}} \\ & & \\ \frac{1}{\sqrt{d_n}}W_{n,1}\frac{1}{\sqrt{d_1}} & & \frac{1}{\sqrt{d_n}}W_{n,n}\frac{1}{\sqrt{d_n}} \end{pmatrix}\\ &= \begin{pmatrix} \frac{1}{\sqrt{d_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{d_n}} \end{pmatrix} \cdot W \cdot \begin{pmatrix} \frac{1}{\sqrt{d_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{d_n}} \end{pmatrix}\end{aligned}$$

令 $D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}$, 则 $S = D^{-\frac{1}{2}} \cdot W \cdot D^{-\frac{1}{2}}$, 所以得到此时的迭代更新公式为 (已迭代 t 轮，正在进行第 t+1 轮参数更新)

$$F^{(t+1)} = \frac{1}{1+\mu} \cdot S \cdot F^{(t)} + \frac{\mu}{1+\mu} \cdot Y$$

可以通过数学归纳法得到迭代最终收敛（分别令 t=0,1,2,...,n）。所以正则化框架的最优解 F^* 如下所示

$$F^* = \lim_{x\to\infty} F^{(t)} = \frac{\mu}{1+\mu}(I - \frac{1}{\mu+1}S)^{-1}Y$$

2. **[1] 相似性：**

   - 正则化框架和多分类标记传播框架都基于半监督学习共同的学习假设：相似样本有相似标记

   - 由第一问的正则化框架简化迭代式可知，若令 $\mu = \frac{1-\alpha}{\alpha}$，正则化框架最优解和多分类标记传播相同

   [2] 不同点：

   - 正则化框架中考虑的是离散输出值，多分类标记传播的能量函数 E(f) 考虑连续输出值 $(f^T \cdot (D - W) \cdot f)$

三. (40 points) **半监督 SVM 实践**

参照教材中图 13.4 所示的 TSVM 算法，在所提供的半监督数据集上进行训练，报告模型在未标记数据集以及测试集上的性能。

本次实验的数据集为一个二分类的数据集，已提前划分为训练数据和测试数据，其中训练数据划分为有标记数据和无标记数据。数据的特征维度为 30，每一维均为数值类型。数据文件的具体描述如下:

- `label_X.csv,label_y.csv` 分别是有标记数据的特征及其标签。
- `unlabel_X.csv,unlabel_y.csv` 分别是无标记数据的特征及其标签。
- `test_X.csv,test_y.csv` 分别是测试数据的特征及其标签。

注意，训练阶段只可以使用 `label_X.csv,label_y.csv,unlabel_X.csv` 中的数据，其他的数据只可以在测试阶段使用。

1. 本次实验要求使用 Python3.8 以上编写，代码统一集中在 `tsvm_main.py` 中，通过运行该文件就可以完成训练和测试，并输出测试结果。(提交一个额外的 python 文件)
2. 本次实验需要完成以下功能:
   1. (15 points) 参照教材中图 13.4，使用代码实现 TSVM 算法。要求:
      1. 不允许直接调用相关软件包中的半监督学习方法。
      2. 可以直接调用相关软件包的 SVM 算法。
      3. 可以使用诸如 `cvxpy` 等软件包求解 QP 问题。
   2. (10 points) 使用训练好的模型在无标记数据和测试数据上进行预测，报告模型在这两批数据上的准确率和 ROC 曲线以及 AUC 值。
   3. (15 points) 尝试使用各种方法提升模型在测试集上的性能，例如数据预处理，超参数调节等。报告你所采取的措施，以及其所带来的提升。

解：

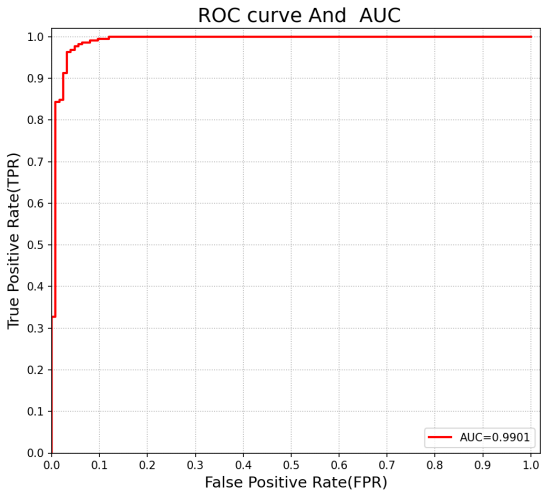
```
step1：利用X_l和y_l训练一个SVM

step2：使用此SVM进行标记指派。对所有未标记样本都进行

step3：给一个初始化C_u和C_l，二者远小于即可

step4：迭代1：直到C_u和C_l很接近

    基于有标记+标记、无标记+指派、C_u和C_l重新训练得到新的划分超平面和松弛向量

    迭代2：直到本轮中所有样本都不满足交换条件

        找到无标记的指派中很有可能发生错误的样本，交换标记

        基于有标记+标记、无标记+指派、C_u和C_l重新训练得到新的划分超平面和松弛向量

    更新 C_u,C_u = min{2 C_u,C_l}
```

