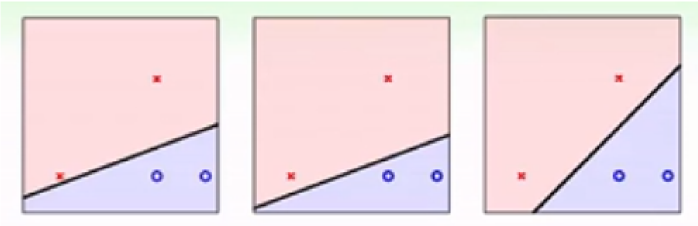
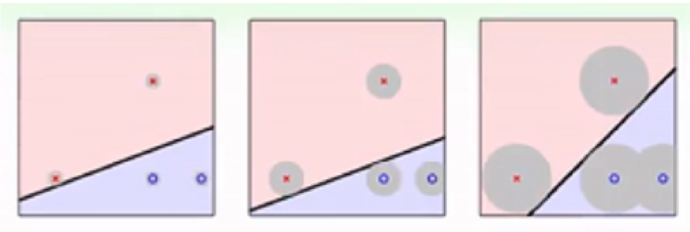
Linear Support Vector Machine

1.Large-Margin Separating Hyperplane



上图三种分割方式都可以正确分割正负点，那么怎么分辨哪种方案更好?

PLA可能会随机选择方案(最终结果与经过的错误点有关)
都满足VC bound要求，模型复杂度一样。



每个样本点距离分界线越远，就表明其对于测量误差的容忍度越高，就越安全。（PS:测量误差是一种类型的noise，而noise是导致过拟合的一个原因）

informal argument

if (Gaussian-like) noise on future $\mathbf{x} \approx \mathbf{x}_n$:

$\mathbf{x}_n$ further from hyperplane	distance to closest $\mathbf{x}_n$
$\Longleftrightarrow$ tolerate more noise	$\Longleftrightarrow$ amount of noise tolerance
$\Longleftrightarrow$ more robust to overfitting	$\Longleftrightarrow$ robustness of hyperplane

rightmost one: **more robust**
because of **larger distance to closest $\mathbf{x}_n$**

分类线由权重w决定，目的就是找到使margin最大时对应的w值。即

$$
\begin{aligned}
\max_{\mathbf{w}} \quad & \text{margin}(\mathbf{w}) \\
\text{subject to} \quad & \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0 \\
& \text{margin}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})
\end{aligned}
$$

2.Standard Large-Margin Problem

如何计算点到直线的距离?
首先，我们将权重$w(w_0, w_1, \cdots, w_d)$中的w_0拿出来，用b表示(即截距)。同时省去x_0项。这样，hypothesis就变成了$h(x) = sign(w^T x + b)$。

'shorten' x and w

distance needs w_0 and $(w_1, \ldots, w_d)$ differently (to be derived)

$$
b = w_0 \qquad \cancel{x_0 = 1}
$$

$$
\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} ; \quad \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}
$$

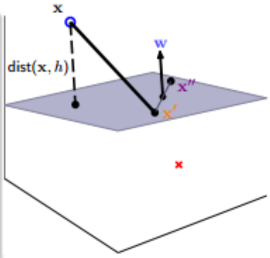
want: distance($\mathbf{x}, b, \mathbf{w}$), with hyperplane $\mathbf{w}^T\mathbf{x}' + b = 0$

consider $\mathbf{x}'$, $\mathbf{x}''$ on hyperplane

1. $\mathbf{w}^T\mathbf{x}' = -b$, $\mathbf{w}^T\mathbf{x}'' = -b$
2. $\mathbf{w} \perp$ hyperplane:

$$\left(\mathbf{w}^T \underbrace{(\mathbf{x}'' - \mathbf{x}')}_{\text{vector on hyperplane}}\right) = 0$$

3. distance = project $(\mathbf{x} - \mathbf{x}')$ to $\perp$ hyperplane



$$\text{distance}(\mathbf{x}, b, \mathbf{w}) = \left|\frac{\mathbf{w}^T}{\|\mathbf{w}\|}(\mathbf{x} - \mathbf{x}')\right| \overset{(1)}{=} \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T\mathbf{x} + b|$$

目标形式转换为：

$$\begin{aligned} \max_{b,\mathbf{w}} \quad & \text{margin}(b, \mathbf{w}) \\ \text{subject to} \quad & \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0 \\ & \text{margin}(b, \mathbf{w}) = \min_{n=1,\ldots,N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b) \end{aligned}$$

