超详细SVM（支持向量机）知识点，面试官会问的都在这了。。。

前言：持续准备面试中。。。准备的过程中，慢慢发现，如果死记硬背的话很难，可当推导一遍并且细细研究里面的缘由的话，面试起来应该什么都不怕，问什么问题都可以由公式推导得到结论，不管问什么，公式摆在那里，影响这个公式的变量就在那，你问什么我答什么。。共勉！！

一. 简单概括一下SVM：

SVM 是一种二类分类模型。它的基本思想是在特征空间中寻找间隔最大的分离超平面使数据得到高效的二分类，具体来讲，有三种情况（不加核函数的话就是个线性模型，加了之后才会升级为一个非线性模型）：

- 当训练样本线性可分时，通过硬间隔最大化，学习一个线性分类器，即线性可分支持向量机；
- 当训练数据近似线性可分时，引入松弛变量，通过软间隔最大化，学习一个线性分类器，即线性支持向量机；
- 当训练数据线性不可分时，通过使用核技巧及软间隔最大化，学习非线性支持向量机。

二. SVM 为什么采用间隔最大化（与感知机的区别）：

当训练数据线性可分时，存在无穷个分离超平面可以将两类数据正确分开。感知机利用误分类最小策略，求得分离超平面，不过此时的解有无穷多个。线性可分支持向量机利用间隔最大化求得最优分离超平面，这时，解是唯一的。另一方面，此时的分隔超平面所产生的分类结果是最鲁棒的，对未知实例的泛化能力最强。

三. SVM的目标（硬间隔）：

有两个目标：第一个是**使间隔最大化**，第二个是**使样本正确分类**，由此推出目标函数：

目标一（使间隔最大化）：$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$

目标二（使样本正确分类）：$y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, i = 1, 2, \ldots, m$

稍微解释一下，w是超平面参数，目标一是从点到面的距离公式化简来的，具体不展开，目标二就相当于感知机，只是把

大于等于0进行缩放变成了大于等于1，为了后面的推导方便。有了两个目标，写在一起，就变成了svm的终极目标：

终极目标：
$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &\text{s.t. } y_i\left(w^T x_i + b\right) \geq 1, \forall i \end{aligned}$$

四. 求解目标（硬间隔）：

从上面的公式看出，这是一个有约束条件的最优化问题，用拉格朗日函数来解决。

上式的拉格朗日函数为：

$$\begin{aligned} min_{w,b} max_{\alpha} L(\omega, b, \alpha) = \frac{1}{2}\|\omega\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\omega^T x_i + b\right)\right) \\ s.t. \alpha_i \geq 0, \forall i \end{aligned}$$

在满足Slater定理的时候，且过程满足KKT条件的时候，原问题转换成对偶问题：

$$\begin{aligned} max_{\alpha} min_{w,b} L(\omega, b, \alpha) = \frac{1}{2}\|\omega\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\omega^T x_i + b\right)\right) \\ s.t. \alpha_i \geq 0, \forall i \end{aligned}$$

先求内部最小值，对 ω 和 b 求偏导并令其等于 0 可得：

$$\begin{aligned} w &= \sum_{i=1}^{m} \alpha_i y_i x_i, \\ 0 &= \sum_{i=1}^{m} \alpha_i y_i. \end{aligned}$$

将其代入到上式中去可得到

$$max_{\alpha} L(\omega, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0 （\alpha_i \geq 0, i = 1, 2, \ldots, m）$$

此时需要求解 α，**利用SMO（序列最小优化）算法：**

SMO算法的基本思路是每次选择两个变量α_i和α_j，选取的两个变量所对应的样本之间间隔要尽可能大，因为这样更新会带给目标函数值更大的变化。**SMO算法之所以高效，是因为仅优化两个参数的过程实际上仅有一个约束条件，其中一个可由另一个表示，这样的二次规划问题具有闭式解。**

