Notes of “Algorithms for Big Data Analysis”

Gary757483617

May 2021

目录

1 线性规划与半定规划 **3**

1.1 线性规划 (LP) 3

1.2 半定规划 (SDP) 4

2 对偶理论 **5**

2.1 Lagrange 对偶方程 5

2.2 Lagrange 对偶问题 6

3 单纯形法与内点法 **7**

3.1 单纯形法 7

3.2 内点法 7

4 压缩感知与稀疏优化 **8**

4.1 压缩感知 8

4.2 稀疏优化算法 9

4.2.1 近似点梯度法 9

4.2.2 交替方向乘子法 (ADMM) 9

5 低秩矩阵恢复 **10**

5.1 协同过滤 10

5.2 隐变量模型 10

6 最优输运 **11**

7 随机优化 **11**

7.1 次梯度法 11

7.2 可变度量方法 12

7.3 近似矩阵乘积 12

8 数据降维 **13**

9 次模优化 **14**

9.1 次模 (submodularity) 定义 14

9.2 次模极大化 14

10 强化学习 **15**
10.1 马尔科夫决策过程 (MDP) 15
10.2 TD-learning 16
10.3 Q-learning 16
10.4 策略梯度下降 17
10.5 Actor-critic 方法 17

1 线性规划与半定规划

1.1 线性规划 (LP)

一般形式 (primal):

$$\begin{aligned} & min_x\ c^T x \\ & s.t.\ Ax = b, x \geq 0 \end{aligned} \tag{1.1}$$

对偶形式 (dual):

$$\begin{aligned} & max_{y,s}\ b^T y \\ & s.t.\ A^T y + s = c, s \geq 0 \end{aligned} \tag{1.2}$$

含绝对值的优化问题

可以将一般形式

$$\begin{aligned} & min_x\ \sum_i c_i |x_i|, \text{assume } c \geq 0 \\ & s.t.\ Ax \geq b \end{aligned} \tag{1.3}$$

转化为

$$\begin{aligned} & min_x\ \sum_i c_i z_i \\ & s.t.\ Ax \geq b, |x_i| \leq z_i. \end{aligned} \tag{1.4}$$

或定义

$$\begin{aligned} x_i &= x_i^+ - x_i^-, x_i^+, x_i^- \geq 0 \\ |x_i| &= x_i^+ + x_i^-. \end{aligned} \tag{1.5}$$

于是最优化问题等价于

$$\begin{aligned} & min\ \sum_i c_i \left(x_i^+ + x_i^-\right) \\ & s.t.\ A\left(x^+ - x^-\right) \geq b, x_i^+, x_i^- \geq 0. \end{aligned} \tag{1.6}$$

弱对偶性 (weak duality): 假设 x 是 P 的可行解 (1.1), (y, s) 是 D 问题的可行解 (1.2). 则

$$\begin{aligned} x^T s &= x^T(c - A^T y) \\ &= c^T x - b^T y \\ &= \text{duality gap} \\ &\geq 0 \end{aligned} \tag{1.7}$$

其中最后一行是由于 $x_i, s_i \geq 0$. 当等号成立时, 则 P 问题与 D 问题具有**强对偶性**. 此时成立

$$\begin{aligned} x^T s &= 0 \\ x_i s_i &= 0. \end{aligned} \tag{1.8}$$

利用互补条件可以在得到一个解的条件下求出另一个解. (e.g. 支持向量机) 联立 P 问题, D 问题和互补条件 1.8, 理论上可以求出最优解.

1.2 半定规划 (SDP)

半正定矩阵: X 的所有特征值非负, 或可分解为 $X = B^T B$. $X \succeq Y$ 表示对称矩阵 $X - Y$ 是半正定.

首先定义符号 $\langle X, Y\rangle = \sum_{ij} X_{ij}Y_{ij} = Tr(XY)$. 其中 X, Y 为对称矩阵. 简化起见, 先考虑单变量的 SDP. P 问题是

$$\begin{aligned} &min_X \ \langle C, X\rangle \\ &s.t. \ \langle \mathbf{A}, X\rangle = \mathbf{b}, \ X \succeq 0 \end{aligned} \tag{1.9}$$

其中 $\mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ ... \\ A_m \end{bmatrix}, \mathbf{b} = (b_1, b_2, ...b_m)^T$.

同时 D 问题是

$$\begin{aligned} &max_{y,S} \ b^T y \\ &s.t. \ \sum_i y_i A_i + S = C, \ S \succeq 0. \end{aligned} \tag{1.10}$$

SDP 举例: 特征值优化

考虑求最大特征值的最小值问题:

$$min \ \lambda_{max} \left[A_0 + \sum_i x_i A_i \right]. \tag{1.11}$$

可以将其表示为 SDP 问题:

$$\begin{aligned} &min \ z \\ &s.t. \ zI - \sum_i x_i A_i \succeq \ A_0. \end{aligned} \tag{1.12}$$

可以根据 $\langle X, Y\rangle = Tr(XY)$ 的关系得到这一结论.

