Lecture 2：Dual Support Vector Machine

> 课件链接：[Hsuan-Tien Lin - dual support vector machine]()
>
> Dual Support Vector Machine(支撑向量机的对偶形式)
>
> - Motivation of Dual SVM：将SVM转化为对偶形式的原因
> - Lagrange Dual SVM：SVM的拉格朗日对偶问题
> - Solving Dual SVM：解对偶问题
> - Messages behind Dual SVM：对偶SVM背后的信息

1. Motivation of Dual SVM: 将SVM转化为对偶形式的原因

在上一章最后我们提到，将SVM配合特征转换一起使用对解决Non-linear问题效果很好。在此过程中，我们需先对原始数据进行特征转换，将它们映射到高维空间(从X空间到Z空间)：

$$\mathbf{z}_n = \Phi(\mathbf{x}_n)$$

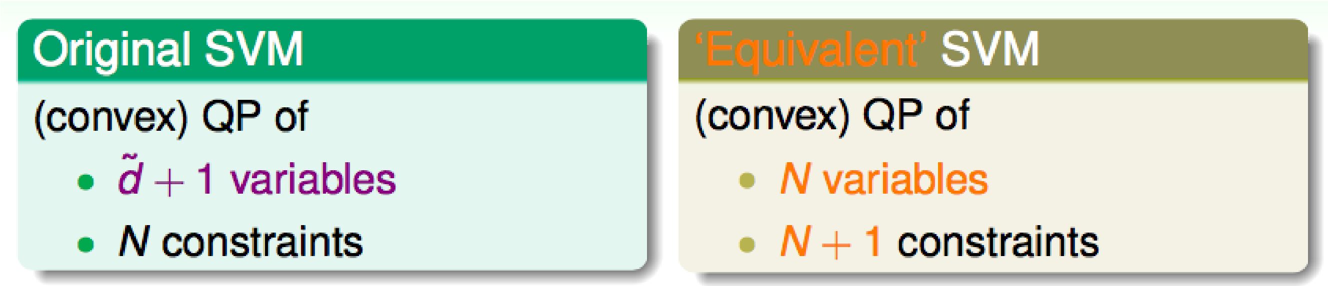
然后在Z空间中做SVM。

若此，我们需要解决的QP问题将从原来的$d+1$个变量、N个条件，变为$\tilde{d}+1$个变量、N个条件。如果$\tilde{d}$非常大，甚至无限大，即：

$$\tilde{d} >> d$$

这样的QP问题就会变得非常难解。

因此，我们希望移除Z空间中SVM计算对于$\tilde{d}$的依赖。具体而言：



上图中"等价的"SVM，实际上即为原来SVM问题的**对偶问题(dual problem)**。

拉格朗日乘子法(Lagrange Multipliers)

在《基石》正则化的推导里，我们曾使用过拉格朗日乘子法，那里的拉格朗日乘子λ被作为调整正则化程度的参数给定。而在dual SVM中，我们将λ视为未知变量，需要求解。因为有N个约束条件，因此相应地有N个拉格朗日乘子。因为在SVM的相关文献中常把拉格朗日乘子记做α而非λ，接下来的一系列推导将使用这一习惯。

Step 1：将有约束的最优化问题转化为无约束的最优化问题

方法：构建拉格朗日函数——

$$\mathcal{L}(b,\mathbf{w},\alpha)=\frac{1}{2}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{N}\alpha_n\Big(1-y_n(\mathbf{w}^T\mathbf{x}_n+b)\Big)$$

根据拉格朗日函数可将原SVM问题转化为如下等价的**"最小最大问题"**：

$$\min_{b,\mathbf{w}}\Big(\max_{all\ \alpha_n\geq 0}\mathcal{L}(b,\mathbf{w},\alpha)\Big)$$