五. 软间隔：

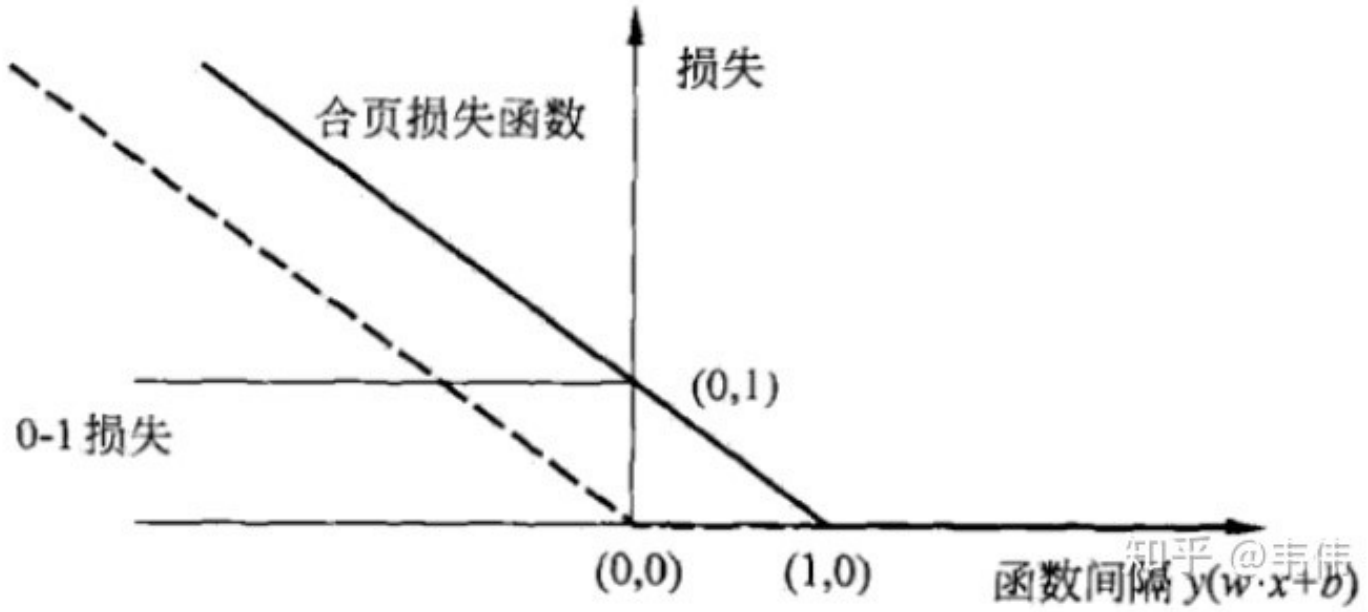
不管直接在原特征空间，还是在映射的高维空间，我们都假设样本是线性可分的。虽然理论上我们总能找到一个高维映射使数据线性可分，但在实际任务中，寻找一个合适的核函数核很困难。此外，由于数据通常有噪声存在，一味追求数据线性可分可能会使模型陷入过拟合，因此，**我们放宽对样本的要求，允许少量样本分类错误**。这样的想法就意味着对目标函数的改变，之前推导的目标函数里不允许任何错误，并且让间隔最大，现在给之前的目标函数加上一个误差，就相当于允许原先的目标出错，引入松弛变量 $\xi_i \geq 0$，公式变为：

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \xi_i$$

那么这个松弛变量怎么计算呢，最开始试图用0，1损失去计算，但0，1损失函数并不连续，求最值时求导的时候不好求，所以引入合页损失（hinge loss）：

$$l_{hinge}(z) = \max(0, 1 - z)$$

函数图张这样：



理解起来就是，原先制约条件是保证所有样本分类正确，$y_i\left(w^Tx_i+b\right)\geq 1,\forall i$，现在出现错误的时候，一定是这个式子不被满足了，即$y_i\left(w^Tx_i+b\right)<1,\forall i_{错误}$，衡量一下错了多少呢？因为左边一定小于1，那就跟1比较，因为1是边界，所以用1减去$y_i\left(w^Tx_i+b\right)$来衡量错误了多少，所以目标变为（正确分类的话损失为0，错误的话付出代价）：

$$\min_{w,b}\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}max(0,1-y_i\left(w^Tx_i+b\right))$$