考虑二次约束二次规划 (Quadratically Constrained Quadratic Programming, QCQP) 问题:

$$\begin{aligned} &min \ x^T A_0 x + 2b_0^T x + c_0 \\ &s.t. \ x^T A_i x + 2b_i^T x + c_i \leq 0 \end{aligned} \tag{1.13}$$

其中假设 $A_i \in \mathcal{S}^n$. 因为 $x^T A_i x = \langle A_i, xx^T\rangle$, 记 $xx^T = X$, 于是 1.13等价于

$$\begin{aligned} &min \ Tr\left(A_0 X\right) + 2b_0^T x + c_0 \\ &s.t. \ Tr\left(A_i X\right) + 2b_i^T x + c_i \leq 0. \end{aligned} \tag{1.14}$$

记

$$\begin{aligned} x^T A_i x + 2b_i^T x + c_i &= \langle \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}, \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \rangle \\ &= \langle \overline{A}_i, \overline{X} \rangle \end{aligned} \tag{1.15}$$

于是可以化为标准的 SDP 问题:

$$\begin{aligned} min\ Tr\left(\overline{A}_0\overline{X}\right) \\ s.t.\ Tr\left(\overline{A}_i\overline{X}\right) \leq 0 \\ \overline{X} \succeq 0. \end{aligned} \tag{1.16}$$

SDP 中的对偶性质: 类似于 1.7, 假设 X, S 均为半正定矩阵, 可以得到

$$\begin{aligned} \langle X, S\rangle &= \langle C, S\rangle - b^T y \\ &= Tr\left(XS^{1/2}S^{1/2}\right) \\ &= Tr\left(S^{1/2}XS^{1/2}\right) \geq 0. \end{aligned} \tag{1.17}$$

当满足取等条件 $\langle X, S\rangle = 0$ 时 P 问题与 D 问题具有强对偶性. 此时有 $XS = 0$(证明见课件).

2 对偶理论

此部分内容参考 Boyd: Convex Optimization.

2.1 Lagrange 对偶方程

标准优化问题

$$\begin{aligned} min\ f_0(x) \\ s.t.\ f_i(x) \leq 0, i = 1, ..., m, \\ h_i(x) = 0, i = 1, ..., p. \end{aligned} \tag{2.1}$$

由此可以定义 Lagrangian $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{2.2}$$

其中 λ_i 是不等式约束相关的乘子, ν_i 是等式约束相关的乘子.

定义 Lagrange 对偶方程 $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ 为 Lagrangian 关于 x 的最小值, 即

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x\in\mathcal{D}} L(x, \lambda, \nu) \\ &= \inf_{x\in\mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)\right). \end{aligned} \tag{2.3}$$

对偶方程给出了 2.2最优解 p^* 的下边界, 即

$$g(\lambda, \nu) \leq p^* \tag{2.4}$$

其中要求 $\lambda \succeq 0$. (证明见书 216 页). 为了得到 nontrivial 下界还需要 $g(\lambda, \nu) > -\infty$.

Lagrange 对偶方程举例

例 1: 线性最小二乘

原问题:

$$\begin{aligned} min\ x^Tx \\ s.t.\ Ax = b. \end{aligned} \tag{2.5}$$

没有不等式约束, 可以求出 Lagrangian $L(x, \nu) = x^Tx + \nu^T(Ax - b)$. 于是对偶方程 $g(\nu) = \inf_x L(x, \nu)$. 由于 $L(x, \nu)$ 是 x 的凸函数, 可以由 $\nabla_x L(x, \nu) = 0$ 得到原问题下界.

例 2: 线性规划标准形式

LP 标准形式:

$$\begin{aligned} min\ c^Tx \\ s.t. Ax = b,\ x \succeq 0. \end{aligned} \tag{2.6}$$

求出 Lagrangian

$$\begin{aligned} L(x, \lambda, \nu) &= c^Tx - \sum_{i=1}^{n} \lambda_i x_i + \nu^T(Ax - b) \\ &= -b^T\nu + (c + A^T\nu - \lambda)^Tx. \end{aligned} \tag{2.7}$$

于是对偶方程

$$\begin{aligned} g(\lambda, \nu) &= \inf_x\ L(x, \lambda, \nu) \\ &= -b^T\nu + \inf_x\ (c + A^T\nu - \lambda)^Tx \\ &= \begin{cases} -b^T\nu,\ A^T\nu - \lambda + c = 0 \\ -\infty,\ \text{otherwise} \end{cases} \end{aligned} \tag{2.8}$$

2.2 Lagrange 对偶问题

问题提出: 从 Lagrange 对偶方程中寻找最好的下界, 即 Lagrange 对偶问题:

$$\begin{aligned} max\ g(\lambda, \nu) \\ s.t.\ \lambda \succ 0. \end{aligned} \tag{2.9}$$

对于标准形式的 LP 问题, 可以将 2.8中的约束条件显性化, 即

$$\begin{aligned} max\ \ -b^T\nu \\ s.t.\ A^T\nu + c \succeq 0. \end{aligned} \tag{2.10}$$

同样得, 从不等式形式 LP 也可以逆推回标准形式.(对称性. 证明见书 225 页).

Slater 条件: 如果 P 问题 2.1是凸的, 则可表述为

$$\begin{aligned} min\ f_0(x) \\ s.t.\ f_i(x) \leq 0,\ i = 1, ..., m \\ Ax = b \end{aligned} \tag{2.11}$$

其中 f_i 为凸函数. 若存在 $x \in \mathrm{relint}\mathcal{D}$, 使得 $f_i(x) < 0$, 则强对偶性成立.

