Subject: Optimization for Data Analysis, Section 1, Introduction

Date: from May 13, 2025 to May 16, 2025

Contents

Introduction

Data Analysis and Optimization

General Framework for Optimization Problem in Data Analysis

Data set: 数据分析中的典型任务是找到一个模型，使其与收集到的数据(称为数据集, **data set**)表现一致同时遵守一些结构约束. 一个包含 m 个对象的数据集可以表示为

$$\mathcal{D} := \{(x_j, y_j), j = 1, 2, \ldots, m\}, \tag{1}$$

其中 x_j 为向量/矩阵，称之为特征(**feature**)，y_j 本书中一般考虑为标量情形，称之为观察(observation) 或标签(**label**).

Train: 模型与数据表现一致是指寻找映射/函数 ϕ 使得对于大多数 $j = 1, 2, \cdots, m$ 都有 $\phi(x_j) \approx y_j$，这一寻找的过程被称为学习(learning)或训练(training).

Parametrization: 函数 ϕ 往往能够被向量 θ 或 矩阵 Θ 参数化(**parametrization**)，确定 ϕ 的问题就转变为数据拟合问题：在特定参数化模式下寻找最优的(**optimal**)参数 θ 使得 $\phi(x_j) \approx y_j, j = 1, 2, \ldots, m$.

Objective function: 参数化后得到了"最优"的概念，这样就能够将原问题转化为优化问题，即最小化如下目标函数(**objective function**)：

$$\mathcal{L}_{\mathcal{D}}(\theta) := \frac{1}{m} \sum_{j=1}^{m} \ell(x_j, y_j; \theta), \tag{2}$$

其中 $l(\cdot)$ 反映了 $\phi(a)$ 与 lable y 的差异，因此目标函数度量了参数向量等于 θ 时数据集 $\mathcal{D}$ 上累积的平均损失.

Framework: 总结上述内容，数据分析任务转化为优化问题的思路如下：

采集数据(**data**)→参数化模型→寻找"最优参数"→定义目标函数→进行优化(**optimization**)

Different Tasks in Data Analysis

Tasks related to label 数据分析任务的不同可以通过标签 y_j 的差异体现：

- y_j 为实数：经典的回归问题(**regression**)
- y_j 为指标(index)($\{1, \cdots, N\}$)：N-分类问题(**classification**)
- y_j 根本不存在，数据集只包含特征 x_j：同时学习出标签 y_j 和映射 ϕ，例如聚类问题(**clustering**)

Tasks related to data 数据分析任务中，数据清洗后仍然有许多复杂问题需要解决：

- **Noise:** 获取的数据 (x_j, y_j) 可能包含噪声，这需要我们的模型具有稳定性(**robust**)
- **Missing data:** 获取的数据可能缺失部分标签/特征
- **Streaming:** 数据获取是持续的而非一次全部获取，这要求构建在线学习范式(**online**)

Overfitting 一个任务的真实数据是无限的，由于我们仅是在其子集 $\mathcal{D}$ 上训练模型，因此我们希望映射 ϕ 在观察到的子集 $\mathcal{D}$ 和未观察到的数据点上都表现良好. 若模型对 $\mathcal{D}$ 太敏感则会导致过拟合(**overfitting**)，即在 $\mathcal{D}$ 上表现很好但在未观察到的数据点上表现很差.

Regularization 解决 overfitting 的有效方法是正则化(**regularization**)，将模型(2)变为:

$$\min_{\theta\in\Omega} \mathcal{L}_{\mathcal{D}}(\theta) + \lambda \,\mathrm{pen}(\theta), \tag{3}$$

其中 Ω 为参数 θ 的可行集合，$\mathrm{pen}(\cdot)$ 为正则化函数(regularization function / regularizer)，$\lambda \geqslant 0$ 为正则化参数(regularization parameter)，用于平衡训练数据 $\mathcal{D}$ 的拟合程度与模型复杂度：较小的 λ 倾向更准确拟合数据集 $\mathcal{D}$，更大的 λ 倾向降低模型复杂度.