但这个代价需要一个控制的因子，引入C>0，惩罚参数，即：

$$\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}max(0,1-y_i\left(w^Tx_i+b\right))$$

可以想象，C越大说明把错误放的越大，说明对错误的容忍度就小，反之亦然。当C无穷大时，就变成一点错误都不能容忍，即变成硬间隔。实际应用时我们要合理选取C，C越小越容易欠拟合，C越大越容易过拟合。

所以软间隔的目标函数为：

$$\begin{aligned}&\min_{w,b,\xi}\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i\\&\text{s.t. } y_i\left(x_i^Tw+b\right)\geq 1-\xi_i\\&\xi_i\geq 0, i=1,2,\dots n\end{aligned}$$

其中：

$$\xi_i = max(0, 1-y_i\left(w^Tx_i+b\right))$$

六. 软间隔求解：

与硬间隔类似：

上式的拉格朗日函数为：

$$min_{w,b,\xi}max_{\alpha,\beta}L(\omega,b,\alpha,\xi,\beta)=\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{n}\xi_i+\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\omega^Tx_i+b\right)-\xi_i\right)-\sum_{i=1}^{n}\beta_i\xi_i$$
$$s.t.\ \alpha_i\geq 0且\beta_i\geq 0,\forall i$$

在满足Slater定理的时候，且过程满足KKT条件的时候，原问题转换成对偶问题：

$$max_{\alpha,\beta}min_{w,b,\xi}L(\omega,b,\alpha,\xi,\beta)=\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{n}\xi_i+\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\omega^Tx_i+b\right)-\xi_i\right)-\sum_{i=1}^{n}\beta_i\xi_i$$
$$s.t.\ \alpha_i\geq 0且\beta_i\geq 0,\forall i$$

先求内部最小值，对 ω , b 和 ξ 求偏导并令其等于 0 可得：

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i,$$

$$0 = \sum_{i=1}^{m} \alpha_i y_i.$$

$$C = \alpha_i + \beta_i$$

将其代入到上式中去可得到，注意 β 被消掉了：

$$max_{\alpha,\beta} L(\omega, b, \alpha, \xi, \beta) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0 （0 \le \alpha_i \le C, i = 1, 2, \ldots, m）$$

此时需要求解 α，同样**利用SMO（序列最小优化）算法。**

七. 核函数：

为什么要引入核函数：

> 当样本在原始空间线性不可分时，可将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分。而引入这样的映射后，所要求解的对偶问题的求解中，无需求解真正的映射函数，而只需要知道其核函数。核函数的定义：K(x,y)=<ϕ(x),ϕ(y)>，即在特征空间的内积等于它们在原始样本空间中通过核函数 K 计算的结果。一方面数据变成了高维空间中线性可分的数据，另一方面不需要求解具体的映射函数，只需要给定具体的核函数即可，这样使得求解的难度大大降低。

用自己的话说就是，在SVM不论是硬间隔还是软间隔在计算过程中，都有X转置点积X，若X的维度低一点还好算，但当我们想把X从低维映射到高维的时候（让数据变得线性可分时），这一步计算很困难，等于说在计算时，需要先计算把X映射到高维的的ϕ(x)，再计算ϕ(x1)和ϕ(x2)的点积，这一步计算起来开销很大，难度也很大，此时引入核函数，这两步的计算便成了一步计算，即只需把两个x带入核函数，计算核函数，举个列子一目了然（图片来自：从零推导支持向量机）：