KKT 条件: 如果强对偶性成立且 x, λ, ν 为最优解, 则满足:

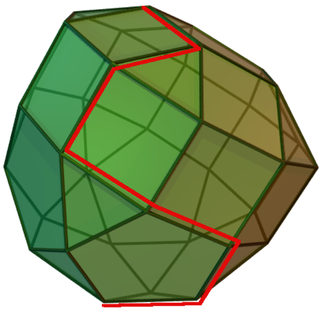
- P 问题限制: $f_i(x) \leq 0, h_i(x) = 0$
- D 问题限制: $\lambda \geq 0$
- 互补松弛性 (complementary slackness): $\lambda_i f_i(x) = 0$
- Lagrangian 梯度为零

3 单纯形法与内点法

3.1 单纯形法

(教材部分并未详细说明, 下面只简单描述)

对于标准形式的 LP 问题, 用一个多面体表示 $\{x|Ax \geq b\}$, 其中的顶点 (vertex) 被称为基础可行解 (basic feasible solution, BFS).



由于顶点数最多可达 $\mathcal{O}(2^n)$, 因此单纯形法是非多项式复杂度, 不适合大规模问题. 但目前单纯形法仍是主要使用的线性规划方法之一.

3.2 内点法

此部分内容参考 Boyd: Convex Optimization

考虑含不等式约束的基本凸优化问题 2.11, 假设最小值 $f_0(x^*) = p^*$. Newton 法可以用来解决含线性等式约束的二次优化, 内点法可以解决含线性不等式和等式约束的二次优化.

为了将不等式约束改为等式, 可以将 2.11变式为

$$\begin{gathered} min\ f_0(x) + \sum_{i=1}^{m} I_-(f_i(x)) \\ s.t.\ Ax = b \end{gathered} \tag{3.1}$$

其中

$$I_-(u) = \begin{cases} 0, & u \leq 0 \\ -\infty, & u > 0 \end{cases}. \tag{3.2}$$

为了使 $I_-(u)$ 二次可微, 选取近似函数

$$\hat{I}_-(u) = -\frac{1}{t}\log(-u),\ t>0 \tag{3.3}$$

这里的 $\hat{I}_-(u)$ 被称为 barrier 函数. 可以看到 t 增大, $\hat{I}_-(u)$ 越接近 $I_-(u)$, 但是其 Hessian 在可行域边界变化越快, Newton 法求解更困难.

代入后得到

$$\begin{aligned} min\ tf_0(x)+\phi(x) \\ s.t.\ Ax=b \end{aligned} \tag{3.4}$$

其中 $\phi(x) = -\sum_{i=1}^m \log(-f_i(x))$. 利用 KKT 条件, 可以求出内点轨迹 (central path):

$$\begin{aligned} & t\nabla f_0(x^*(t)) + \nabla\phi(x^*(t)) + A^T\hat{\nu} \\ & = t\nabla f_0(x^*(t)) + \sum_{i=1}^m \frac{1}{-f_i(x^*(t))}\nabla f_i(x^*(t)) + A^T\hat{\nu} \\ & = 0 \end{aligned} \tag{3.5}$$

其中 A^T 项来自于等式约束的微分.

含不等式约束的线性规划

考虑

$$\begin{aligned} min\ c^Tx \\ s.t.\ Ax \preceq b. \end{aligned} \tag{3.6}$$

则 $\phi(x) = -\sum_{i=1}^m \log(b_i - a_i^Tx)$, $\mathbf{dom}\phi = \{Ax \prec b\}$. 其中 a_i^T 是 A 的行. 微分后代入 3.5可以得到 central condition:

$$tc + A^td = 0. \tag{3.7}$$

随着 $t\to\infty$, 内点轨迹趋近于 x^*. (详细过程见书 565 页).

4 压缩感知与稀疏优化

4.1 压缩感知

对于线性方程组

$$Ax=b,\ A\in\mathbb{R}^{m\times n} \tag{4.1}$$

当 $m \ll n$ 时, 方程组存在无穷多解. 一般情况下考虑的是稀疏解, 即

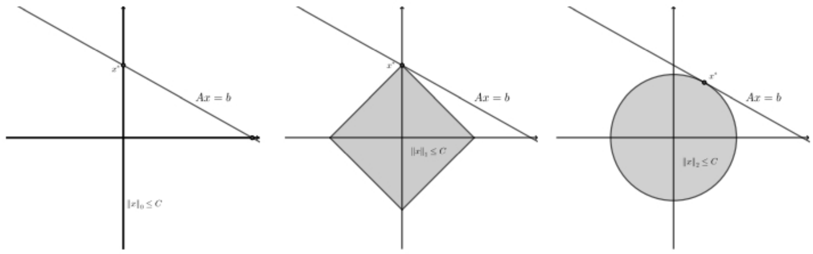
$$\begin{aligned} min_{x\in\mathbb{R}^n}\ \|x\|_0 \\ s.t.\ Ax=b \end{aligned} \tag{4.2}$$

其中 $\|x\|_0$ 指的是 x 中非零元素个数.

