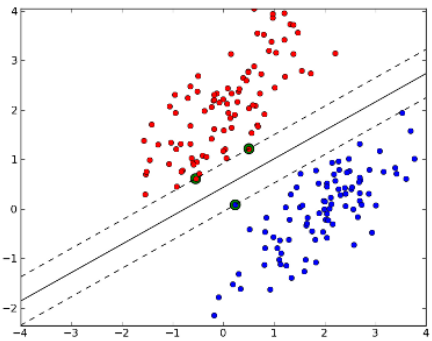
攀登传统机器学习的珠峰-SVM (上)

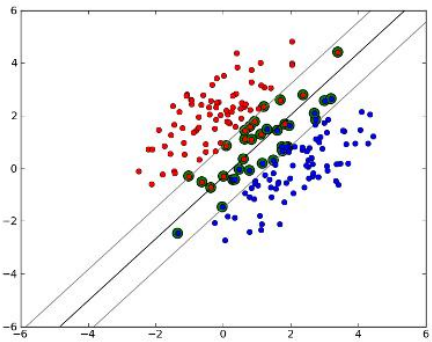
----------**author：August助教（网易云课堂，机器学习微专业）**----------

SVM的基本形式是一个有监督的决策线性（只输出 -1 和 1，没有输出概率的功能）二分类模型，它是间隔最大化的分类器。主要包括以下几种形式：

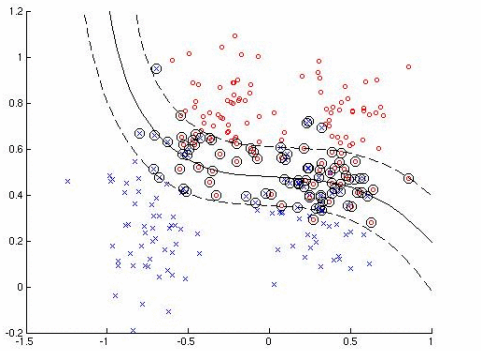
- 当训练数据线性可分时，支持向量机通过硬间隔最大化学习分类器，称为硬间隔支持向量机；



- 当训练数据近似线性可分时，支持向量机通过软间隔最大化学习分类器，称为软间隔支持向量机；



- 当训练数据线性不可分时，支持向量机通过核技巧和软间隔最大化学习分类器，称为非线性支持向量机；



此外，SVM 既支持二元分类也支持多元分类，既支持分类问题也支持回归问题。

SVM 诞生于上世纪九十年代，由于它良好的分类性能，自一诞生便席卷了机器学习领域，并牢牢压制了神经网络领域好多年，据说 LeNet5（一种CNN手写数字识别算法，属于神经网络）自1998年诞生，在后来的好长一段时间并未能火起来，最主要的原因就是 SVM这货，因为 SVM 也能达到类似的效果甚至超过LeNet5， 而且比LeNet5 计算量小。在不考虑集成学习的算法和特定场景情况下，在分类算法中SVM毫无疑问是性能最好的分类器。

哇！如此强大的算法，我们不懂可就亏大了！！！

1. 预备知识：感知机模型

1.1 假设函数

感知机的模型就是尝试找到一个超平面（线性决策边界），能够把所有的二元类别隔离开。

用数学的语言来说，对于一个二元分类问题，如果我们有 m 个样本， n 维特征，如下：

$(x_1^{(0)}, x_2^{(0)}, \dots x_n^{(0)}, y_0), (x_1^{(1)}, x_2^{(1)}, \dots x_n^{(1)}, y_1), \dots (x_1^{(m)}, x_2^{(m)}, \dots x_n^{(m)}, y_m)$