Several data analysis problems

Least Squares

最经典的数据分析问题应该就是(线性)最小二乘问题(**(Linear) Least Squares, LS**).
Data $(x_j, y_j) \in \mathbb{R}^n \times \mathbb{R}$, $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^m$
Model 线性模型 $\phi(a) = \theta^T a$，若要显式写出截距就是 $\phi(a) = \theta^T a + \beta, \beta \in \mathbb{R}$.
Objective function

$$\mathcal{L}_{\mathcal{D}}(\theta) = \frac{1}{2m}\sum_{j=1}^{m}\left(x_j^T\theta - y_j\right)^2 = \frac{1}{2m}\|A\theta - y\|_2^2, \tag{4}$$

其中 $A = (a_1, \cdots, a_n)^T$，$y = (y_1, y_2, \cdots, y_m)^T$，目标是最小化 $\mathcal{L}_{\mathcal{D}}(\theta)$.
Ridge regression 使用 l_2-norm 进行正则化，得到岭回归:

$$\min_{\theta} \frac{1}{2m}\|A\theta - y\|_2^2 + \lambda\|\theta\|_2^2, \quad \lambda > 0. \tag{5}$$

岭回归得到的解对于数据 (x_j, y_j) 的扰动(**perturbation**) 不敏感.
LASSO 使用 l_1-norm 进行正则化，得到 LASSO:

$$\min_{\theta} \frac{1}{2m}\|A\theta - y\|_2^2 + \lambda\|\theta\|_1. \tag{6}$$

LASSO 倾向于产生稀疏的解 θ，即 θ 中非零元素较少Tibshirani (1996)，因此可用于特征选择，即只需要使用 θ 中非零元素位置所对应数据 x_j 中的特征就可以很好地进行预测任务.
LASSO property l_1-norm 有非光滑性(**non-smooth**) 和凸性(**convex**)，其结构简单可能更容易被算法利用从而求解.

Matrix Factorization Problems

数据分析中也有一些任务要求从稀疏的观测数据中估计低秩矩阵(**low-rank matrix**)，其可被视为 LS 问题的自然推广.
Data $(x_j, y_j) \in \mathbb{R}^{n\times p} \times \mathbb{R}$, $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^m$
Objective function

$$\mathcal{L}_{\mathcal{D}}(\Theta) = \frac{1}{2m}\sum_{j=1}^{m}(\langle x_j, \Theta\rangle - y_j)^2, \tag{7}$$

其中 $\langle A, B\rangle := \text{trace}(A^T B)$. 可将 x_j 看作未知矩阵 Θ 的"探测"(**probing**).

Regularized version 1 为得到低秩矩阵，可以通过限制参数矩阵 X 的范数大小：

$$\min_{\Theta} \frac{1}{2m} \sum_{j=1}^{m} (\langle x_j, \Theta\rangle - y_j)^2 + \lambda \|\Theta\|_*, \tag{8}$$

其中 $\langle A, B\rangle = \text{trace}(A^T B)$， $\|\Theta\|_*$ 为核范数(nuclear norm).

Note 1. 核范数 $\|\cdot\|_*$ 与 l_1-*norm* 作用相似，核范数诱导低秩矩阵， l_1-*norm* 诱导稀疏向量.

Property

1. 虽然核范数非光滑，但凸，因此目标函数是凸的.
2. 式(8) 被证明在 Θ 低秩、观察矩阵 A 满足 restricted isometry property（一些随机矩阵可以满足）时，能够产生统计意义上的有效解.
3. 当 Θ 是 **incoherent**(没有几个较于其他元素特别大的元素)，x_j 只有单元素时，也可以产生有效解Candes and Recht (2012).