引理 19. 映射

$$\phi\colon x \mapsto \exp(-x^2)\begin{bmatrix} 1 \\ \sqrt{\frac{2}{1}}x \\ \sqrt{\frac{2^2}{2!}}x^2 \\ \vdots \end{bmatrix} \tag{41}$$

对应于核函数

$$\kappa(x_i, x_j) := \exp(-(x_i - x_j)^2)\,. \tag{42}$$

证明.

$$\begin{aligned}
\kappa(x_i, x_j) &= \exp(-(x_i - x_j)^2) \\
&= \exp(-x_i^2)\exp(-x_j^2)\exp(2x_i x_j) \\
&= \exp(-x_i^2)\exp(-x_j^2)\sum_{k=0}^{\infty}\frac{(2x_i x_j)^k}{k!} \\
&= \sum_{k=0}^{\infty}\left(\exp(-x_i^2)\sqrt{\frac{2^k}{k!}}x_i^k\right)\left(\exp(-x_j^2)\sqrt{\frac{2^k}{k!}}x_j^k\right) \\
&= \boldsymbol{\phi}(x_i)^\top \boldsymbol{\phi}(x_j)\,.
\end{aligned} \tag{43}$$

个人对核函数的理解：核函数就是一个函数，接收两个变量，这两个变量是在低维空间中的变量，而**核函数求的值**等于**将两个低维空间中的向量映射到高维空间后的内积**。

八. 如何确定一个函数是核函数：

验证正定核啥的，咱也不太懂，给出：

> 设$\mathcal{X} \subset R^n$,$K(x,z)$是定义在$\mathcal{X} \times \mathcal{X}$上的对称函数，如果对任意的$x_i \in \mathcal{X}, i = 1,2,...,m$，$K(x,z)$对应的Gram矩阵$K = [K(x_i, x_j)]_{m \times m}$是半正定矩阵，则$K(x,z)$是正定核

所以不懂，就用人家确定好的常见核函数及其优缺点：

名称	形式	优点	缺点
线性核	$x_i^\top x_j$	有高效实现，不易过拟合	无法解决非线性可分问
多项式核	$(\beta x_i^\top x_j + \theta)^n$	比线性核更一般，n 直接描述了被映射空间的复杂度	参数多，当 n 很大时会
RBF 核	$\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$	只有一个参数，没有计算不稳定问题	计算慢，过拟合风险大

九. 如何选择核函数：

- 当特征维数 d 超过样本数 m 时 (文本分类问题通常是这种情况), 使用线性核;
- 当特征维数 d 比较小. 样本数 m 中等时, 使用RBF核;
- 当特征维数 d 比较小. 样本数 m 特别大时, 支持向量机性能通常不如深度神经网络

十. 关于支持向量的问题：

1.先说硬间隔：

先看KKT条件

定理 14 (线性支持向量机的 KKT 条件). 线性支持向量机的 KKT 条件如下.

- 主问题可行: $1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq 0$;
- 对偶问题可行: $\alpha_i \geq 0$;
- 互补松弛: $\alpha_i(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) = 0$.

支持向量，对偶变量 αi > 0 对应的样本；

- **线性支持向量机中, 支持向量是距离划分超平面最近的样本, 落在最大间隔边界上。**

证明. 由线性支持向量机的 KKT 条件可知, $\alpha_i(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) = 0$. 当 $\alpha_i > 0$ 时, $1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 0$. 即 $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$.

- **支持向量机的参数 (w; b) 仅由支持向量决定, 与其他样本无关。**

证明. 由于对偶变量 $\alpha_i > 0$ 对应的样本是支持向量,

$$\begin{aligned}\boldsymbol{w} &= \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \\ &= \sum_{i:\ \alpha_i=0}^{m} 0 \cdot y_i \boldsymbol{x}_i + \sum_{i:\ \alpha_i>0}^{m} \alpha_i y_i \boldsymbol{x}_i \\ &= \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i \,, \end{aligned} \tag{35}$$

其中 SV 代表所有支持向量的集合. b 可以由互补松弛松弛算出. 对于某一支持向量 $\boldsymbol{x}_s$ 及其标记 y_s, 由于 $y_s(\boldsymbol{w}^\top \boldsymbol{x}_s + b) = 1$, 则

$$b = y_s - \boldsymbol{w}^\top \boldsymbol{x}_s = y_s - \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x}_s \,. \tag{36}$$

实践中, 为了得到对 b 更稳健的估计, 通常使用对所有支持向量求解得到 b 的平均值.