上述问题是 NP 难. 但可以证明, 一定条件下 l_0 范数问题可以转化为 l_1 范数优化问题. 即

$$\begin{aligned} min_{x\in\mathbb{R}^n}\ \|x\|_1 \\ s.t.\ Ax=b. \end{aligned} \tag{4.3}$$

但不能转化为更易求解的 l_2 范数. 严格证明过于复杂 (见课件), 但可以这样理解: 考虑简化情况, 在二维空间讨论求解 $Ax=b$, 此时三种优化问题等价于寻找最小 C, 使得**范数球** $\{x|\|x\|\leq C\}$ 恰好与 $Ax=b$ 相交. 显然, 对 l_0 范数, C=1 时问题解即直线与坐标轴交点. 对 l_1 范数根据不同 C, 结果也在坐标轴上. 而 l_2 范数不能保证解的稀疏性.



(a) ℓ_0 范数 (b) ℓ_1 范数 (c) ℓ_2 范数

4.2 稀疏优化算法

4.2.1 近似点梯度法

考虑 $f(x)=\frac{1}{2}\|Ax-b\|_2^2$, 于是有 $\nabla f(x)=A^T(Ax-b)$. 考虑含正则项的优化问题

$$min\ \psi_\mu(x)=\mu\|x\|_1+f(x). \tag{4.4}$$

采用一阶展开 + 梯度近似的方法, 可以得到

$$\begin{aligned}x^{k+1}&=\operatorname{argmin}_{x\in\mathbb{R}^n}\ \mu\|x\|_1+\left(\nabla f(x^k)\right)^T(x-x^k)+\frac{1}{2\tau}\|x-x^k\|_2^2\\&=\operatorname{argmin}_{x\in\mathbb{R}^n}\ \mu\|x\|_1+\frac{1}{2\tau}\|x-\left(x^k-\tau\nabla f(x^k)\right)\|_2^2\\&=shrink\left(x^k-\tau\nabla f(x^k),\mu\tau\right)\end{aligned} \tag{4.5}$$

其中

$$\begin{aligned}\operatorname{shrink}(y,\nu)&=\operatorname{argmin}_{x\in\mathbb{R}}\ \nu\|x\|_1+\frac{1}{2}\|x-y\|_2^2\\&=\operatorname{sgn}(y)\ max(|y|-\nu,0)\\&=\begin{cases}y-\nu\operatorname{sgn}(y), & |y|>\nu\\ 0, & else\end{cases}\end{aligned} \tag{4.6}$$

4.2.2 交替方向乘子法 (ADMM)

首先介绍增广 Lagrange 乘子法: 考虑

$$\begin{aligned}&min_x\ \|x\|_1\\&s.t.\ Ax=b.\end{aligned} \tag{4.7}$$

在 Lagrange 函数基础上引入二次惩罚项, 得到

$$L_\sigma(x,\lambda)=\|x\|_1+\lambda^T(Ax-b)+\frac{1}{2\sigma}\|Ax-b\|_2^2. \tag{4.8}$$

于是迭代过程

$$\begin{cases}x^{k+1}=\operatorname{argmin}_x\ L_\sigma(x,\lambda^k)\\ \lambda^{k+1}=\lambda^k+\frac{1}{\sigma}\left(Ax^{k+1}-b\right)\end{cases}. \tag{4.9}$$

再考虑 4.7的对偶问题 (参考书 255 页). 可以写出为

$$\begin{aligned} & max \ \ -b^T\nu \\ & s.t.\ \|A^T\nu\|_* \leq 1 \end{aligned} \tag{4.10}$$

约束条件也可写为

$$s.t.\ A^T\nu = s,\ \|s\|_* \leq 1. \tag{4.11}$$

于是增广 Lagrange 方程:

$$L(\nu, s, \lambda) = b^T\nu + \lambda^T\left(A^T\nu - s\right) + \frac{1}{2\sigma}\|A^T\nu - s\|_2^2. \tag{4.12}$$

ADMM 的主要思想是: 对变量分别作迭代. 即

$$\begin{aligned} & \nu^{k+1} = \text{argmin}_\nu\ L(\nu, s^k, \lambda^k) \\ & s^{k+1} = \text{argmin}_s\ L(\nu^{k+1}, s, \lambda^k),\ s.t.\ \|s\| \leq 1 \\ & \lambda^{k+1} = \lambda^k + \frac{A^T\nu^{k+1} - s^{k+1}}{\sigma}. \end{aligned} \tag{4.13}$$

这三个式子都可以写出显式解 (见课件).

5 低秩矩阵恢复

本章主要以推荐系统为例考虑.

5.1 协同过滤

对于用户商品对 (x, i), 假设评价

$$\hat{r}_{xi} = b_{xi} + \sum_{j\in N(i;x)} w_{ij}(r_{xj} - b_{xj}) \tag{5.1}$$

其中 b_{xi} 是商品 i 的基准评价估计. $N(i;x)$ 是用户 x 评价过的与 i 相似的商品. 用最小二乘法可以估计出上面的 w_{ij}.

5.2 隐变量模型

回忆奇异值分解 (SVD): 如果 A 是 $m\times n$ 的实矩阵, 则存在 U, V 满足 $U^TU = I, V^TV = I$, 且 $U^TAV = \text{diag}(\sigma_1, ..., \sigma_p), p = min(m, n)$.