我们的目标是找到这样一个超平面，即：

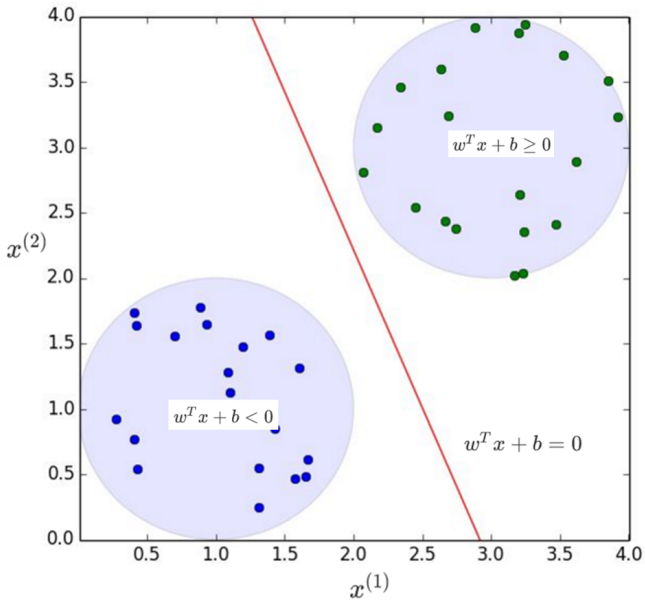
$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b = 0 \Leftrightarrow w^T x + b = 0$$

作为决策边界，将不同类别的数据分开。即：

$$y = sign(w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b = 0) \Leftrightarrow y = sign(w^T x + b)$$

其中 $sign(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$

翻译成汉语，找到一个超平面 $w^T x + b = 0$ 作为线性决策边界，在超平面的上方我们定义为 $y = 1$，在超平面的下方我们定义为 $y = -1$。模型的原理如下图所示:



Q1: 感知机可以处理非线性数据吗？和我们之前讲得什么算法是类似？

从以上感知机的模型原理可以发现，目标变量 y 是特征 x 的线性组合的符号函数，也就是说感知机是一个线性分类器，用于处理线性可分的数据。是不是和逻辑回归差不多？

1.2 损失函数

与逻辑回归模型不同，我们正负样本的 label 定义为 1 和 -1 (逻辑回归正负样本的 label 是1 和 0) 为什么这样定义呢？

因为正确分类的样本满足 $y(w^T x + b) > 0$，而错误分类的样本满足 $y(w^T x + b) < 0$，这样可以更加容易找到正确分类的样本和误分类样本。感知机代价函数的优化目标，就是期望所有误分类的样本，到超平面的距离之和最小。怎样定义这个距离呢？

我们知道，我们的决策边界的方程式是 $w^T x + b = 0$，

我们来研究一下这个式子 $w^T x^{(i)} + b$ ，假如我们的样本点在决策边界上，必然满足 $w^T x^{(i)} + b = 0$， 随着该样本点偏离决策边界 $|w^T x^{(i)} + b|$ 也会变大（这是我们初中学的）。且如果样本点在决策边界下方 $w^T x^{(i)} + b < 0$，如果样本点在决策边界上方 $w^T x^{(i)} + b > 0$。我们可以用 $|w^T x^{(i)} + b|$ 来表示样本点距离决策边界的距离。

Q2: 绝对值不可导，我们不想用它，那有没有更好的方法呢？

下面我们再来研究这个式子 $-y^{(i)}(w^T x^{(i)} + b)$ ，由上面的解释我们知道误分类的样本满足 $y^{(i)}(w^T x^{(i)} + b) < 0$，那么 $-y^{(i)}(w^T x^{(i)} + b) = |w^T x^{(i)} + b| > 0$ 表示样本点距离决策边界的距离。假设所有误分类的点的集合为 M_{erro}，对所有误分类的样本距离进行求和得到感知机模型的代价函数为：

$$J(w, b) = -\sum_{x_i \in M_{erro}} y^{(i)}(w^T x^{(i)} + b)$$

1.3 优化算法

Q3: 可以用基于所有样本的批量梯度下降法（BGD）来优化感知机可行吗？

基于所有样本的批量梯度下降法（BGD）是行不通的，原因在于我们的损失函数里面只有误分类的 M 集合里面的样本才能参与损失函数的优化。感知机模型选择的是采用随机梯度下降，这意味着我们每次仅仅需要使用一个误分类的点来更新梯度。即：

$$w := w + \alpha y^{(i)} x^{(i)}$$
$$b := b + \alpha \sum_{x_i \in M_{erro}} y^{(i)}$$