2. 再说软间隔：

先看kkt条件：

定理 27 (软间隔支持向量机的 KKT 条件). 软间隔支持向量机的 KKT 条件如下.

- 主问题可行: $1-\xi_i-y_i(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_i)+b)\leq 0$, $-\xi_i\leq 0$;
- 对偶问题可行: $\alpha_i\geq 0$, $\beta_i\geq 0$;
- 互补松弛: $\alpha_i(1-\xi_i-y_i(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_i)+b))=0$, $\beta_i\xi_i=0$.

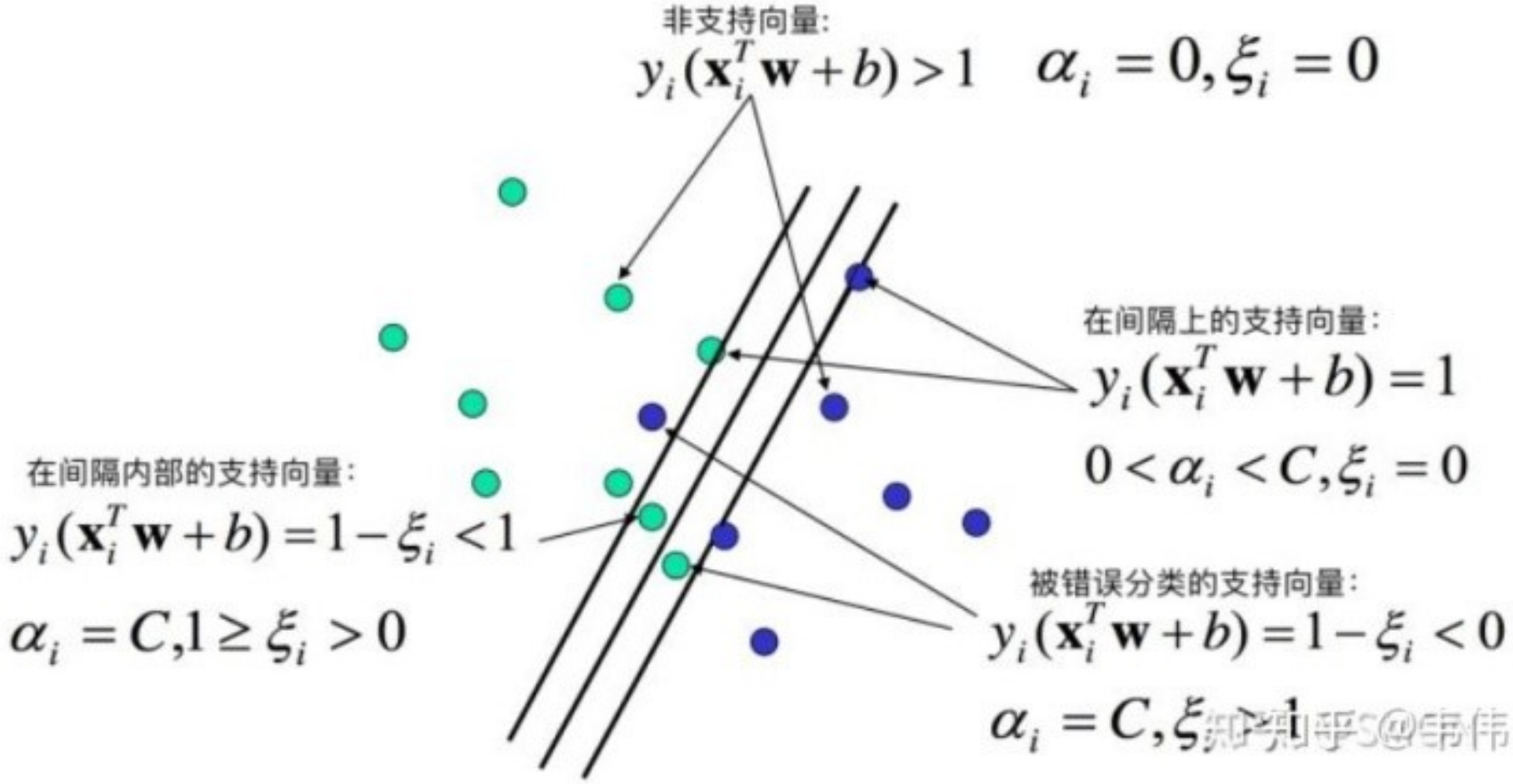
经过SMO后，求得 $\hat{\boldsymbol{\alpha}}$，$\mathbf{0}<\hat{\boldsymbol{\alpha}}_{\boldsymbol{j}}<\boldsymbol{C}$ 。