进行简化：

$$\begin{aligned} \max_{b,\mathbf{w}} \quad & \frac{1}{\|\mathbf{w}\|} \\ \text{subject to} \quad & \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0 \\ & \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \end{aligned}$$

我们的目标就是根据这个条件，计算$\frac{1}{||w||}$的最大值。

可以把目标$\frac{1}{||w||}$最大化转化为计算$\frac{1}{2}w^Tw$的最小化问题

necessary constraints: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ for all n

original constraint: $\min_{n=1,\ldots,N} \; y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$
want: optimal $(b, \mathbf{w})$ **here (inside)**

if optimal $(b, \mathbf{w})$ outside, e.g. $y_n(\mathbf{w}^T\mathbf{x}_n + b) > 1.126$ for all n
—can scale $(b, \mathbf{w})$ to “more optimal” $(\frac{b}{1.126}, \frac{\mathbf{w}}{1.126})$ **(contradiction!)**

final change: max $\Longrightarrow$ min, remove $\sqrt{}$, add $\frac{1}{2}$

$$\begin{aligned} \min_{b,\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n \end{aligned}$$

最终的条件就是$y_n(w^Tx_n + b) \geq 1$，而我们的目标就是最小化$\frac{1}{2}w^Tw$值。

3.Support Vector Machine

现在，条件和目标变成：

$$\begin{aligned} \min_{b,\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n \end{aligned}$$

Support Vector Machine(SVM)这个名字从何而来？为什么把这种分类面解法称为支持向量机呢？这是因为分类面仅仅由分类面的两边距离它最近的几个点决定的，其它点对分类面没有影响。决定分类面的几个点称之为支持向量（Support Vector），好比这些点“支撑”着分类面。而利用Support Vector得到最佳分类面的方法，称之为支持向量机（Support Vector Machine）。

这是一个典型的二次规划问题，即Quadratic Programming（QP）。因为SVM的目标是关于w的二次函数，条件是关于w和b的一次函数，所以，它的求解过程还是比较容易的，可以使用一些软件（例如Matlab）自带的二次规划的库函数来求解。下图给出SVM与标准二次规划问题的参数对应关系：

optimal $(b, \mathbf{w}) = ?$

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad \text{for } n = 1, 2, \ldots, N$$

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

$$\min_{\mathbf{u}} \quad \frac{1}{2}\mathbf{u}^T\mathrm{Q}\mathbf{u} + \mathbf{p}^T\mathbf{u}$$
$$\text{subject to} \quad \mathbf{a}_m^T\mathbf{u} \geq c_m, \quad \text{for } m = 1, 2, \ldots, M$$

$$\text{objective function:} \quad \mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}; \mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}; \mathbf{p} = \mathbf{0}_{d+1}$$
$$\text{constraints:} \quad \mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}; c_n = 1; M = N$$

那么，**线性SVM**算法可以总结为三步：
1.计算对应的二次规划参数Q，p，A，c
2.根据二次规划库函数，计算b，w
3.将b和w代入g_{SVM}，得到最佳分类面

Linear Hard-Margin SVM Algorithm

1. $\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}; \mathbf{p} = \mathbf{0}_{d+1}; \mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}; c_n = 1$
2. $\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \leftarrow \text{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$
3. return b & $\mathbf{w}$ as g_{SVM}

如果是非线性的，可以先用特征转换的方法，先做特征变换。将非线性的x域映射到线性的z域，再利用线性SVM算法进行求解。

4.Reasons behind Large-Margin Hyperplane

SVM的思想与正则化regularization思想很类似。

	minimize	constraint
regularization	E_{in}	$\mathbf{w}^T\mathbf{w} \leq C$
SVM	$\mathbf{w}^T\mathbf{w}$	$E_{\text{in}} = 0$ [and more]

如果Dichotomies越少，那么复杂度就越低，即有效的VC Dimension就越小，得到$E_{out} \approx E_{in}$，泛化能力强。

总结

本节课主要介绍了线性支持向量机（Linear Support Vector Machine）。我们先从视觉角度出发，希望得到一个比较“胖”的分类面，即满足所有的点距离分类面都尽可能远。
然后，我们通过一步步推导和简化，最终把这个问题转换为标准的二次规划（QP）问题。二次规划问题可以使用软件来进行求解，得到我们要求的w和b，确定分类面。
这种方法背后的原理其实就是减少了dichotomies的种类，减少了有效的VC Dimension数量，从而让机器学习的模型具有更好的泛化能力。