等价的原因如下：

- 内层最大化问题，固定了$(b,\mathbf{w})$：
  - 对于那些违反了原问题约束条件的$(b,\mathbf{w})$，一定存在某些样本n，使得$(1-y_n(\mathbf{w}^T\mathbf{x}_n+b))>0$；由于是最大化问题，因此对应的$\alpha_n$应该尽可能往大取，最大取到$\infty$——最优值为$\infty$；
  - 对于那些没有违反原问题约束条件的$(b,\mathbf{w})$，所有样本n应该都满足$(1-y_n(\mathbf{w}^T\mathbf{x}_n+b))\leq 0$；由于是最大化问题，因此对应的$\alpha_n$应该尽可能往小取，最小取到0——最优值为$\frac{1}{2}\mathbf{w}^T\mathbf{w}$；
  - 可见，"未违反者"比"违反者"的目标函数值小；
- 外层再取最小化，等价于在"未违反者"中做最小化，也就是在那些没有违反原问题约束条件的$(b,\mathbf{w})$中，最优化$\frac{1}{2}\mathbf{w}^T\mathbf{w}$——这也就是原来的SVM问题。

2. Lagrange Dual SVM: SVM的拉格朗日对偶问题

上一节最后，我们通过推导得到了**原始问题(primal problem)**的另一种形式——基于拉格朗日函数的最小最大问题形式。接下来我们将要推导出原始问题的**对偶问题(对偶形式，dual)。**

Step 2：从原始问题到对偶问题

对于某一个固定的α'，有：

$$\min_{b,\mathbf{w}}\Big(\max_{all\ \alpha_n\geq 0}\mathcal{L}(b,\mathbf{w},\alpha)\Big)\geq\min_{b,\mathbf{w}}\mathcal{L}(b,\mathbf{w},\alpha')$$

这是因为："最好(best)"必然大于等于"任何某一个(any)"。对于右侧"最好"(使右侧值最大)的α，有：

$$\min_{b,\mathbf{w}}\Big(\max_{all\ \alpha_n\geq 0}\mathcal{L}(b,\mathbf{w},\alpha)\Big)\geq\max_{all\ \alpha_n\geq 0}\ \min_{b,\mathbf{w}}\mathcal{L}(b,\mathbf{w},\alpha)$$

这是因为："最好"属于"任何某一个"。

上式右侧的**"最大最小问题"**，即为**拉格朗日对偶问题。**可见，对偶问题的最优值是原始问题最优值的**下界(lower bound)。**

Step 3：根据强对偶关系确定等价性

对偶问题更容易解，这是因为内层最小化问题是针对$(b,\mathbf{w})$的无约束最优化问题。但是，$\geq$告诉我们，原始问题与对偶问题好像并不等价——仅存在**弱对偶关系(weak duality)。**

根据最优化理论，当原始问题满足一些条件时(充分条件)，它和它的对偶问题间存在**强对偶关系(strong duality)，即完全等价**：

1. 原始问题是**凸优化问题**：目标函数是凸函数，不等式约束是凸函数，等式约束是仿射函数；

Convex Optimization

A convex optimization problem with variables x:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, 2, \ldots, m \\ & a_i^T x = b_i, \quad i = 1, 2, \ldots, p \end{array}$$

where $f_0, f_1, \ldots, f_m$ are convex functions.

- Minimize convex objective function (or maximize concave objective function)
- Upper bound inequality constraints on convex functions ($\Rightarrow$ Constraint set is convex)
- Equality constraints must be affine

2. **Slater条件**：对于非仿射的不等式约束$f_i(x) \leq 0$，存在可行解，严格满足$f_i(x) < 0$——there exists strictly feasible primal variables $f_i(x) < 0$ for non-affine f_i。

Strong Duality

Strong duality (zero optimal duality gap):

$$d^* = p^*$$

If strong duality holds, solving dual is 'equivalent' to solving primal. But strong duality does not always hold

Convexity and constraint qualifications $\Rightarrow$ Strong duality

A simple constraint qualification: Slater's condition (there exists strictly feasible primal variables $f_i(x) < 0$ for non-affine f_i)

