Logistic

- 今日截止：
  - 1. HW3提交：请大家检查是否**正确**提交了Writing**和**Coding部分的答案。
  - 2. Course Project组队：请大家确认自己的信息是否完整、准确的填写在了表格上。
- 即将到来：
  - 1. HW4预计将于12月5日发布，依然包括Writing和Coding两部分。
- 特别提醒：

本次TAC关于SVM求解的内容在期末考试中不会做过高要求，本次习题课以介绍相关技巧和性质为主。

SVM

TAC05

数学基础

首先我们回顾数学分析中介绍的拉格朗日乘数法和KKT条件：一种处理**有约束的**多元变量的极值问题的数学技术。

考虑一个有约束的多元变量极值问题的一般形式：

$$\begin{aligned} &\arg\min_{x} f(\mathbf{x}) \\ \text{subject to } & g_i(\mathbf{x}) \le \mathbf{0},\ \mathbf{i=1,2,\cdots,m} \\ & h_i(\mathbf{x}) = \mathbf{0},\ \mathbf{i=1,2,\cdots,m} \end{aligned}$$

通过给每个约束引入一个拉格朗日乘数λ，我们可以构建原问题的拉格朗日函数：

$$L(\mathbf{x},\lambda) = \mathbf{f(x)} + \sum_{\mathbf{i=1}}^{\mathbf{m}} \lambda_{\mathbf{i}}\mathbf{g_i(x)} + \sum_{\mathbf{i=1}}^{\mathbf{m}} \lambda'_{\mathbf{i}}\mathbf{h_i(x)}$$

注意，其中$\lambda_i \geq 0$，λ'_i正负皆可。此时我们令**L**对**x，λ，λ'**的偏导为零，联立方程，所得解中符合约束条件的即为极值。

FYI:WikiPedia 知乎专栏

Back to SVM

- 什么是SVM?
  - 回顾我们在感知机中的知识，感知机使用迭代更新超平面参数的方法在线性可分数据上寻找一个完全正确分类训练数据的超平面。
  - 如何评价若干个在训练数据上全部正确的超平面的质量？**引入超平面与正负样本的间隔 （margin）**
  - SVM是一种寻找拥有最大间隔的超平面的算法
- 我将首先探讨怎么求解hard-margin SVM，再关注Support Vector的意义。

以下内容来自 赵子平老师 2022年春 讲授本课程期间的课件和作业

Optimization Problem – I

- Optimization problem (the primal problem):

$$\begin{aligned} \underset{\mathbf{w}\in\mathbf{R}^d,\, w_0}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1, \quad \forall t \end{aligned}$$

- This is a convex quadratic programming (QP), the complexity of which depends on d.
  - This QP can be solved directly via QP numerical solving methods to find $\mathbf{w}$ and w_0, i.e., the optimal canonical separating hyperplane.
- On both sides of the hyperplane, there will be instances that are $\frac{1}{\|\mathbf{w}\|}$ away from the hyperplane and the total margin will be $\frac{2}{\|\mathbf{w}\|}$.

Optimization Problem – II

- As discussed in previous lectures, if the classification problem is not linearly separable, instead of fitting a nonlinear function, one trick is to map the problem to a new space $\mathcal{Z}$ by using nonlinear basis functions.
  - It is generally the case that this new space has more dimensions than the original space (i.e., larger than d), and, in such a case, we are interested in a method whose complexity does not depend on the input dimensionality.
- In optimization theory, it is very common and sometimes advantageous to turn the primal problem into a dual problem and then solve the latter instead.
  - In our case, it also turns out to be more convenient to solve the dual problem (whose complexity depends on the sample size N) rather than the primal problem directly (whose complexity depends on the dimensionality d).
- It will be shown that the dual problem also makes it easy for a nonlinear extension using kernel functions.

Lagrangian

- Lagrangian:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, w_0, \{\alpha_t\}) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha_t \left[r^t(\mathbf{w}^T\mathbf{x}^t + w_0) - 1 \right] \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha_t r^t(\mathbf{w}^T\mathbf{x}^t + w_0) + \sum_{t=1}^{N} \alpha_t \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w} - \mathbf{w}^T \sum_{t=1}^{N} \alpha_t r^t \mathbf{x}^t - w_0 \sum_{t=1}^{N} \alpha_t r^t + \sum_{t=1}^{N} \alpha_t\end{aligned}$$

  with Lagrange multipliers $\alpha_t \geq 0$.
- The optimal solution is a saddle point which minimizes $\mathcal{L}$ w.r.t. the primal variables $\mathbf{w}, w_0$ and maximizes $\mathcal{L}$ w.r.t. the dual variables α_t.