Regularized version 2 另一种正则化方式是令 $\Theta = LR^T$，其中 $L \in \mathbb{R}^{n\times r}, R \in \mathbb{R}^{p\times r}, r \ll \min(n, p)$:

$$\min_{L,R} \frac{1}{2m} \sum_{j=1}^{m} (\langle x_j, LR^T\rangle - y_j)^2. \tag{9}$$

由于 r 需要 Θ 强制满足，因此无需正则项. 并且 (L, R) 中的元素总数为 $(n + p)r \ll np$.

Property 将式(9)的函数考虑为 (L, R) 的函数时是非凸的，但已证明在对数据进行特定假设和使用特定算法时，也能够得到好的解. 原因简单来说是上述形式是对一个可解决问题近似：若对 Θ 有完全的观测，那么就可以对其做奇异值分解.

Nonnegative matrix factorization 有一些在计算机视觉中的任务要求式(9)中的矩阵 L, R 所有元素非负，若观测矩阵 $Y \in \mathbb{R}^{n\times p}$ 完全已知，那么就得到非负矩阵分解问题：

$$\min_{L,R} \|LR^T - Y\|_F^2, \quad \text{subject to } L \geq 0, R \geq 0. \tag{10}$$

Support Vector Machines

支持向量机(**support vector machines, SVM**)是机器学习中的经典问题，起源于 1960s.

Data $(x_j, y_j), x_j \in \mathbb{R}^n, y_j \in \{-1, 1\}$, $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^m$

Target SVM 寻找向量 $\theta \in \mathbb{R}^n, \beta \in \mathbb{R}$ 能够将数据 (x_j, y_j) 区分开：

$$\begin{aligned} \theta^T x_j - \beta &\geqslant 1 \quad \text{when } y_j = +1, \\ \theta^T x_j - \beta &\leqslant -1 \quad \text{when } y_j = -1. \end{aligned} \tag{11}$$

Separating hyperplane 参数对 (θ, β) 所定义的超平面 $y = \theta^T x$ 被称为分割超平面(**separating hyperplane**). 在所有的分割超平面中，$\|\theta\|^2$ 最小的超平面可以最大化两类别的边距(**margin**)，即到任一类的最近点 x_j 的距离最大.

Objective function

$$H(\theta, \beta) = \frac{1}{m} \sum_{j=1}^{m} \max\left(1 - y_j\left(\theta^T x_j - \beta\right), 0\right). \tag{12}$$

Regularized version SVM 的正则化中包含非光滑的损失函数(12)和光滑的正则项 $\|\theta\|_2^2$:

$$H(\theta,\beta)=\frac{1}{m}\sum_{j=1}^{m}\max(1-y_j(\theta^T x_j-\beta),0)+\frac{1}{2}\lambda\|\theta\|_2^2. \tag{13}$$

Property 当 λ 足够小、分离超平面存在时，最小化式(13)的参数对 (θ,β) 就是最大边距(**maximum-margin**)的分离超平面.

Nonlinear SVM 有时线性的模型 $\theta^T x$ 并不足以将数据很好地分开，此时就可以引入非线性变化 $\psi(x_j), j=1,\cdots,m$，使模型有更强的表达能力，此时式(11)(13)分别变为：

$$\begin{aligned}&\theta^T\psi(x_j)-\beta\geqslant 1 \quad \text{when } y_j=+1;\\ &\theta^T\psi(x_j)-\beta\leqslant -1 \quad \text{when } y_j=-1,\end{aligned} \tag{14a}$$

$$H(\theta,\beta)=\frac{1}{m}\sum_{j=1}^{m}\max(1-y_j(\theta^T\psi(x_j)-\beta),0)+\frac{1}{2}\lambda\|\theta\|_2^2. \tag{14b}$$

Property SVM 可以自然地表示为凸集上的最小化问题：通过引入人工变量可以将损失函数(8)(9)转化为凸二次规划[a]问题. 通过取对偶(**dual**)可以得到：

$$\min_{\alpha\in\mathbb{R}^m}\frac{1}{2}\alpha^T Q\alpha-1^T\alpha \quad \text{subject to } 0\leqslant\alpha\leqslant\frac{1}{\lambda}\mathbf{1},\quad y^T\alpha=0, \tag{15}$$

其中 $Q_{kl}=y_k y_l\psi(x_k)^T\psi(x_l),\quad y=(y_1,y_2,\ldots,y_m)^T,\quad \mathbf{1}=(1,1,\ldots,1)^T$.