Another reason why convex optimization is 'easy'

SVM原始问题显然满足上述两个条件。因此，SVM的对偶问题与原始问题完全等价——也就是说，**存在primal-dual optimal solution** $(b, \mathbf{w}, \alpha)$，既是原始问题也是对偶问题的最优解。因此，我们可以放心求解Dual问题。

Step 4：化简对偶问题

对偶问题为：

$$\underset{all\ \alpha_n \geq 0}{max} \left(\underset{b,\mathbf{w}}{min}\ \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\mathbf{w}^T\mathbf{z}_n + b)) \right)$$

首先对内层进行最优化：

①对b求偏导后置零：$\frac{\partial \mathcal{L}}{\partial b} = 0$，得$\sum_{n=1}^{N} \alpha_n y_n = 0$；带入上式，可以消去b，得：

$$\underset{all\ \alpha_n \geq 0, \sum y_n \alpha_n = 0}{max} \left(\underset{b,\mathbf{w}}{min}\ \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\mathbf{w}^T\mathbf{z}_n)) \right)$$

②对$\mathbf{w}$求偏导后置零：$\frac{\partial \mathcal{L}}{\partial w_i} = 0$，得$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n$；带入上式，得：

$$\underset{all\ \alpha_n \geq 0, \sum y_n \alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n}{max} \quad -\frac{1}{2} || \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n ||^2 + \sum_{n=1}^{N} \alpha_n$$

至此，我们得到了SVM的对偶形式。

Step 5：明确KKT条件

什么是KKT条件？

- KKT条件是任何最优化问题的解需要满足的**必要条件**；
- 在强对偶关系成立时，KKT条件是原始-对偶最优解需要满足的**必要条件**；
- 在凸优化+Slater成立时，KKT条件是原始-对偶最优解需要满足的**充分必要条件**——这是我们的情境。

KKT条件——对于primal-dual optimal $(b, \mathbf{w}, \alpha)$，有：

- primal feasible：原始问题的约束条件

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$$

- dual feasible：对偶问题的约束条件

$$\alpha_n \geq 0$$

- dual-inner optimal：对偶问题内层的最优化条件

$$\sum y_n \alpha_n = 0,\ \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n$$

- primal-inner optimal：原始问题内层的最优化条件——**complementary slackness**

$$\alpha_n(1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)) = 0$$

我们将使用上述KKT条件，从最优解α得到最优的$(b, \mathbf{w})$。

3. Solving Dual SVM: 解对偶问题

在上一节中我们通过推导得到了SVM的对偶形式，即**Standard hard-margin SVM dual**：

standard hard-margin SVM dual

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{z}_n^T\mathbf{z}_m - \sum_{n=1}^{N}\alpha_n \\
\text{subject to} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0; \\
& \alpha_n \geq 0, \text{ for } n = 1, 2, \ldots, N
\end{aligned}
$$

(convex) QP of N variables & $N+1$ constraints, as promised

该问题的确有N个变量(每个样本对应一个α)，N+1个约束条件(每个乘子大于等于0，并且有一个dual-inner optimal条件)。可见，**该Dual问题依然是一个QP问题**。因此，依然可以像解决原始问题的QP问题那样，将数据构造好后传入相应程序中求解：

optimal $\boldsymbol{\alpha} = ?$

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{z}_n^T\mathbf{z}_m \\
& -\sum_{n=1}^{N}\alpha_n \\
\text{subject to} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0; \\
& \alpha_n \geq 0, \\
& \text{for } n = 1, 2, \ldots, N
\end{aligned}
$$

optimal $\boldsymbol{\alpha} \leftarrow \text{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\mathrm{Q}\boldsymbol{\alpha} + \mathbf{p}^T\boldsymbol{\alpha} \\
\text{subject to} \quad & \mathbf{a}_i^T\boldsymbol{\alpha} \geq c_i, \\
& \text{for } i = 1, 2, \ldots
\end{aligned}
$$

