运筹学与优化方法

晁国清
计算机科学与技术学院

课程回顾

- 制约函数法

基本思想：通过构造制约函数，把约束优化问题转化为一系列无约束优化问题，进而用无约束优化方法求解。

- 外点法(惩罚函数法)

惩罚函数可从任一点$x^{(1)}$开始，一般情况下$x^{(1)} \notin S, S = \{x|x \in D, g(x) \leq 0, h(x) = 0\}$, 随着$\mu$的增大，$x^{(k)}$逐步接近$S$，因此，惩罚函数法也称外点法。

- 内点法(障碍函数法)

基本思想：

从S_0中的一个点（内点）出发，在目标函数中加入惩罚项，使迭代保持在S_0内

- 乘子法

乘子罚函数：

$$\phi(x, v, \mu) = f(x) + \sum_{i=1}^{l} v_i h_i(x) + \sum_{i=1}^{l} \mu_i h_i^2(x)$$

其中：$v \in \mathbb{R}^l$为乘子，$\mu \in \mathbb{R}^l$为罚因子。

求解$\begin{cases} \min & \phi(x, v^{(k)}, \mu^{(k)}) \\ s.t. & x \in D \end{cases}$得到$x^{(k+1)}, k = 0,1,\cdots$

课程内容

- 对偶分解Dual Decomposition
- 乘子法 Method of Multipliers
- 交替方向乘子法 Alternating Direction Method of Multipliers
- 常见的模式 Common Patterns
- 一致性Consensus

对偶问题 Dual Problem

对于等式约束的凸优化问题：

$$\begin{cases} \min & f(x) \\ s.t. & Ax = b \end{cases}$$

$f(x)$ is strictly convex and closed

Lagrangian: $L(x,u) = f(x) + u^T(Ax - b)$

Dual function: $g(u) = \inf_x L(x,u)$

Dual Problem: $\max_u g(u)$

Recover $x^* = \mathrm{argmin}_x L(x,u^*)$

对偶上升 Dual Ascent

Dual Problem: $\max_{u} g(u)$

对于对偶问题用梯度法：$u^{k+1} = u^k + \alpha^k \nabla g(u^k)$

$\nabla g\left(u^k\right) = A\tilde{x} - b$, 其中$\tilde{x} = \text{argmin}_{\text{x}} L(x, u^k)$

Dual ascent method is

$$x^{k+1} = \text{argmin}_{\text{x}} L\left(x, u^k\right) \quad \text{//x-minimization}$$

$$u^{k+1} = u^k + \alpha^k (Ax^{k+1} - b) \quad \text{//dual update}$$

对偶分解 Dual Decomposition

假设函数f是可分解的，即$f(x) = f_1(x_1) + f_2(x_2) + \cdots + f_B(x_B), x = (x_1, x_2, \cdots, x_B) \in \mathbb{R}^n, A = [A_1, A_2, \cdots, A_B], A_i \in \mathbb{R}^{m \times n_i}$

即优化问题：

$$\begin{cases} \min\limits_x \sum\limits_{i=1}^{B} f_i(x_i) \\ s.t.\ Ax = b \end{cases}$$

拉格朗日L也可以分解为$L(x, u) = L_1(x_1, u) + \cdots + L_B(x_B, u) - u^T b$

其中$L_i(x_i, u) = f_i(x_i) + u^T A_i x_i$

x-minimization也可分解为B个单独的最小化，即

$$x^{k+1} = \mathrm{argmin}_{\mathrm{x}} L(x, u) \Leftrightarrow x_i^{k+1} = \mathrm{argmin}_{\mathrm{x}} f_i(x_i) + u^T A_i x_i$$

