SVM：

问 1：SVM 如何实现的最大分离间隔？

SVM 的优化目标是（其中对目标函数进行了函数间隔的归一化）

$$\min \ \|w\| \quad s.t. \ y_i(w^T x_i + b) \geq 1$$

由于要求 $w^T x_i = \|w\|\|x_i\|\cos < w, x_i >$ 必须尽可能地大，$\|w\|$ 尽可能地小，而 $\|x_i\|$ 一定，所以可得 $< w, x_i >= 0$。说明此时超平面的法向量应该与样本同向，而这就造成求出的超平面一定满足最大分离间隔。

还有另一个角度：最小化 $\|w\|$ 相当于最大化 $1/\|w\|$，由于此时函数距离为 1，因此 $1/\|w\|$ 就是样本的几何间隔。因此最后的结果一定满足最大分离间隔。

问 2：SVM 的具体算法流程和操作细节。

1、使用拉格朗日乘子法将目标函数约束条件加到目标函数中，乘子为 α

2、交换最大最小化顺序，先最小 w,b，再最大化 α。其中求导赋零可以得到

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \ , \ \sum_{i=1}^{m} \alpha_i y_i = 0 \ , \ b = -\frac{\max\limits_{i:y_i=-1} w^T x_i + \min\limits_{i:y_i=1} w^T x_i}{2}$$

这样超平面

$$f(x) = \left(\sum_{i=1}^{m} \alpha_i y_i x_i \right)^T x + b = \sum_{i=1}^{m} \alpha_i y_i \langle x_i, x \rangle + b$$

只表示为新样本与支持向量之间的内积。

3、使用 SMO（启发式的选择一对 α_i, α_j，固定其他 α）迭代求解 α

问 3：SVM 核函数给出的是原空间的内积，如何实现空间映射，具体操作如何使用的？

计算两个向量在隐式映射过后的空间中的内积的函数叫做核函数(Kernel Function)，因此 SVM 给出的核函数 $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ 就是映射到的高维空

间上的内积。根据定义的内积很难得到映射函数 $\phi(x)$，好在整个计算过程本不需要 $\phi(x)$。

当获得一个新的样本，对应超平面方程

$$f(x)=\sum_{i=1}^{m}\alpha_i y_i K\left(x_i,x\right)+b$$

就可直接判断。由于只有少数支持向量的 α_i 不为零，因此可能很快得得到结果。

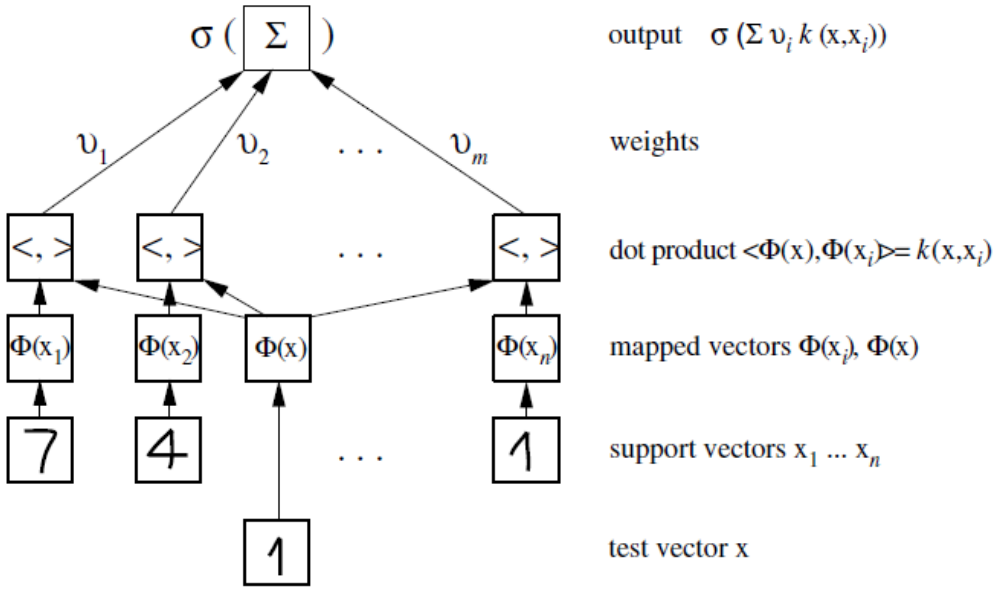

Figure 1.9 Architecture of SVMs and related kernel methods. The input x and the expansion patterns (SVs) x_i (we assume that we are dealing with handwritten digits) are nonlinearly mapped (by Φ) into a feature space $\mathcal{H}$ where dot products are computed. Through the use of the kernel k, these two layers are in practice computed in one step. The results are linearly combined using weights υ_i, found by solving a quadratic program (in pattern recognition, $\upsilon_i = y_i\alpha_i$; in regression estimation, $\upsilon_i = \alpha_i^* - \alpha_i$) or an eigenvalue problem (Kernel PCA). The linear combination is fed into the function σ (in pattern recognition, $\sigma(x) = \mathrm{sgn}(x + b)$; in regression estimation, $\sigma(x) = x + b$; in Kernel PCA, $\sigma(x) = x$).