由于 A 存在缺失项, 所以可以用梯度下降法求出 P, Q:

$$min_{P,Q} \sum_{(i,x)\in \text{train set}} \left(r_{xi} - q_i p_x^T\right)^2 + \lambda\left[\sum_x \|p_x\|_2^2 + \sum_i \|q_i\|_2^2\right]. \tag{5.2}$$

矩阵恢复: 在尝试得到 A 缺失值时, 我们希望得到低秩矩阵. 即

$$\begin{aligned} & min\ rank(X) \\ & s.t.\ X_{ij} = M_{ij} \end{aligned} \tag{5.3}$$

其中 M_{ij} 是已知的部分. 这一问题是 NP 难, 但利用 SVD, 可以转换为 $rank(X) = \|X\|_0$. 进而转化为 l_1 范数优化.(见 4.1 节)

稀疏与低秩矩阵分解: 考虑视频分解问题. 需要将视频分解为静态和动态两部分. 即

$$\begin{aligned} &min_{W,E}\ \|W\|_* + \mu\|E\|_1 \\ &s.t.\ W + E = M \end{aligned} \tag{5.4}$$

于是增广 Lagrangian

$$L(W,E,\Lambda) = \|W\|_* + \mu\|E\|_1 + \langle \Lambda, W+E-M\rangle + \frac{1}{2\beta}\|W+E-M\|_F^2. \tag{5.5}$$

可以用 ADMM(式 4.13) 进行优化.

6 最优输运

定义 $C = M_{XY}$, 其中 $(M_{XY})_{ij} = d(x_i,y_j)^p$ 表示点 x_i, y_j 的 p 次距离. 则 Wasserstein 距离可表示为

$$L(a,b,C) = min\{\sum_{ij} C_{ij}\Pi_{ij}; \Pi \in \mathbf{U}(a,b)\}. \tag{6.1}$$

输运问题可以表示为: 给定一组点阵和对应的概率 $\{Y^j, b^j\}$, 求解 $\{X^i, a^i\}$ 满足

$$min_{X,a}\ \frac{1}{N}\sum_{k=1}^{N} L\left(a, b^k, M_{XY^k}\right). \tag{6.2}$$

为了避开 $\Pi_{ij} >= 0$ 的约束, 可以引入 entropy. 即**熵正则化**. 定义

$$H(\Pi) = -\sum_{ij}\Pi_{ij}\left(\log(\Pi_{ij}) - 1\right). \tag{6.3}$$

于是熵正则化问题可以写为

$$L^\epsilon(a,b,C) = min_{\Pi\in\mathbf{U}(a,b)}\ \langle \Pi, C\rangle - \epsilon H(\Pi). \tag{6.4}$$

当 $\epsilon = 0$ 时回到原始问题.

上述问题可以用 Sinkhorn 解法. 具体参考课件.

7 随机优化

7.1 次梯度法

在机器学习中大规模的数据集上, 考虑最优化问题

$$min_{x\in\mathbb{R}^n}\ f(x) \tag{7.1}$$

梯度下降给出的解法是

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \tag{7.2}$$

但由于数据量大, 事实上每次可以选取一部分样本做 mini-batch 梯度下降. 即次梯度下降 (subgradient method):

$$x_{k+1} = x_k - \alpha_k g_k,\ \ g_k \in \partial f(x_k). \tag{7.3}$$

考虑到正则项, 优化方程式也等价于

$$x_{k+1} = \operatorname{argmin}_x \left[f(x_k) + \langle g_k, x - x_k\rangle + \frac{1}{2\alpha_k}\|x - x_k\|_2^2 \right]. \tag{7.4}$$

7.2 可变度量方法

首先给出可变度量方法 (Variable Metric Method) 的收敛性结论:

$$\mathbb{E}\left[\sum_{k=1}^{K}\left(f(x_k)-f(x^*)\right)\right] \le \frac{1}{2}\mathbb{E}\left[\|x_1-x^*\|_{H_1}^2+\sum_{k=1}^{K}\|g_k\|_{H^{-1}}^2\right] \tag{7.5}$$

其中 $\|x\|_{H_k}^2=\langle x, H_k x\rangle$. 证明见课件. 上式中左边即所有与最优结果误差之和, 右边第二项即需要最小化的部分.

下面结合 AdaGrad 算法介绍. 可变度量法的一般形式为

$$x_{k+1}=\mathrm{argmin}_{x\in C}\{\langle g_k, x-x_k\rangle+\frac{1}{2}\langle x-x_k, H_k(x-x_k)\rangle\}. \tag{7.6}$$

对于 AdaGrad 算法, 有

$$H_k=\frac{1}{\alpha}\mathrm{diag}\left(\sum_{i=1}^{k} g_i * g_i\right)^{1/2}. \tag{7.7}$$

于是优化问题为

$$\begin{aligned} &min\ \sum_{t=1}^{K}\|g_t\|_{H^{-1}}^2 \\ &s.t.\ H\succeq 0,\ tr(H)\le c \end{aligned} \tag{7.8}$$

其中

$$H=\begin{pmatrix} s_1 & 0 & \cdots & 0\\ 0 & s_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & s_d \end{pmatrix}. \tag{7.9}$$

式 7.8可以等价为

$$\begin{aligned} &min_s\ \sum_{i=1}^{d}\frac{\sum_{t=1}^{K} g_{t,i}^2}{s_i} \\ &s.t.\ \mathbf{1}^T s\le c,\ s\ge 0. \end{aligned} \tag{7.10}$$