可分解互不干扰，故可并行计算

对偶分解 Dual Decomposition

Dual Decomposition algorithm:

$$x_i^{k+1} = \text{argmin}_{x_i} f_i(x_i) + (u^k)^T A_i x_i, i = 1, \cdots, B$$

$$u^{k+1} = u^k + \alpha^k \left(\sum_{i=1}^{B} A_i x_i^{k+1} - b\right)$$

可理解为：

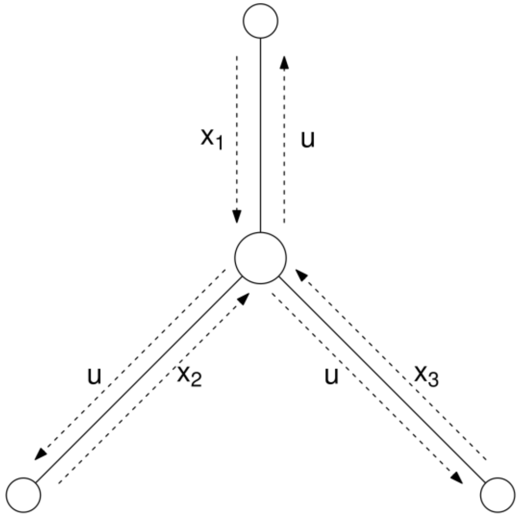
分发：分发u^k到B个处理器，每个处理器并行更新x_i

收集：从B个处理器收集$A_i x_i^{k+1}$，然后更新全局变量u

缺点：要求假设多，通常速度慢

比如对线性函数$f(x)$无法适用，

无解



乘子法 Method of Multipliers

也叫ALM(Augmented Lagrangian Method)增广拉格朗日法。

$$\begin{cases} \min & f(x) + \frac{\rho}{2}\|Ax-b\|_2^2 \\ s.t. & Ax = b \end{cases}$$

注意$\rho > 0$，$\frac{\rho}{2}\|Ax-b\|_2^2$相当于外点法的惩罚项，等价于原问题。

乘子法：

$$x^{k+1} = \mathrm{argmin}_{\mathrm{x}}\, f(x) + (u^k)^T(Ax-b) + \frac{\rho}{2}\|Ax-b\|_2^2$$

$$u^{k+1} = u^k + \rho(Ax^{k+1} - b)$$

注意：对偶更新步长为ρ

乘子法 Method of Multipliers

因为x^{k+1}最小化$f(x)+\frac{\rho}{2}\|Ax-b\|_2^2+(u^k)^T(Ax-b)$，有

$$0=\nabla_x\left(f(x)+\frac{\rho}{2}\|Ax-b\|_2^2+(u^k)^T(Ax-b)\right)$$
$$=\nabla_x f(x)+\mathrm{A}^{\mathrm{T}}\left(\rho(Ax-b)+u^k\right)$$
$$=\nabla_x f(x)+A^T u^{k+1}$$

可见对偶更新$u^{k+1}=u^k+\rho(Ax^{k+1}-b)$使得原问题的稳定性条件成立

此外，在满足收敛条件下，当$k\to\infty$时，$Ax^{k+1}-b\to 0$

故KKT条件在极限情况下满足并且x^{k+1},u^{k+1}收敛于最优解

优点：给出了较好的收敛性

缺点：失去了可分解性由于$\frac{\rho}{2}\|Ax-b\|_2^2$的出现

交替方向乘子法ADMM(Alternating Direction Method of Multipliers)

ADMM的出现是为了具有ALM的收敛性并克服不可分解的缺点

ADMM适用的问题形式：

$$\begin{cases} \min & f(x)+g(z) \\ s.t. & Ax+Bz=c \end{cases}$$

具有两套变量和可分解的目标函数

定义增广拉格朗日：

$$L_\rho(x,z,u)=f(x)+g(z)+u^T(Ax+Bz-c)+\frac{\rho}{2}\|Ax+Bz-c\|_2^2$$

ADMM算法：

$$x^{k+1}=\mathrm{argmin}_x L_\rho(x,z^k,u^k) \qquad //\text{x-minimization}$$

$$z^{k+1}=\mathrm{argmin}_z L_\rho(x^{k+1},z,u^k) \qquad //\text{z-minimization}$$

$$u^{k+1}=u^k+\rho(Ax^{k+1}+Bz^{k+1}-c) \qquad //\text{dual update}$$

注意：如果同时优化x和z，就退化为ALM

放缩形式的ADMM Scaled form ADMM

放缩形式：定义$w=\frac{u}{\rho}$, 则有

$$L_\rho(x,z,u)=f(x)+g(z)+u^T(Ax+Bz-c)+\frac{\rho}{2}\|Ax+Bz-c\|_2^2$$
$$=f(x)+g(z)+\frac{\rho}{2}\|Ax+Bz-c+w\|_2^2-\frac{\rho}{2}\|w\|_2^2$$