问 4：SMO 算法具体操作流程。

1、选择一个不满足要求的乘子 x_i，并根据规律选择另一个乘子 x_j（可简单随机）

2、计算它们预测值与真实值之差 E_i

$$E_i = \sum_{j=1}^{m} \alpha_i y_i K\left(x_j, x_i\right) + b - y_i$$

3、根据它们真实值的符号判断乘子的范围

$$\begin{cases} L = \max\left(0,\ \alpha_j^{old} - \alpha_i^{old}\right), \quad H = \min\left(C,\ C + \alpha_j^{old} - \alpha_i^{old}\right), \ \ if \ \ y_i \neq y_j \\ L = \max\left(0,\ \alpha_j^{old} + \alpha_i^{old} - C\right), \quad H = \min\left(C,\ \alpha_j^{old} + \alpha_i^{old}\right), \ \ if \ \ y_i = y_j \end{cases}$$

4、根据公式计算新的乘子，并按照范围进行截断

$$\alpha_j^{new} = \alpha_j^{old} + \frac{y_j\left(E_i - E_j\right)}{K\left(x_i, x_i\right) + K\left(x_j, x_j\right) - 2K\left(x_i, x_j\right)}$$

$$\alpha_j^{new} = \max\left(\min\left(\alpha_j^{new}, H\right), L\right)$$

$$\alpha_i^{new} = \alpha_i^{old} + y_i y_j\left(\alpha_j^{old} - \alpha_j^{new}\right)$$

5、按照公式计算 b

$$b = \begin{cases} b_1, & 0 < \alpha_1^{new} < C \\ b_2, & 0 < \alpha_2^{new} < C \\ \left(b_1 + b_2\right)/2, & otherwise \end{cases}$$

$$\begin{cases} b_1 = b - E_i - y_i\left(\alpha_i^{new} - \alpha_i^{old}\right)K\left(x_i, x_i\right) - y_j\left(\alpha_j^{new} - \alpha_j^{old}\right)K\left(x_i, x_j\right) \\ b_2 = b - E_j - y_i\left(\alpha_i^{new} - \alpha_i^{old}\right)K\left(x_i, x_j\right) - y_j\left(\alpha_j^{new} - \alpha_j^{old}\right)K\left(x_j, x_j\right) \end{cases}$$

6、根据公式得到分隔超平面

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K\left(x_i, x\right) + b$$

1、它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，其学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

2、一个线性分类器的学习目标便是要在 n 维的数据空间中找到一个超平面(hyper plane)，这个超平面的方程可以表示为（w^T 中的 T 代表转置）：

$$w^T x + b = 0$$

其实$\theta^T x = w^T x + b$。

3、几何间隔就是函数间隔除以||w||

4、对于新点 x 的预测，只需要计算它与训练数据点的内积即可（$\langle \cdot, \cdot \rangle$表示向量内积），这一点至关重要，是之后使用 Kernel 进行非线性推广的基本前提。此外，所谓 Supporting Vector 也在这里显示出来——事实上，所有非 Supporting Vector 所对应的系数 α 都是等于零的，因此对于新点的内积计算实际上只要针对少量的“支持向量”而不是所有的训练数据即可。

5、通过求解对偶问题得到最优解，这就是线性可分条件下支持向量机的对偶算法，这样做的优点在于：一者对偶问题往往更容易求解；二者可以自然的引入核函数，进而推广到非线性分类问题

6、在线性不可分的情况下，支持向量机首先在低维空间中完成计算，然后通过核函数将输入空间映射到高维特征空间，最终在高维特征空间中构造出最优分离超平面，从而把平面上本身不好分的非线性数据分开。

7、计算两个向量在隐式映射过后的空间中的内积的函数叫做核函数 (Kernel Function)，直接在原来的低维空间中进行计算，而不需要显式地写出映射后的结果。

8、高斯核 $\kappa(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$，如果 σ 选得很大的话，高次特征上的权重实际上衰减得非常快，所以实际上（数值上近似一下）相当于一个低维的子空间；反过来，如果 σ 选得很小，则可以将任意的数据映射为线性可分——当然，这并不一定是好事，因为随之而来的可能是非常严重的过拟合问题。不过，总的来说，通过调控参数 σ，高斯核实际上具有相当高的灵活性，也是使用最广泛的核函数之一。

9、SVM 它本质上即是一个分类方法，用 $w^T + b$ 定义分类函数，于是求 w、b，为寻最大间隔，引出 1/2||w||^2，继而引入拉格朗日因子，化为对拉格朗日乘子 a

的求解（求解过程中会涉及到一系列最优化或凸二次规划等问题），如此，求 w.b 与求 a 等价，而 a 的求解可以用一种快速学习算法 SMO，至于核函数，是为处理非线性情况，若直接映射到高维计算恐维度爆炸，故在低维计算，等效高维表现。

10、期望风险、经验风险、结构化风险：经验风险是局部的，基于训练集所有样本点损失函数最小化的。期望风险是全局的，是基于所有样本点的损失函数最小化的，但需要训练集的联合概率分布。经验风险函数是现实的，可求的；期望风险函数是理想化的，不可求的。经验风险越小，模型决策函数越复杂，其包含的参数越多，当经验风险函数小到一定程度就出现了过拟合现象。我们需要同时保证经验风险函数和模型决策函数的复杂度都达到最小化，一个简单的办法把两个式子融合成一个式子得到结构风险函数然后对这个结构风险函数进行最小化——正则化。

11、SMO 算法：SVM 需要处理 m 阶的矩阵，m 为训练样本数。SMO 每次只更新其中的两个拉格朗日乘子，最后如果都能够满足 KKT 条件，且多次更新均没有改变参数，那么认为已得最终结果。

12、