于是可以构造 Lagrange 方程

$$L(s,\lambda,\theta)=\sum_{i=1}^{d}\frac{\|g_{1:K,i}\|_2^2}{s_i}-\lambda^T s+\theta\left(\mathbf{1}^T s-c\right). \tag{7.11}$$

由 $\partial L/\partial s_i=0$ 以及 KKT 条件 $\lambda_i s_i=0$ 可以得到

$$s_i=\frac{c\|g_{1:K,i}\|_2}{\sum_{i=1}^{d}\|g_{1:K,i}\|_2}. \tag{7.12}$$

7.3 近似矩阵乘积

对于数据量很大的矩阵乘法, 可以用抽样的方法计算乘积. 矩阵乘积问题

$$AB=\sum_{i=1}^{n} A_{m\times 1}^i\ B_{1\times p}^i. \tag{7.13}$$

确定一组概率 $p_i, i=1:n, \sum p_i=1$, 使得 $P(j_t=i)=p_i, t=1:c$. 于是上式可以估计为

$$AB\approx\frac{1}{c}\sum_{t=1}^{c}\frac{1}{p_{j_t}}A_{m\times 1}^{j_t}\ B_{1\times p}^{j_t}. \tag{7.14}$$

于是原始的 $AB=(m\times n)\cdot(n\times p)$ 问题简化为 $CR=(m\times c)\cdot(c\times p)$ 问题. 对于该算法, 可以证明

$$\mathbb{E}\left(\|AB-CR\|_F\right)\le\frac{1}{c}\|A\|_F\|B\|_F. \tag{7.15}$$

证明见课件. 上式给出了误差上界.

在此基础上可以进一步得到近似奇异值分解 (SVD).

Algorithm 1: Approximate SVD

Input: $A_{m\times n}$, p_i, $1\le k\le c\le n$

Output: k-maximum SV of matrix A and corresponding basis vector

```
for t = 1 : c do
    pick i_t ∈ 1, ..., n with P(i_t = α) = p_α;
    C^t ← A^{i_t} / sqrt(c p_{i_t})        % sampling
C^T C = Σ_{t=1}^{c} σ_t(C)^2 y^t (y^t)^T    % SVD of C
for t = 1 : k do
    h^t = C y^t / σ_t(C);
    H_k^t = h^t
return σ_t(C), H_k^t for t = 1 : k
```

8 数据降维

假设 $x\in\mathbb{R}^D$ 是均值为 μ, 方差为 Σ 的随机向量. PCA 问题即使得 $y=z^Tx$ 的方差最大:

$$\begin{aligned}Var(y)&=\mathbb{E}\left[((z^Tx)-\mathbb{E}(z^Tx))^2\right]\\&=z^T\mathbb{E}[(x-\mathbb{E}x)(x-\mathbb{E}x)^T]z\\&=z^Tcov(\mathbf{X})z\end{aligned} \tag{8.1}$$

于是可以写出最优化问题

$$\begin{aligned}&max\ z^Tcov(\mathbf{X})z\\&s.t.\ \|z\|_2^2=1.\end{aligned} \tag{8.2}$$

记 $cov(\mathbf{X})=B$, 可以写出 Lagrange 函数

$$L(z,\lambda)=z^TBz-\lambda(\|z\|_2^2-1). \tag{8.3}$$

利用 $\nabla_zL=0$ 可以求出 $Bz=\lambda z$, 说明 PCA 问题本质上为求 $cov(\mathbf{X})$ 的特征向量.

PCA 与 SVD 的关系

记

$$\bar{X} = X - \frac{1}{n}\mathbf{1}\mathbf{1}^T X. \tag{8.4}$$

另外 $\bar{X}$ 的 SVD 结果为

$$\bar{X} = U\Sigma V^T, \Sigma = diag(\sigma_1, ..., \sigma_n). \tag{8.5}$$

于是

$$\begin{aligned} cov(X) &= \frac{1}{n-1}(X - \mathbb{E}X)^T(X - \mathbb{E}X) \\ &= \frac{1}{n-1}\left(X - \frac{1}{n}\mathbf{1}\mathbf{1}^T X\right)^T \left(X - \frac{1}{n}\mathbf{1}\mathbf{1}^T X\right) \\ &= \bar{X}^T\bar{X} \\ &= V\Sigma^2 V^T. \end{aligned} \tag{8.6}$$

PCA 的结果即取 $\bar{X}V = U\Sigma$ 的前 d 列.

9 次模优化

本章主要以推荐算法为例考虑.

9.1 次模 (submodularity) 定义

问题举例: 对于全体用户和所有商品, 假设每个商品 i 的受众是 S_i, 定义

$$F(A) = |\cup_{i\in A}\ S_i| \tag{9.1}$$

为商品集合 A 的所有受众个数. 优化问题即

$$max_{|A|<k}\ F(A). \tag{9.2}$$

下面给出次模的定义: 假设 F 是次模函数, 且集合 $A \subseteq B$, $s \notin B$, 则有

$$F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B). \tag{9.3}$$

这反映了边际效用递减原理.