ADMM迭代公式变为：

$$x^{k+1}=\mathrm{argmin}_{\mathrm{x}}\left(f(x)+\frac{\rho}{2}\left\|Ax+Bz^k-c+w^k\right\|_2^2\right)$$
$$z^{k+1}=\mathrm{argmin}_{\mathrm{z}}\left(g(z)+\frac{\rho}{2}\left\|Ax^{k+1}+Bz-c+w^k\right\|_2^2\right)$$
$$w^{k+1}=w^k+Ax^{k+1}+Bz^{k+1}-c$$

注意：w的最后一次更新是所有残差residual的求和

$$w^{k+1}=w^1+\sum_{i=2}^{k+1}(Ax^i+Bz^i-c)$$

收敛性

假设f, g convex, closed, proper(定义域非空并且不会取$-\infty$和$+\infty$), L_0有鞍点

则ADMM收敛

$$Ax^k + Bz^k - c \to 0$$
$$f\left(x^k\right) + g\left(z^k\right) \to p^*$$
$$u^k \to u^*$$

p^*目标函数的最优值, u^*是对偶变量的最优解

关于ADMM的收敛率，大体和一阶方法相当，仍在研究发展中

常见的模式 Common Patterns

有几类常见的模式，可以利用它们特有的结构简化变量的更新操作

- 可分解Decomposition
- 邻近算子Proximal Operator
- 二次目标函数Quadratic Objective
- 光滑目标函数Smooth Objective

可分解Decomposition

x update step 需要最小化$f(x)+\frac{\rho}{2}\|Ax-v\|_2^2$, 其中$v=-Bz^k+c-w^k$是常数。

同理，z update step需要最小化$g(z)+\frac{\rho}{2}\|Bz-v'\|$, 其中$v'=-Ax^k+c-w^k$

假如f是块可分的，$f(x)=f_1(x_1)+f_2(x_2)+\cdots+f_B(x_B), x=(x_1,x_2,\cdots,x_B)\in\mathbb{R}^n$

A是一致块可分的：A^TA是块对角的

那么 x update可分为B个x_i的并行update

$$\nabla f(x)+\rho A^T(Ax-v)=\begin{pmatrix}\nabla_{x_1}f_1\\ \vdots\\ \nabla_{x_B}f_B\end{pmatrix}+\rho\begin{pmatrix}(A^TA)_1x_1\\ \vdots\\ (A^TA)_Bx_B\end{pmatrix}-\rho\begin{pmatrix}(A^Tv)_1\\ \vdots\\ (A^Tv)_B\end{pmatrix}=0$$

可见，x update是可分的

邻近算子Proximal Operator

x update: $x^{k+1}=\mathrm{argmin_x}\left(f(x)+\frac{\rho}{2}\|Ax-v\|_2^2\right)$

$if\ A=I,$

则$x^{k+1}=\mathrm{argmin_x}\left(f(x)+\frac{\rho}{2}\|x-v\|_2^2\right)=\mathrm{prox_{f,\rho}}(v)$

举例1：$f=I_C=\begin{cases}0, & x\in C\\ +\infty, & x\notin C\end{cases}$

x update: $x^{k+1}=\mathrm{argmin_{x\in C}}\|x-v\|_2^2=\Pi_C(v)$

举例2：$f=\lambda\|\ \ \|_1\ (\ell_1\ norm)$

$$x^{k+1}=\mathrm{argmin_x}\left(\lambda\|x\|_1+\frac{\rho}{2}\|x-v\|_2^2\right)$$
$$=\mathrm{argmin_x}\left(\frac{\lambda}{\rho}\|x\|_1+\frac{1}{2}\|x-v\|_2^2\right)$$

令 $a=\frac{\lambda}{\rho}, x^{k+1}=\mathrm{argmin_x}\left(a\|x\|_1+\frac{1}{2}\|x-v\|_2^2\right)$

