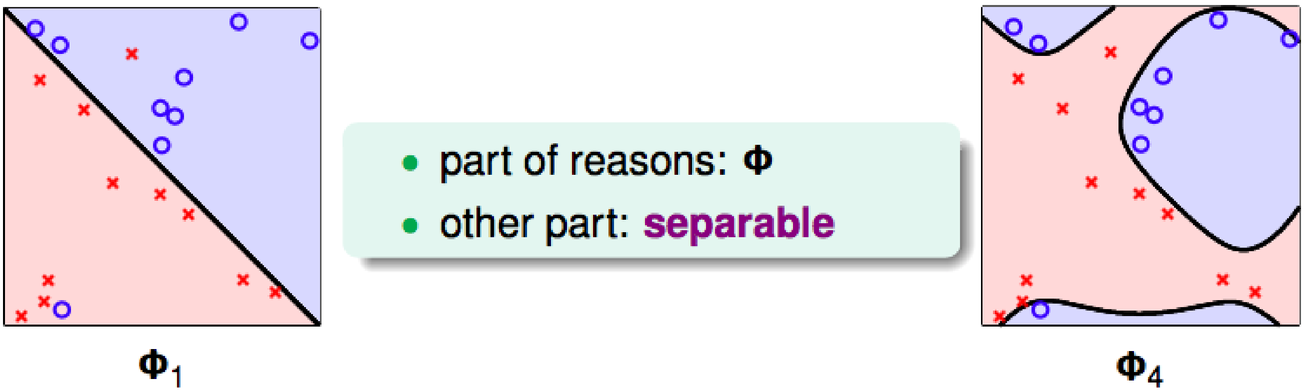
Lecture 4：Soft-Margin Support Vector Machine

> 课件链接：[Hsuan-Tien Lin - soft-margin support vector machine]
>
> **Soft-Margin Support Vector Machine(软间隔支撑向量机)**
>
> - Motivation and Primal Problem: 动机与原始问题
> - Dual Problem: 对偶问题
> - Messages behind Soft-Margin SVM: 软间隔SVM背后的信息
> - Model Selection: 模型选择

1. Motivation and Primal Problem: 动机与原始问题

动机

即使SVM试图做到最大间隔，但仍然可能过拟合。原因之一是使用了如rbf核函数的强大的特征转换，另一个原因是——**坚持将所有资料分开(separable)**。如下图所示：



右侧的图即坚持将所有的圈圈叉叉分开而不犯任何错误，左侧则犯了少数几个错误。但我们显然会认为左侧的分离超平面更好——右侧的过拟合了。因此，我们将放弃Hard-Margin(不犯任何错误)，选择Soft-Margin(犯一些错误)。

如何放弃"不犯错误"？借鉴pocket算法

Pocket算法试图解决的最优化问题为，即寻找犯错最少的分离超平面：

$$\min_{b,\mathbf{w}} \sum_{n=1}^{N} I\left[y_n \neq sign(\mathbf{w}^T\mathbf{z}_n + b)\right]$$

而Hard-Margin SVM的最优化问题为：

$$\begin{aligned} \min_{b,\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ s.t \quad & y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 \ for \ all \ n \end{aligned}$$

我们可以将Pocket算法中"尽量少犯错"的思想整合进Hard-Margin中：对于没犯错的点，我们要求Large-Margin；对于犯错的点，那就"随它去"。但是，犯错的点要尽量少，这体现在最小化的目标函数里：

$$\begin{aligned}\min_{b,\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\cdot\sum_{n=1}^{N} I\Big[y_n \neq sign(\mathbf{w}^T\mathbf{z}_n + b)\Big] \\ s.t \quad & y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 \ for\ correct\ n \\ & y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq -\infty \ for\ incorrect\ n\end{aligned}$$

其中，参数C是权衡系数，权衡的是large margin & noise tolerance。我们可将上面的最优化问题的约束条件写成一个：

$$\begin{aligned}\min_{b,\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\cdot\sum_{n=1}^{N} I\Big[y_n \neq sign(\mathbf{w}^T\mathbf{z}_n + b)\Big] \\ s.t \quad & y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 - \infty\cdot I\Big[y_n \neq sign(\mathbf{w}^T\mathbf{z}_n + b)\Big]\end{aligned}$$

然而，该最优化问题有两个缺点：

1. 目标函数与约束条件中存在布林运算，不是线性函数，因此整个最优化问题不再是QP问题——dual，kernel都无法使用；
2. 无法区分"小错误"与"大错误"——错分的样本如果离边界比较近，应该是小错误；离边界很远，那肯定是大错误。