9.2 次模极大化

最简单的方法是贪心算法, 即每次选择元素 s, 使得 $F(A_i \cup \{s\})$ 最大化. 可以证明:

$$F(A_{greedy}) \geq \left(1 - \frac{1}{e}F(A^*)\right) \tag{9.4}$$

表明贪心算法可以给出近似最优解.

在大规模数据集上, 可以优化贪心算法: 例如第一次计算 $\triangle F(s|A_i)$ 时对结果进行排序. 之后的每次添加元素只计算靠前的部分元素带来的增益.

10 强化学习

10.1 马尔科夫决策过程 (MDP)

回顾定义

- on-policy value function:

$$V^{\pi}(s) = \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t | s\right] \tag{10.1}$$

- on-policy action-value function:

$$Q^{\pi}(s,a) = \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t | (s,a)\right] \tag{10.2}$$

由此进而给出 Bellman 方程:

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{a\sim\pi,s'\sim P}[R(s,a,s') + \gamma V^{\pi}(s')] \\ Q^{\pi}(s,a) &= \mathbb{E}_{s'\sim P}[R(s,a,s') + \gamma \mathbb{E}_{a'\sim\pi}[Q^{\pi}(s',a')]]. \end{aligned} \tag{10.3}$$

相应得, 有 Bellman 最优方程:

$$\begin{aligned} V^{*}(s) &= max_a\ \mathbb{E}_{s'\sim P}[R(s,a,s') + \gamma V^{*}(s')] \\ Q^{*}(s,a) &= \mathbb{E}_{s'\sim P}[R(s,a,s') + \gamma\ max_{a'}[Q^{*}(s',a')]]. \end{aligned} \tag{10.4}$$

上式 10.4可以等价于 LP 问题:

$$\begin{aligned} &min_{\mathbf{v}\in\mathbb{R}^s} \sum_{s} w_s \mathbf{v}_s \\ &s.t.\ \mathbf{v}_s \geq R(s,a) + \gamma P(s,a)^T \mathbf{v},\ \forall a,s \end{aligned} \tag{10.5}$$

其中 $w_s > 0$. 可以证明 V^* 是上述 LP 问题的最优解 (见课件), 进而说明了等价性.

迭代优化算法:

- Value Iteration

Algorithm 2: value iteration

Input: random initial v^0, ϵ

Output: optimal policy π^*

while $\|\mathbf{v}^k(s) - \mathbf{v}^{k-1}(s)\|_\infty > \epsilon\frac{1-\gamma}{2\gamma}$ **do**

  for $s \in S$ **do**

    $\mathbf{v}^k(s) \leftarrow max_{a\in A}\left[R(s,a) + \gamma\sum_{s'} P(s,a,s')\mathbf{v}^{k-1}(s')\right]$

  $k \leftarrow k+1$

$\pi(s) \in argmax_a R(s,a) + \gamma P(s,a)^T \mathbf{v}^k$

- Policy Iteration. 解决了 value iteration 中 $P(s,a,s')$ 不更新的问题.
- Q-value iteration. 可以仿照 value iteration 算法进行优化.

Algorithm 3: policy iteration

Input: random initial π^0, k

Output: optimal policy π^*

for $t = 1, ..., k$ **do**

  policy evaluation: $\mathbf{v}^k(s) \leftarrow \mathbb{E}_{a\sim\pi(s)}\left[R(s,a,s') + \gamma \sum_{s'} P(s,a,s')\mathbf{v}^k(s')\right], \forall s$

  policy improvement: $\pi^{k+1}(s) \leftarrow argmax_a R(s,a) + \gamma \mathbb{E}_{s'}[\mathbf{v}^k(s')|s,a], \forall s$

10.2 TD-learning

在优化 Bellman 方程 10.3时采用 temporal difference 方法, 即

$$\begin{aligned} \hat{V}(s_t) &\leftarrow (1-\alpha_t)\hat{V}(s_t) + \alpha_t\left(r_t + \gamma\hat{V}(s_{t+1})\right) \\ \hat{V}(s_t) &\leftarrow \hat{V}(s_t) + \alpha_t\delta_t \end{aligned} \tag{10.6}$$

其中 $\delta_t = r_t + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t)$.

上述 $\hat{V}(s_t)$ 更新规则可以使用到 value iteration 算法中. 这里也可以用参数化的 value function: $\hat{V}_\phi(s_t)$, 即可得到深度强化学习 DQN 模型. 类似得, 也可以将 TD 方法用到 policy iteration 中, 只需

$$\begin{aligned} \hat{Q}(s_t,a_t) &\leftarrow \hat{Q}(s_t,a_t) + \alpha_t\delta_t \\ \pi^{k+1}(s) &\leftarrow argmax_a\ \hat{Q}^{\pi^k}(s_t,a_t) \end{aligned} \tag{10.7}$$

10.3 Q-learning

从 Q-value iteration 算法中可以看出, 其本质是计算给定 policy 下的 Q value. 而 Q-learning 则是通过采样方式直接计算最优 Q value.

在 Q-value iteration 中, Q function 的迭代由

$$Q_{k+1}(s,a) \leftarrow R(s,a) + \gamma\sum_{s'} P(s,a,s')\left[max_{a'}Q_k(s',a')\right] \tag{10.8}$$

给出. 而在 Q-learning 中, 这一步通过类似 TD-learning 的方式给出. 由于 $\hat{Q}$ 由 array 形式存储, 因此也称为 tabular Q-learning 算法.