邻近算子Proximal Operator

令 $a=\frac{\lambda}{\rho}, x^{k+1}=\mathrm{argmin}_{\mathrm{x}}\left(a\|x\|_1+\frac{1}{2}\|x-v\|_2^2\right)$

如果$x>0$，则$a+x-v=0, x=v-a>0$, 即$v>a$

如果$x<0$，则$-a+x-v=0, x=v+a<0$, 即$v<-a$

如果$x>0$ 并且 $-\mathrm{a}\le v\le a$, 则$x=v-a\le 0, x^{k+1}=0$

如果$x<0$ 并且$-\mathrm{a}\le v\le a$, 则$x=v+a\ge 0, x^{k+1}=0$

$$x^{k+1}=\begin{cases} v-a & v>a \\ 0 & -a\le v\le a \\ v+a & v<-a \end{cases}$$

$$=S_{\frac{\lambda}{\rho}}(v)$$

Soft-thresholding

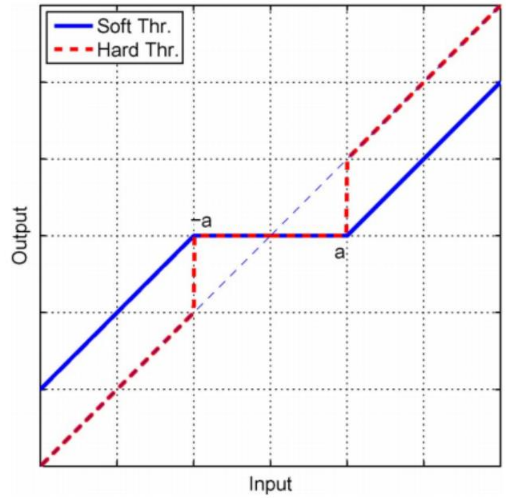


Fig. 1. *The soft and hard shrinkage curves.*

二次目标函数Quadratic Objective

最小化二次目标函数$f(x) = \frac{1}{2}x^T P x + q^T x + r$

$P \in S_+^n$并假定$(P + \rho A^T A)$可逆

$L_\rho = f(x) + \frac{\rho}{2}\|Ax - v\|_2^2$ 用最优性条件求解

$$\begin{aligned}\nabla L_\rho &= Px + q + \rho A^T(Ax - v) \\ &= (P + \rho A^T A)x + q - \rho A^T v = 0\end{aligned}$$

则$x = -(P + \rho A^T A)^{-1}(q - \rho A^T v)$

光滑目标函数Smooth Ojective

假如目标函数f光滑，

则可以用光滑函数最小化的标准方法求解

- 梯度下降法
- 牛顿法
- 拟牛顿法

也可以用

- warm start
- early stopping

举例Examples

Constrained Convex Optimization

问题形式$\begin{cases}\min f(x) \\ s.t. x \in C\end{cases}$

取g为C的indicator function，即$g(x) = \begin{cases}0 & x \in C \\ +\infty & x \notin C\end{cases}$

则ADMM形式为：$\begin{cases}\min & f(x) + g(z) \\ s.t. & x - z = 0\end{cases}$

增广拉格朗日$L_\rho(x,z,u) = f(x) + g(z) + u^T(x-z) + \frac{\rho}{2}\|x-z\|_2^2$,

令$w = \frac{u}{\rho}$，上式$= f(x) + g(z) + \frac{\rho}{2}\|x - z + w\|_2^2 - \frac{\rho}{2}\|w\|_2^2$

ADMM算法：$x^{k+1} = \mathrm{argmin}_x f(x) + \frac{\rho}{2}\left\|x - z^k + w^k\right\|_2^2$

$$z^{k+1} = \Pi_C(x^{k+1} + w^k)$$

$$w^{k+1} = w^k + x^{k+1} - z^{k+1}$$

哈爾濱工業大學 HARBIN INSTITUTE OF TECHNOLOGY

Lasso

Lasso问题最小化$\frac{1}{2}\|Ax-b\|_2^2+\lambda\|x\|_1$

ADMM形式：$\begin{cases}\min & \frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1 \\ s.t. & x-z=0\end{cases}$