- $q_{n,m} = y_n y_m \mathbf{z}_n^T\mathbf{z}_m$
- $\mathbf{p} = -\mathbf{1}_N$
- $\mathbf{a}_{\geq} = \mathbf{y}$, $\mathbf{a}_{\leq} = -\mathbf{y}$; $\mathbf{a}_n^T = n\text{-th unit direction}$
- $c_{\geq} = 0$, $c_{\leq} = 0$; $c_n = 0$

这里需要注意矩阵Q。矩阵Q的元素为：$q_{n,m} = y_n y_m \mathbf{z}_n^T\mathbf{z}_m$，该值往往非零——即矩阵是Dense的——会导致存储该矩阵需要耗费大量空间——如果N等于3万，那么dense的Q_D需要占据超过3G的内存！因此实践中，有一些专门为解SVM问题而设计的QP程序，推荐使用它们来解SVM的dual问题。

得到最优解α后，可根据KKT条件计算最优解$(b, \mathbf{w})$，因为这里的α只是中间产物：

- 解optimal $\mathbf{w}$，用dual-inner optimal条件：$\mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n$
- 解optimal b，用primal-inner optimal条件：$\alpha_n(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b)) = 0$，选择$\alpha_n > 0$的样本点，那么必有$1 = y_n(\mathbf{w}^T\mathbf{z}_n + b)$，两边同时乘以$y_n$并化简得：$b = y_n - \mathbf{w}^T\mathbf{z}_n$

我们将$\alpha_n > 0$的样本点，称为**支撑向量(support vector)**。由primal-inner optimal条件可知，支撑向量必定满足$y_n(\mathbf{w}^T\mathbf{z}_n + b) = 1$——即在边界上。

4. Messages behind Dual SVM: 对偶SVM背后的信息

上一节最后，我们将$\alpha_n > 0$的样本点，称为**支撑向量(support vector)**，它们位于边界上。但需要注意，位于边界上的样本并不一定是支撑向量：

$$SV(positive\ \alpha_n) \subseteq SV\ candidates(on\ boundary)$$

SV是“有用”的数据，其他样本是“无用”的数据，因为：

- 计算$\mathbf{w}$仅需要SV：$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n = \sum_{SV} \alpha_n y_n \mathbf{z}_n$
- 计算b仅需要SV：$b = y_n - \mathbf{w}^T \mathbf{z}_n$，其中$(\mathbf{z}_n, y_n)$是任意一个SV

综上，SVM是——learn **fattest hyperplane** by identifying **support vectors** with **dual** optimal solution。