因此，我们引进新的变量ξ_n，用来记录每个样本点的**margin violation**(对间隔的违反)：

$$\begin{aligned}\min_{b,\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\cdot\sum_{n=1}^{N} \xi_n \\ s.t \quad & y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 - \xi_n, \quad n = 1,\cdots,N \\ & \xi_n \geq 0, \quad n = 1,\cdots,N\end{aligned}$$

参数C的权衡作用：

- 大的C：希望少犯错；
- 小的C：可以犯错，margin大一点——**正则化。**

上述最优化问题是一个QP问题，有$\tilde{d}+1+N$个变量与2N个约束条件。下一节我们将求解其对偶问题，即Soft-Margin SVM dual。

2. Dual Problem: 对偶问题

根据primal问题构造拉格朗日函数：

$$\begin{aligned}\mathcal{L}(b,\mathbf{w},\xi,\alpha,\beta) = & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\cdot\sum_{n=1}^{N}\xi_n \\ & + \sum_{n=1}^{N}\alpha_n\cdot(1-\xi_n - y_n(\mathbf{w}^T\mathbf{z}_n + b)) + \sum_{n=1}^{N}\beta_n\cdot(-\xi_n)\end{aligned}$$

拉格朗日对偶问题为：

$$\max_{\alpha_n \geq 0, \beta_n \geq 0} \left(\min_{b, \mathbf{w}, \xi} \mathcal{L}(b, \mathbf{w}, \xi, \alpha, \beta) \right)$$

对于内层优化问题，令(KKT条件之一):

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 = C - \alpha_n - \beta_n$$

因此，我们可以用α_n表示β_n：$\beta_n = C - \alpha_n$，但约束条件需变更为:

$$0 \leq \alpha_n \leq C$$

如此替换，我们还可以顺便将ξ_n消去:

ξ can also be removed :-), like how we removed b

$$\max_{0 \leq \alpha_n \leq C,\ \beta_n = C - \alpha_n} \left(\min_{b, \mathbf{w}, \xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\mathbf{w}^T\mathbf{z}_n + b)) \right.$$
$$\left. \cancel{+ \sum_{n=1}^{N} (C - \alpha_n - \beta_n) \cdot \xi_n} \right)$$

得到:

$$\max_{0 \leq \alpha_n \leq C, \beta_n = C - \alpha_n} \left(\min_{b, \mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\mathbf{w}^T\mathbf{z}_n + b)) \right)$$

内层最优化问题与hard-margin SVM的dual一模一样，因此我们同样对$\mathbf{w}$与b偏导置零，得到相同的结果:

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

最终，我们得到**Soft-Margin SVM Dual**问题:

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

只有一个地方与Hard-Margin SVM Dual不一样：α_n有一个**上界C**——这是由于ξ_n的拉格朗日乘子造成的。同样，该问题是一个QP问题，有N个变量和2N+1个约束条件。

3. Messages behind Soft-Margin SVM: 软间隔SVM背后的信息

Kernel Soft-Margin SVM算法：

Kernel Soft-Margin SVM Algorithm

1. $q_{n,m} = y_n y_m K(\mathbf{x}_n, \mathbf{x}_m)$; $\mathbf{p} = -\mathbf{1}_N$; $(\mathrm{A}, \mathbf{c})$ for equ./lower-bound/**upper-bound** constraints
2. $\boldsymbol{\alpha} \leftarrow \mathrm{QP}(\mathrm{Q}_\mathrm{D}, \mathbf{p}, \mathrm{A}, \mathbf{c})$
3. $b \leftarrow ?$
4. return SVs and their α_n as well as b such that for new $\mathbf{x}$,

$$g_{\mathrm{SVM}}(\mathbf{x}) = \mathrm{sign}\left(\sum_{\text{SV indices } n}\alpha_n y_n K(\mathbf{x}_n, \mathbf{x}) + b\right)$$

b怎么求?