增广拉格朗日形式$L_\rho=\frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1+u^T(x-z)+\frac{\rho}{2}\|x-z\|_2^2$

最优性条件得$\nabla_x L_\rho=A^T(Ax-b)+u+\rho(x-z)$

得到$x^{k+1}=(A^TA+\rho I)^{-1}(A^Tb+\rho z-u)$（岭回归）

$$z^{k+1}=S_{\frac{\lambda}{\rho}}(x^{k+1}+\frac{u^k}{\rho})$$

$$u^{k+1}=u^k+\rho(x^{k+1}-z^{k+1})$$

一致性Consensus

最小化B个目标函数的和，即$\min\sum_{i=1}^{B} f_i(x)$

ADMM形式：$\begin{cases}\min\ \ \sum_{i=1}^{B} f_i(x) \\ s.t.\ \ x_i - z = 0\end{cases}$

x_i局部变量，z全局变量，$x_i - z = 0$是一致性约束，还可以增加规范化项$g(z)$

增广拉格朗日$L_\rho(x,z,u) = \sum_{i=1}^{B}(f_i(x_i) + u_i^T(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2)$

ADMM算法：

$$x_i^{k+1} = argmin_{x_i}(f_i(x_i) + u_i^{k^T}(x_i - z^k) + \frac{\rho}{2}\|x_i - z^k\|_2^2)$$

$$z^{k+1} = \frac{1}{B}\sum_{i=1}^{B}(x_i^{k+1} + \frac{1}{\rho}u_i^k)$$

$$u_i^{k+1} = u_i^k + \rho(x_i^{k+1} - z^{k+1})$$

有规范化项$g(z)$, 则先执行$\mathrm{prox}_{g,\rho}$在对z平均更新

$$x_i^{k+1} = argmin_{x_i}(f_i(x_i) + u_i^{k^T}(x_i - z^k) + \frac{\rho}{2}\|x_i - z^k\|_2^2)$$

$$z^{k+1} = \frac{1}{B}\sum_{i=1}^{B}(x_i^{k+1} + \frac{1}{\rho}u_i^k)$$

$$u_i^{k+1} = u_i^k + \rho(x_i^{k+1} - z^{k+1})$$

取$\bar{x} = \frac{1}{B}\sum_{i=1}^{B} x_i^k$，$\bar{u} = \frac{1}{B}\sum_{i=1}^{B} u_i^k$由后两更新公式得到$\sum_{i=1}^{B} u_i^k = 0$

ADMM算法可简化为：

$$x_i^{k+1} = \mathrm{argmin}_{\mathrm{x_i}}\left(f_i(x_i) + u_i^{k^T}(x_i - \bar{x}^k) + \frac{\rho}{2}\|x_i - \bar{x}^k\|_2^2\right)$$

$$u_i^{k+1} = u_i^k + \rho\left(x_i^{k+1} - \bar{x}^{k+1}\right)$$

一致性Consensus

取$\bar{x}=\frac{1}{B}\sum_{i=1}^{B}x_i^k$，$\bar{u}=\frac{1}{B}\sum_{i=1}^{B}u_i^k$由后两更新公式得到$\sum_{i=1}^{B}u_i^k=0$

ADMM算法可简化为：

$$x_i^{k+1}=\mathrm{argmin}_{\mathrm{x_i}}\left(f_i(x_i)+{u_i^k}^T\left(x_i-\bar{x}^k\right)+\frac{\rho}{2}\left\|x_i-\bar{x}^k\right\|_2^2\right)$$
$$u_i^{k+1}=u_i^k+\rho\left(x_i^{k+1}-\bar{x}^{k+1}\right)$$

在每次迭代中：

1. 收集x_i^k并平均得到$\bar{x}$
2. 分发$\bar{x}$到各个处理器
3. 局部更新u_i^k(在每个处理器并行执行)
4. 局部更新x_i

哈爾濱工業大學 HARBIN INSTITUTE OF TECHNOLOGY

Consensus SVM

一致分类问题：数据集$(a_i, b_i), i = 1, \cdots, N, a_i \in \mathbb{R}^n, b_i \in \{-1, +1\}$