其中 α 为步长，$y^{(i)}$ 为样本输出 1 或者 -1，$x^{(i)}$为 $n \times 1$的向量。

1.4 模型总结

算法的输入为 m 个样本，每个样本对应于 n 维特征和一个二元类别输出 1 或者 -1，如下：

$(x_1^{(0)}, x_2^{(0)}, \ldots x_n^{(0)}, y_0), (x_1^{(1)}, x_2^{(1)}, \ldots x_n^{(1)}, y_1), \ldots (x_1^{(m)}, x_2^{(m)}, \ldots x_n^{(m)}, y_m)$

输出为线性决策边界的模型系数 w, b

算法的执行步骤如下：

1) 初始化模型系数 w , b 和步长 α 。

2) 在训练集里面选择一个误分类的点 $(x^{(i)}, y^{(i)})$，这个点应该满足：$y^{(i)}(w^T x^{(i)} + b) < 0$ 。

3) 对 θ 向量进行一次随机梯度下降的迭代：

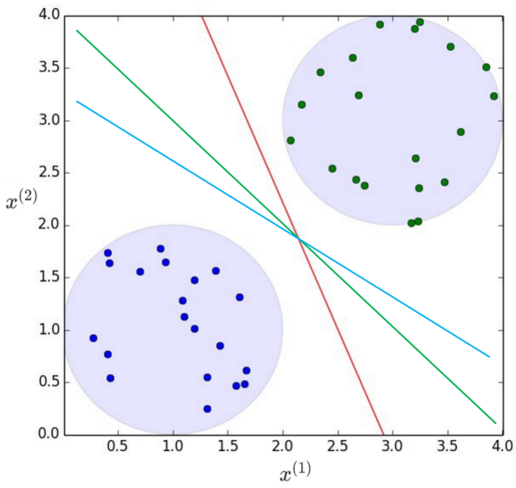
$w := w + \alpha y^{(i)} x^{(i)}$ $b := b + \alpha \sum_{x_i \in M} y^{(i)}$。

4) 检查训练集里是否还有误分类的点，如果没有，算法结束。如果有，继续第 2 步。

1.5 感知机的遗留问题

Q4：从感知机的分类原理中，可以看出满足条件的超平面并不止一个，也就是说感知机模型可以有多个解。这么多的可以分类的超平面，哪个是最好的呢？

当然是泛化能力最好的模型性能最好了。对于训练集来说，三个超平面都可以很好地将两类数据划分开，但是并不一定能保证很好的将测试的数据集分开。因此我们需要找到最优的超平面。



2. 线性支持向量机

2.1 线性支持向量机的三要素

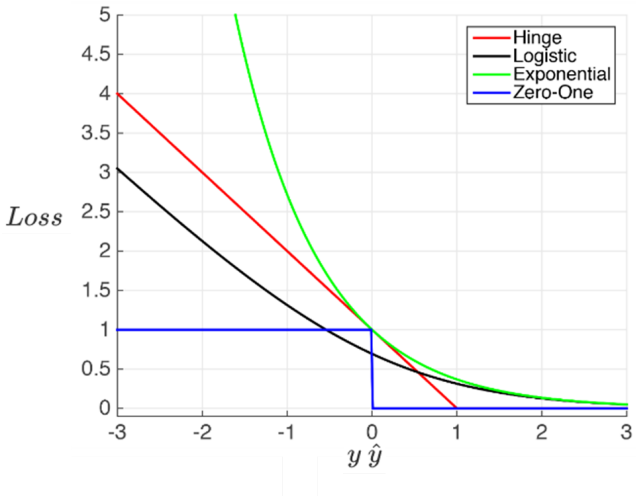
假设函数：

$$\hat{y} = sign(w^T x + b)$$

其中 $sign(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$

代价函数：

$$J(\theta) = max\{0, 1 - y\hat{y}\}$$



优化算法：

主要是凸优化 (KKT条件，拉格朗日对偶)，SMO算法。下面会主要介绍这部分。

总结对比一下：

	逻辑回归	**感知机**	**支持向量机**
模型输出	概率模型 (概率)	决策模型 (样本类别)	决策模型 (样本类别)
损失函数	交叉熵	误分类点的函数间隔	Hinge 损失函数
优化算法	梯度下降	随机梯度下降	凸优化，SMO算法