Algorithm 4: tabular Q-learning

Input: state distribution D, $\hat{Q}$ array for Q-value estimate, ϵ

Output: $\hat{Q}$

while $\triangle\hat{Q} > \epsilon$ **do**

  $t = 1, s_1 \sim D$ % starting an episode

  while *episode not terminates* **do**

    take action a_t, observe reward r_t and new state s_{t+1};

    $\delta_t \leftarrow \left(r_t + \gamma\max_{a'}\hat{Q}(s_{t+1},s')\right) - \hat{Q}(s_t,a_t)$;

    $\hat{Q}(s_t,a_t) \leftarrow \hat{Q}(s_t,a_t) + \alpha_t\delta_t$;

    $t \leftarrow t+1$

10.4 策略梯度下降

在之前的 Q-learning 算法中, 使用的 policy 都是贪心 (最大) 策略. 如果假设 π_θ 可微, 也可以通过梯度下降方式去估算随机策略. 最优化问题即

$$max_\theta \ \rho(\pi_\theta). \tag{10.9}$$

其中 $\rho(\pi_\theta)$ 即策略 π_θ 的收益.

先考虑有限 MDP 问题. 此时路径 (即 state-action 序列)$D^\pi(\tau)$ 的收益

$$D^\pi(\tau) = \prod_{i=1}^{H-1} \pi(s_i, a_i) P(s_i, a_i, s_{i+1}). \tag{10.10}$$

进而算出梯度

$$\begin{aligned}\nabla_\theta \rho(\pi_\theta) &= \mathbb{E}_\tau \left[R(\tau) \nabla_\theta \log(D^\pi(\tau))\right] \\ &= \mathbb{E}_\tau \left[R(\tau) \sum_{t=1}^{H-1} \nabla_\theta \log(\pi_\theta(s_t, a_t))\right]\end{aligned} \tag{10.11}$$

其中 $R(\tau) = \sum_{t=1}^{H-1} \gamma^{t-1} R(s_t, a_t)$. 证明参考课件. 另外可以看出, 在上式中如果将 $R(\tau)$ 替换为 $R(\tau) - b$, 并不影响结果. 这里 b 可以与状态和时间相关, 与 π_θ 无关. 这样可以减小梯度估计的方差.

对于无穷 MDP 问题, 可以定义平均收益

$$\begin{aligned}\rho^\pi &= \lim_{T\to\infty} \frac{1}{T} \mathbb{E}[r_1 + ... + r_T | \pi] \\ &= \sum_s d^\pi(s) \sum_a \pi(s, a) R(s, a)\end{aligned} \tag{10.12}$$

其中 $d^\pi(s) = \lim_{t\to\infty} \Pr(s_t = s | s_1, \pi)$. 还可以定义衰减型收益

$$\begin{aligned}\rho(\pi, s_1) &= \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t | \pi, s_1\right] \\ &= \sum_s d^\pi(s) \sum_a \pi(s, a) R(s, a)\end{aligned} \tag{10.13}$$

其中 $d^\pi(s) = \lim_{T\to\infty} \sum_{t=1}^{T} \gamma^{t-1} \Pr(s_t = s | s_1, \pi)$. 这两种模型中都假设 $d^\pi(s)$ 是给定策略 π 下的稳态分布, 且分布对所有策略的初始态 s_1 无关.

于是对于无穷 MDP 问题, 梯度计算

$$\begin{aligned}\nabla_\theta \rho(\pi_\theta, s_1) &= \sum_s d^{\pi_\theta}(s) \sum_a Q^{\pi_\theta}(s, a) \nabla_\theta \pi_\theta(s, a) \\ &= \sum_s d^{\pi_\theta}(s) \left[\mathbb{E}_{a\sim\pi(s)} \left(Q^{\pi_\theta}(s, a) \nabla_\theta \log(\pi_\theta(s, a))\right)\right].\end{aligned} \tag{10.14}$$

证明见课件. 上式也可写作

$$\nabla_\theta \rho(\pi_\theta) = \sum_a Q^{\pi_\theta}(s, a) \nabla_\theta \pi_\theta(s, a). \tag{10.15}$$

10.5 Actor-critic 方法

前面提到的方法中, 一类可以归纳为 Actor-only 方法 (e.g. policy gradient), 即通过采样模拟来优化参数化的 policy. 这类方法的缺点是: (1) 梯度估计方差太大; (2) 新的梯度估

计与之前的估计无关, 即没有 learning 的过程. 另一类是 critic-only 方法 (e.g. Q-learning), 它们在非最优的 policy 空间中很难得到好的结果.

因此通过 Actor-critic 方法可以结合二者. 在式 10.11中已经指出, 可以通过 baseline 方式减小方差. 这里另一种 Q-value 近似的方式是使用 value function. 类似式 10.14的形式, 可以写出

$$\nabla_\theta \rho(\pi_\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(s,a) f_w(s,a) \tag{10.16}$$

证明以及 w 满足的条件见课件. 在常用的 Softmax policy 中, 可以取

$$\begin{aligned} f_w(s,a) &= A^\pi(s,a) \\ &= Q^\pi(s,a) - V^\pi(s) \end{aligned} \tag{10.17}$$

其中 $A^\pi(s,a)$ 是 Advantage function.