线性分类器：$sign(a^T w + v)$, w权值，v偏置

第i个样本数据的间隔是$b_i(a_i^T w + v)$, 希望间隔为正，相应的损失函数为$l(b_i(a_i^T w + v))$, l可以是hingle, logistic, probit, exponential loss

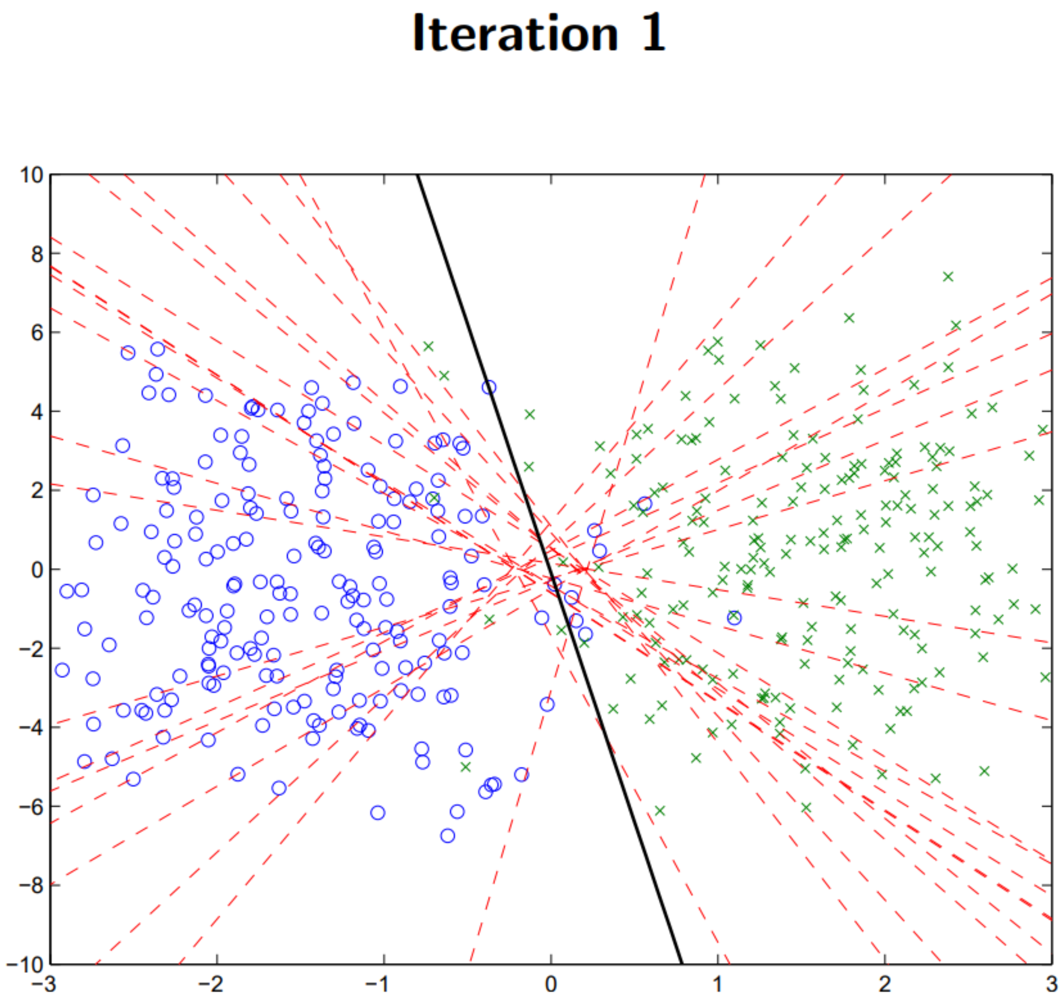
形成优化问题$\min \frac{1}{N}\sum_{i=1}^{N} l(b_i(a_i^T w + v)) + r(w)$, 其中$r(w)$是规范化项，可取$\ell_1, \ell_2$等