Kernel trick 式(15)中实际上无需知道映射 ψ 的具体形式，只需要通过核函数(**kernel function**) $K:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ 定义 Q 即可，使用 $K(x_k,x_l)$ 代替 $\psi(x_k)^T\psi(x_l)$，这被称为核技巧(**kernel trick**). 常见的核函数为高斯核(**Gaussian kernel**):

$$K(x_k,x_l):=\exp\left(-\frac{1}{2\sigma}\|x_k-x_l\|^2\right),\sigma>0. \tag{16}$$

[a] 凸二次目标函数+线性的限制条件

Logistic Regression

逻辑回归(**logistic regression**) 可以视作二分类-SVM 的拓展，其并不是返回类别，而是返回所属于每个类别的概率(**odds**).

Target 寻找概率计算函数 $p(x;\theta)=\frac{1}{1+\exp(\theta^T x)}$，由 $\theta\in\mathbb{R}^n$ 参数化，计算数据 x 属于某一类别的概率.

2-classification 在二分类中期望

$$\begin{aligned}&p(x_j;\theta)\approx 1 \quad \text{when } y_j=+1;\\ &p(x_j;\theta)\approx 0 \quad \text{when } y_j=-1,\end{aligned} \tag{17}$$

通过取负对数似然定义损失函数为

$$L(\theta):=-\frac{1}{m}\left[\sum_{j:y_j=-1}\log(1-p(x_j;\theta))+\sum_{j:y_j=1}\log p(x_j;\theta)\right](+\lambda\|\theta\|_1). \tag{18}$$

Muti-classification 多分类是二分类任务的拓展，注意标签 $y\in\mathbb{R}^d$，当其属于第 k 个

类别时 $y_k = 1$，其余分量均为0. 我们期望

$$y_{jk} = \begin{cases} 1 & \text{when } x_j \text{ belongs to class } k, \\ 0 & \text{otherwise.} \end{cases} \tag{19a}$$

$$\begin{aligned} p_k(x_j;\theta) &\approx 1 \quad \text{when } \ y_{jk} = 1; \\ p_k(x_j;\theta) &\approx 0 \quad \text{when } \ y_{jk} = 0. \end{aligned} \tag{19b}$$

同样地可以定义其损失函数为

$$L(\Theta) := -\frac{1}{m}\sum_{j=1}^{m}\left[\sum_{\ell=1}^{M} y_{j\ell}(\theta_{[\ell]}^T x_j) - \log\left(\sum_{\ell=1}^{M}\exp(\theta_{[\ell]}^T x_j)\right)\right]. \tag{20}$$

Deep Learning

暂略，可见原文. 由于原文只做简单介绍引入，故此处略过.

Summary

可以看到本章中不同数据分析问题根据框架(3)得到的具体形式不尽相同，也有不同的性质，但它们有如下共性：

- 目标函数都包含两部分：损失项(**loss term**) + 正则化项(**regularization term**)
- 连续性：连续性(**continuous**)保证算法根据在访问附近点时获得的知识对当前点的函数行为进行良好的推断(**inference**)
- 求和性：都是对逐个数据点进行求和汇总

Note 2. *Worst-case complexity guarantees are only a piece of the story here, and understanding the various parameters and heuristics that form part of any practical algorithmic strategy are critical for building reliable solvers.*

Last updated: 16 May, 2025

References

Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.