1. 略，按照上述伪代码进行编程
2. 以下报告的测试结果为数据预处理、超参数调节后的最佳结果。具体调节方式将在第三小问中给出。

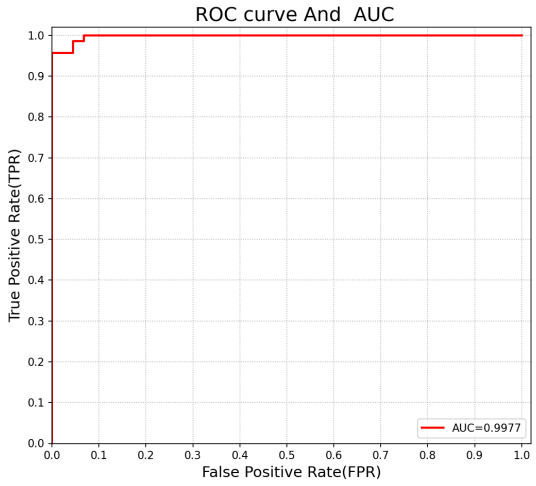
数据类别	准确率 P	AUC 值
未标记数据	96.491%	0.9901
测试数据	96.460%	0.9977

为了更好使 ROC 曲线符合真实预测对应概率值，以下两曲线均为额外生成 SVM 模型输出概率值得到。(不采用直推学习的理解，直推学习的理解在代码中给出（predict 方法)。经过实验，二者在准确率，AUC 值差别并不是很大)

本图为模型在无标记数据中的 ROC 曲线



本图为模型在测试数据中的 ROC 曲线



3.

不进行任何数据预处理、超参数调节时各性能指标如下：

	数据类别	准确率 P	AUC 值
原始数据	未标记数据	93.859%	0.9752
	测试数据	94.690%	0.9835

step1: 数据预处理

由于 SVM 是基于距离的计算，初始数据分布会很大程度影响模型训练效果，对特征之间取值范围非常敏感（如大数吃掉小数）大多数机器学习算法中，会选择 StandardScaler（标准化）来进行特征缩放，因为 MinMaxScaler（归一化）对异常值非常敏感。

以下对两种数据预处理方法均进行实验，分别对输入数据的特征矩阵归一化、标准化、归一化 + 标准化。

	数据类别	准确率 P	AUC 值
归一化	未标记数据	95.321%	0.9877
	测试数据	96.460%	0.9967

	数据类别	准确率 P	AUC 值
标准化	未标记数据	96.491%	0.9901
	测试数据	96.460%	0.9977

	数据类别	准确率 P	AUC 值
归一化 + 标准化	未标记数据	95.321%	0.9877
	测试数据	96.460%	0.9967

综上，最终应对输入数据进行标准化后，再进行后续超参数调节

step2：超参数调节

从优化的目标函数及代码中可以看到，可以优化的超参数主要有以下几个：

1. C_u, C_l 的绝对值，和他们的相对比值：

$\frac{1}{2}\|w\|_2^2$ 约为 2.325934813674294, 相对于 ξ_1 和 ξ_2 的加和（二者都约为-1e-10）, 差了 10^{10} 数量级

方法：不断修改二者相差的数量级 (通过修改 C_u 和 C_l 的值来完成)，在目标函数中后两项分别乘 $10^5 - -10^9$，精度、AUC 均无变化; 同时不断修改二者的相对比例，从两者比例 2 倍-10 倍，进行实验，精度、AUC 均无变化

2. 类别不平衡问题：先看看有没有严重的不平衡问题，有的话可以改进

方法：检测每次遍历完整数据集更新类别后正类父类的比值，负类：正类均基本稳定在 0.6，没有类别不平衡

3. SVC 中的参数

方法：尝试多项式核、高斯核，还是线性核最好，因为数据集基本是标准二分类问题。在线性核中调整了常数 C，但对最终性能几乎没有影响