划分数据并用consensus ADMM求解。

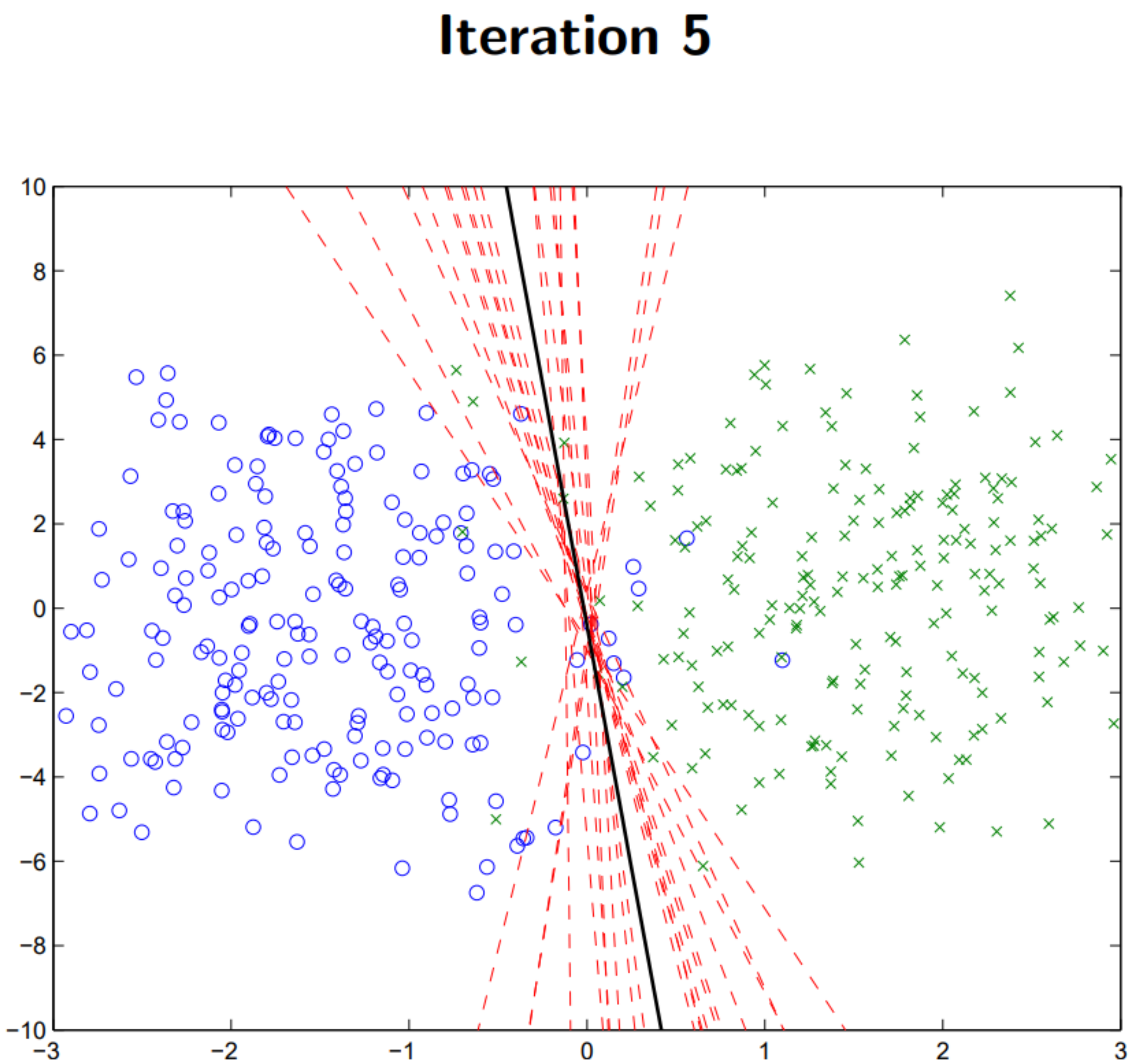
Hing loss $l(u) = (1-u)_+$加上ℓ_2规范化项形成consensus SVM