2.2 线性支持向量机的公式推导

在感知机和逻辑回归模型中，我们希望所有的点都离超平面远。这样抓全局真的有必要吗？效果一定真的好吗？

但是实际上我们只要保证那些离超平面很近的点（这些点很容易被误分类）尽可能远离超平面即可。什么意思呢？小明一家五口（爷爷：75，奶奶：73，爸爸：42，妈妈：40，小明：12），我们想要确定小明家有几个成人，我们只要统计小明的年龄就可以了（前提是小明爷爷奶奶爸爸妈妈都是正常结婚），很显然其他四个人都是成年人，没必要再统计其他人了。对吧？

既然SVM 的定义是最大间隔分类器，那么我们便会产生一系列疑问：

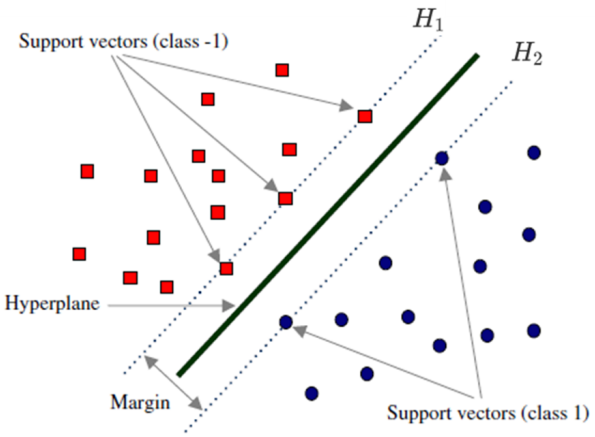
- 这个间隔的定语是什么？是什么东东的间隔？
- 怎样定义间隔？
- 怎样定义支持向量的最大间隔？
- 怎样根据最大间隔求解模型参数？

下面我们将一一讲解这些概念，讲完之后，相信大家对线性支持向量机会有更深刻的认识。

2.2.1 支持向量

首先回答第一个问题，这里的间隔实际上是支持向量到超平面的间隔。那么什么是支持向量呢？

距离超平面最近的样本点，我们定义为支持向量。如下图所示，黑色实线为超平面 (hyperplane) ，在虚线 H_1 和虚线 H_2 上的点即为支持向量 。



Q5: 下面有三个小问题：

- H_1 和 H_2 关于超平面对称吗？

  H_1 和 H_2 关于超平面对称

- 超平面两侧一定都有支持向量吗？

  对于初始化的超平面而言不一定。但是最终一定会求出一个合适的决策边界，既能保证 H_1 和 H_2 关于超平面对称，又能保证超平面两侧都有支持向量。

- 为什么点也称为向量？

  所有的点都可以看做以原点为起点，以该点的坐标为终点的向量。点就是向量，向量就是点。

H_1 和 H_2 之间的距离称为 margin (值为 $\frac{2}{||w||_2}$ ，下面会介绍原因)

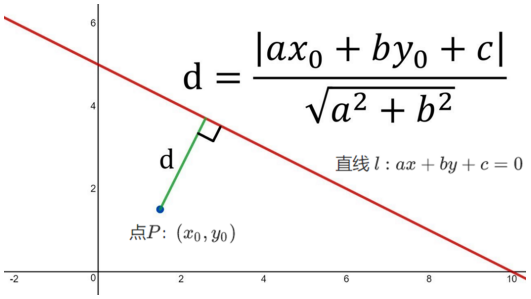
2.2.2 函数间隔 (functional margin) 与几何间隔 (geometric margin)

首先我们定义几何间隔 γ ：

$$\gamma = \frac{y(w^T x + b)}{||w||_2}$$

其中 $||w||_2$ 为 L2 范数。

好奇怪的公式，为什么称为间隔呢？不知道大家还记不记得点到直线的距离：



Q6：是不是有印象了？再对比一下两个，看看是不是一个模子刻出来的？他们有什么不同点？

细心的同学肯定会发现，上面的式子分子是 $y(w^Tx+b)$，而点到直线的距离分子是 $|w^Tx+b|$？ 我们知道在SVM中，目标变量 y 要么为 1，要么为 -1， y 仅仅是调节符号的作用。对于正确分类的样本点 $y(w^Tx+b)>0$，对于误分类的点$y(w^Tx+b)<0$。