Eliminating Primal Variables

- Setting the derivatives of $\mathcal{L}$ w.r.t. $\mathbf{w}$ and w_0 to $\mathbf{0}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{t=1}^{N} \alpha_t r^t \mathbf{x}^t \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{t=1}^{N} \alpha_t r^t = 0 \tag{3}$$

- Plugging (2) and (3) into $\mathcal{L}$ gives the objective function G for the dual problem:

$$\begin{aligned} G(\{\alpha_t\}) &= -\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{t=1}^{N} \alpha_t \\ &= -\frac{1}{2}\sum_{t=1}^{N}\sum_{t'=1}^{N} \alpha_t \alpha_{t'} r^t r^{t'} (\mathbf{x}^t)^T \mathbf{x}^{t'} + \sum_{t=1}^{N} \alpha_t \end{aligned}$$

Dual Optimization Problem – I

- Dual optimization problem:

$$\begin{aligned} \underset{\{\alpha_t\}}{\text{maximize}} \quad & \sum_{t=1}^{N} \alpha_t - \frac{1}{2} \sum_{t=1}^{N} \sum_{t'=1}^{N} \alpha_t \alpha_{t'} r^t r^{t'} (\mathbf{x}^t)^T \mathbf{x}^{t'} \\ \text{subject to} \quad & \sum_{t=1}^{N} \alpha_t r^t = 0 \\ & \alpha_t \geq 0, \quad \forall t \end{aligned}$$

- This is also a QP problem, and its complexity depends on the sample size N (rather than the input dimensionality d):
  - Time complexity: $O(N^3)$ (for generic QP solvers)
  - Space complexity: $O(N^2)$

Dual Optimization Problem – II

- Define
$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} r^1 \\ \vdots \\ r^N \end{bmatrix},$$
and the symmetric matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$ with $h_{ij} = r^i r^j (\mathbf{x}^i)^T \mathbf{x}^j$.
- We get the equivalent reformulation
$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & \boldsymbol{\alpha}^T \mathbf{1} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^T \mathbf{r} = 0 \\ & \boldsymbol{\alpha} \geq \mathbf{0} \end{aligned}$$

Support Vectors

- Based on KKT complementarity slackness condition, we have the following results.
- For points lying beyond the margin (sufficiently away from the hyperplane), i.e., $r^t(\mathbf{w}^T\mathbf{x}^t + w_0) > 1$, they have no effect on the hyperplane. The corresponding dual variables vanish with $\alpha_t = 0$.
  - Even if any subset of them are removed or moved around, we would still get the same solution.
  - It is possible to use a simpler classifier to filter out a large portion of such instances, i.e., decreasing N, thereby decreasing the complexity of the optimization.
- Support vectors (SVs): $\mathbf{x}^t$ with $\alpha_t > 0$, i.e., $r^t(\mathbf{w}^T\mathbf{x}^t + w_0) = 1$ (exactly on the hyperplane), hence the name support vector machine (SVM).
  - Solution is determined by the data on the margin.

Computation of Primal Variables

- From (2) we get

$$\mathbf{w} = \sum_{t=1}^{N} \alpha_t r^t \mathbf{x}^t = \sum_{\mathbf{x}^t \in \mathcal{SV}} \alpha_t r^t \mathbf{x}^t$$

  where $\mathcal{SV}$ denotes the set of support vectors.

- The support vectors must lie on the margin, so they should satisfy

$$r^t(\mathbf{w}^T \mathbf{x}^t + w_0) = 1.$$

  Then, we have

$$w_0 = r^t - \mathbf{w}^T \mathbf{x}^t.$$

  - For numerical stability, in practice all support vectors are used to compute w_0:

$$w_0 = \frac{1}{|\mathcal{SV}|} \sum_{\mathbf{x}^t \in \mathcal{SV}} (r^t - \mathbf{w}^T \mathbf{x}^t)$$