将数据N=400分成20组，利用consensus SVM求解

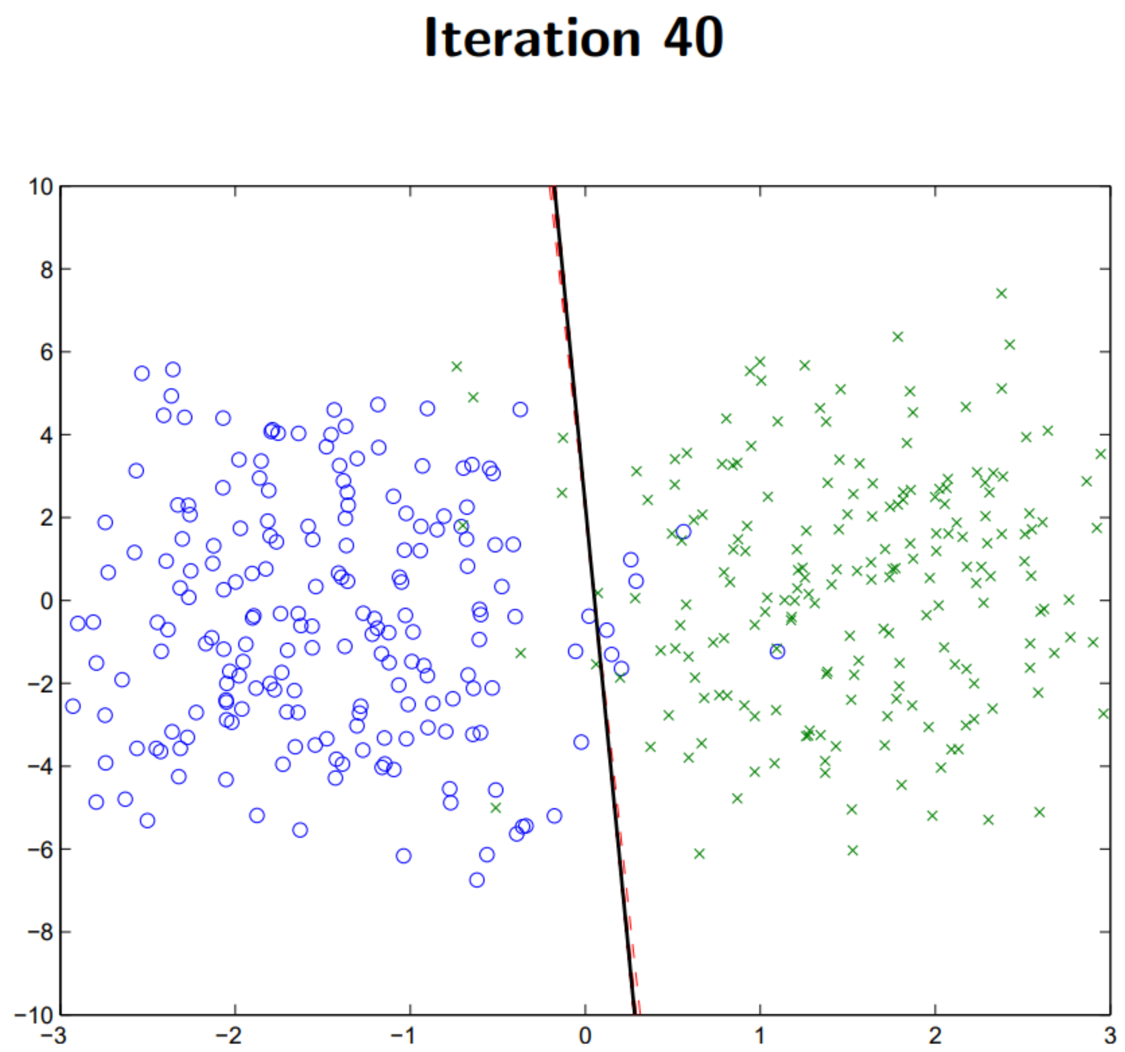
Consensus SVM



Consensus SVM



Consensus SVM



参考文献

- Alternating Direction Methods of Multipliers   Stephen Boyd   EE364b

- Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers (Boyd, Parikh, Chu, Peleato, Eckstein)