对于任意样本$(\boldsymbol{X_i},\boldsymbol{y_i})$,

若$\boldsymbol{\alpha_i}=\mathbf{0}$，此样本点不是支持向量，该样本对模型没有任何的作用
若$\boldsymbol{\alpha_i}>\mathbf{0}$，此样本是一个支持向量（同硬间隔）

若满足$\boldsymbol{\alpha_i}>\mathbf{0}$，进一步地，

如图：



十一. 谈谈SVM的损失函数：

此处说的是软间隔：

先看软间隔的基本型形式：

$$\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}max(0,1-y_i\left(w^Tx_i+b\right))$$

稍微做一点变化：

$$\min_{w,b} \frac{1}{m} \sum_{i=1}^{m} (\max 0, 1 - y_i \left(w^\top \phi(x_i) + b\right) + \frac{\lambda}{2} \|w\|^2$$

这样写是为了符合标准的**损失函数+正则化**的样子，其中, 第一项称为经验风险, 度量了模型对训练数据的拟合程度; 第二项称为结构风险, 也称为正则化项, 度量 了模型自身的复杂度. 正则化项削减了假设空间, 从而 降低过拟合风险. λ 是个可调节的超参数, 用于权衡经验风险和结构风险.

其中：

$$\xi_i = \max\left(0, 1 - y_i \left(w^\top \phi(x_i) + b\right)\right) \geq 0$$

$$\lambda = \frac{1}{mC}$$

这样的话给上式乘以mc，就会变成上上式了。

十二. 为什么SVM对缺失数据敏感？

这里说的缺失数据是指缺失某些特征数据，向量数据不完整。SVM 没有处理缺失值的策略。而 SVM 希望样本在特征空间中线性可分，所以特征空间的好坏对SVM的性能很重要。缺失特征数据将影响训练结果的好坏。

十三. SVM的优缺点：

优点：

1. 由于SVM是一个凸优化问题，所以求得的解一定是全局最优而不是局部最优。
2. 不仅适用于线性线性问题还适用于非线性问题(用核技巧)。
3. 拥有高维样本空间的数据也能用SVM，这是因为数据集的复杂度只取决于支持向量而不是数据集的维度，这在某种意义上避免了“维数灾难”。
4. 理论基础比较完善(例如神经网络就更像一个黑盒子)。

缺点：

1. 二次规划问题求解将涉及m阶矩阵的计算(m为样本的个数), 因此SVM不适用于超大数据集。(SMO算法可以缓解这个问题)
2. 只适用于二分类问题。(SVM的推广SVR也适用于回归问题；可以通过多个SVM的组合来解决多分类问题)

十四：参考文献：