观察SVM对偶的解的形式：

$$\mathbf{w}_{SVM} = \sum_{n=1}^{N} \alpha_n (y_n \mathbf{z}_n)$$

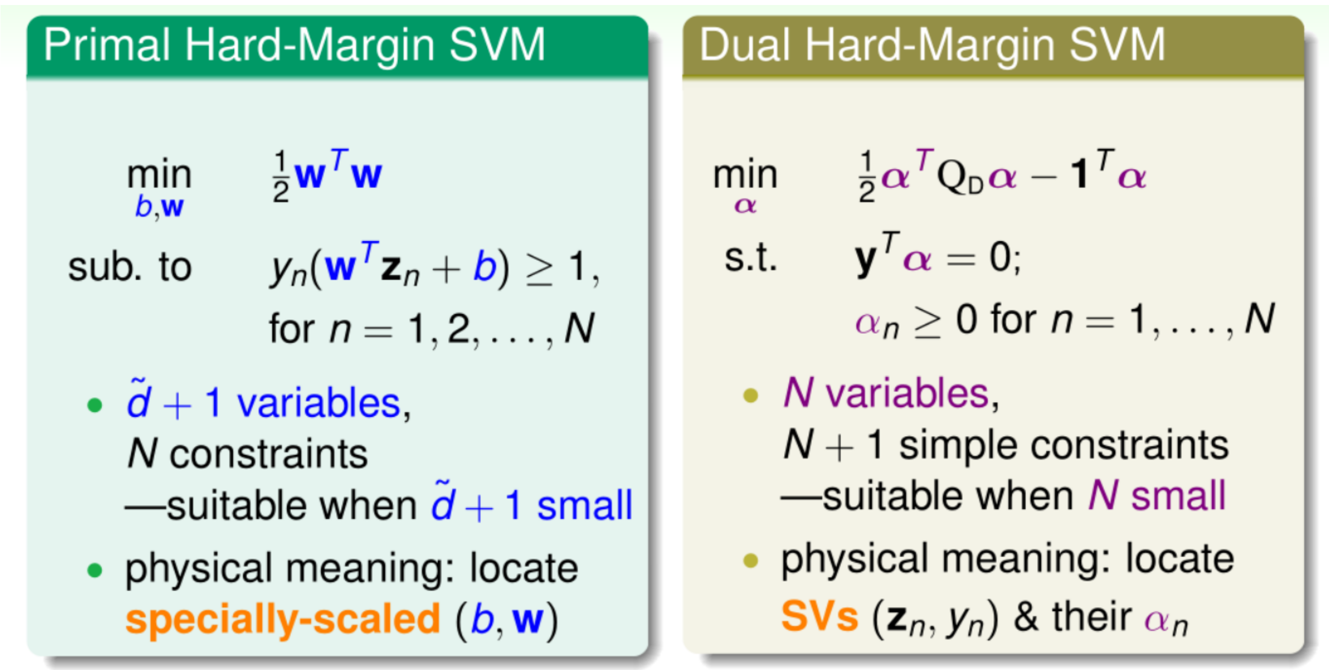
最优的$\mathbf{w}$被所有样本$\mathbf{z}_n$“表示”了出来，“表示”的系数是α_n，来自于解dual问题。这种形式其实并不陌生——在PLA中，解可以写作：

$$\mathbf{w}_{PLA} = \sum_{n=1}^{N} \beta_n (y_n \mathbf{z}_n)$$

也是一种“表示”的形式，其中“表示”的系数是β_n，表示每个点被更正的次数(# mistake corrections)。

对于这种形式的解，我们将之称为**represented by data**。对于SVM，更确切的应该是**represented by SVs only**。

现在，我们可以将Hard-Margin SVM的原始形式(primal)与对偶形式(dual)放在一起比较：



两者对于同一问题的解是一样的：

$$g_{SVM}(\mathbf{x}) = sign(\mathbf{w}^T \Phi(\mathbf{x}) + b)$$

至此，我们结束了吗？回归我们的目标——移除Z空间中SVM计算对于$\tilde{d}$的依赖。这里，我们将问题转化为了N个变量，N+1个约束的最优化问题，看似摆脱了对Z空间维度的依赖。然而，注意Q_D矩阵中元素的计算方式：

$$q_{n,m} = y_n y_m \mathbf{z}_n^T \mathbf{z}_m$$

可见，该元素的计算仍然需要在$\tilde{d}$维空间中进行，计算复杂度为$O(\tilde{d})$。因此，我们实际上还是没能摆脱计算对于$\tilde{d}$的依赖，仍然需要计算很高很高维度的内积。

5. Summary

- SVM的对偶问题将变量数目变为N个，约束条件数目变为N+1个；
- SVM最优化问题的primal与dual满足strong duality的关系，因此可以等价解dual，得到SVM的对偶形式；
- KKT条件：primal feasible，dual feasible，dual-inner optimal，primal-inner optimal；其中primal-inner optimal又称为complementary slackness，十分重要；
- 乘子大于0的样本点是支撑向量，它们位于边界上；
- 对偶问题依然是QP问题，可以递交给专门的程序处理；
- 至此，依然未摆脱对Z空间维度数的依赖：在QP问题Q矩阵的元素计算中，依然需要在Z空间内进行内积运算。