在Hard-Margin SVM中，我们通过complementary slackness：

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

寻找$\alpha_s > 0$的SV；对于它来说，上式中的另一项一定为0，因此：

$$b = y_s - \mathbf{w}^T\mathbf{z}_s$$

在Soft-Margin SVM中，我们依然从complementary slackness中寻找突破口：

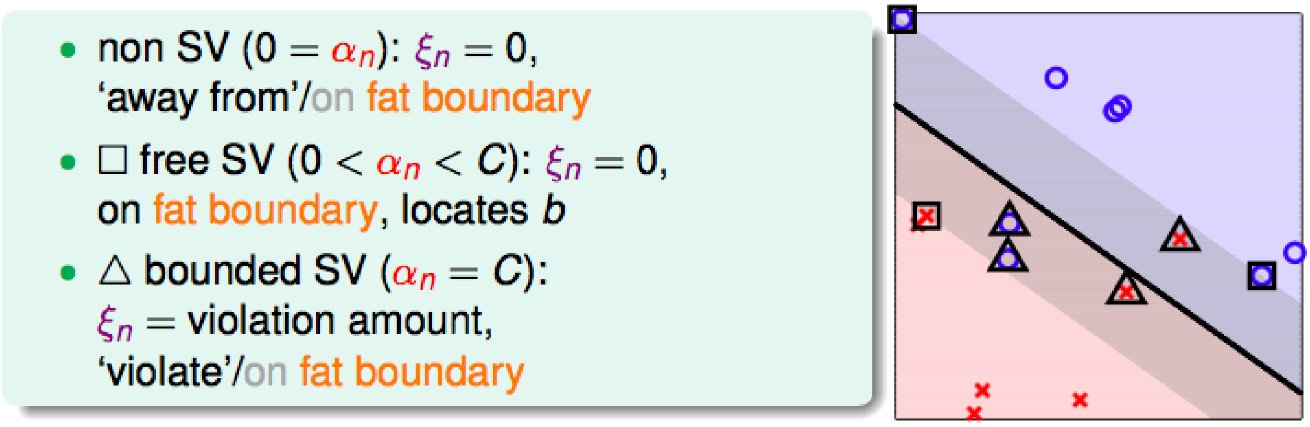
$$
\begin{aligned}
\alpha_n(1-\xi_n-y_n(\mathbf{w}^T\mathbf{z}_n+b)) &= 0 \\
(C-\alpha_n)\xi_n &= 0
\end{aligned}
$$

我们要找的是不再仅仅是SV($\alpha_n > 0$)，还要是free SV($0 < \alpha_n < C$)，这样第一个式子的另一项为0，第二个式子的ξ_n是0，那么：

$$b = y_s - \mathbf{w}^T\mathbf{z}_s$$

这里我们再次强调α_n的含义。根据α_n，样本可分为三类：

- $\alpha_n = 0$:
    - non SV，不是支撑向量；
    - $\xi_n = 0$;
    - 在边界外(罕见在边界上)；
- $0 < \alpha_n < C$:
    - free SV，自由支撑向量；
    - $\xi_n = 0$;
    - 在边界上；
- $\alpha_n = C$:
    - bounded SV，受限支撑向量；
    - $\xi_n \geq 0$;
    - 违反了边界(罕见在边界上)。



4. Model Selection: 模型选择

对于Kernel Soft-Margin SVM using rbf kernel，需要选择参数(C, γ)。常用的方法是**交叉验证**，即计算Cross Validation Error——E_{cv}

使用N(样本数)折交叉验证时，$E_{cv} = E_{loocv}$；对于SVM来说，有：

$$E_{loocv} \leq \frac{\#SV}{N}$$

下面我们简单证明该不等式：

对于某个non-SV，其α为0。对于使用去除该non-SV后的样本集进行训练而得到的g^-中的各α，应较原来g中相应的α，没有变化。这是因为，如果不一样，说明训练g^-时发现了一组新的α，使得目标函数的值较g的更小，那么我们可以将该组α配上$\alpha_i = 0$构成在整个样本集上的α组合，该组合一定会比原来的组合在目标函数上的值更小。这与原来的组合是最优解矛盾。证毕。

当留下的是non-SV时，$g^- = g$。因此：

$$
\begin{aligned}
e_{non-SV} &= err(g^-, non-SV) \\
&= err(g, non-SV) \\
&= 0
\end{aligned}
$$

而：

$$
e_{SV} \le 1
$$

综上：

$$
E_{loocv} = \sum e \le \frac{\#SV}{N}
$$

然而，这种方法仅仅是给出了交叉验证误差的上界，并不是一个精确的判断标准。实务上，用这种方法进行初步的safety check——SV多的"很可能"不好，然后再计算交叉验证误差。

5. Summary

- Soft-Margin的动机是避免过拟合，放弃Hard-Margin将训练样本完全100%正确分开的思想；
- 在Hard-Margin SVM中引入松弛变量ξ_n记录每个样本点的违反情况，并将其纳入目标函数最小化的范畴中，用参数C进行调节——C越小，正则化效果越高；
- Soft-Margin Dual与Hard-Margin Dual十分相像，仅约束条件中α_n有上界C；
- Soft-Margin Dual的最优解α将样本点分为了三类：non SV，free SV与Bounded SV；
- 对于Soft-Margin Dual，又多了一个权衡参数C需要选择，常使用交叉验证法；同时，可以辅助SV数目进行判断——SV越多，模型的泛化性能很可能越差。