总结一下，$\gamma=\frac{y(w^Tx+b)}{||w||_2}$ 就是点到直线的距离在高维空间中的形式，也称为几何间隔。

再看看函数间隔 γ'：

$$\gamma^{'}=y(w^Tx+b)$$

这不就是我们的感知机损失函数吗？这两个间隔有什么关系呢？

$$\gamma=\frac{\gamma^{'}}{||w||_2}$$

几何间隔可以理解为对函数间隔做了归一化的处理。函数间隔可以理解为几何间隔在分母 $||w||_2=1$ 条件下的特殊情况。我们考虑一下如果分子 $\gamma^{'}=1$ 时，几何间隔就变成了 $\gamma=\frac{1}{||w||_2}$，这个有没有意义呢？先埋个伏笔。

下面讲一下参数 w 和 b 放大或缩小 C 倍对超平面，函数间隔和几何间隔的影响。

由超平面的解析式： $w^Tx+b=0$

方程两边同时乘以常数 C，即 $C\,w^Tx+C\,b=0$

显然以上两个方程的解是一致的，超平面没有发生变化。换句话说，参数 w 和 b 同比例缩放对超平面并没有影响。然而参数 w 和 b 放大或缩小 C 倍，函数间隔：

$\gamma^{'}=y(Cw^Tx+Cb)=Cy(w^Tx+b)=C\gamma^{'}$

这意味着函数间隔 $\gamma^{'}$ 也会放大和缩小相同的倍数。

而几何间隔：

$\gamma=\frac{C\gamma^{'}}{C||w||_2}=\gamma$

这意味着几何间隔不变。

Q7：既然几何间隔也是一种距离定义，而且还是归一化的距离，不受参数 w 和 b 同比例缩放的影响，是不是更适合作为感知机的损失函数吗？为什么不用这个公式作为损失函数呢？

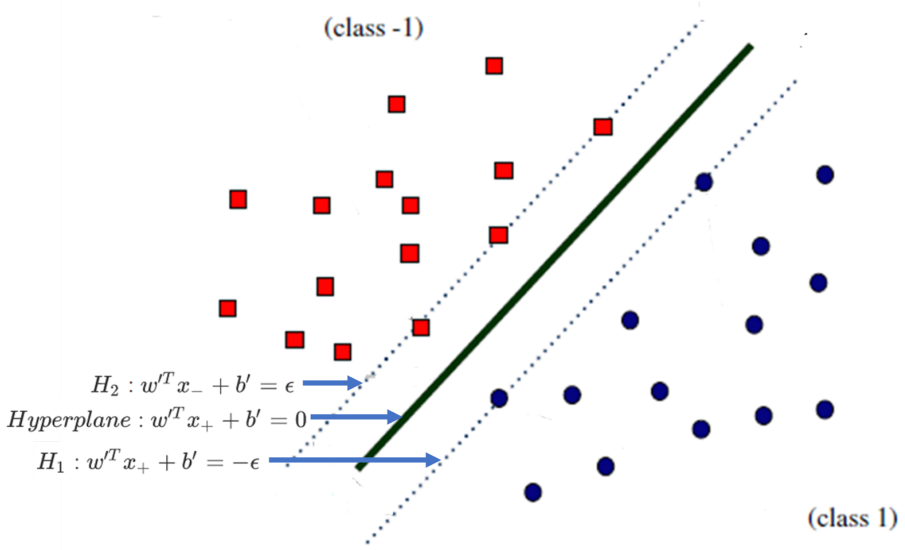
分子和分母都含有未知数，如果用梯度下降法去求解，可以想象求导的难度系数。相对来说，函数间隔求导要简单得多。而且虽然函数间隔没有进行归一化，但是对于同一个模型而言，我们仅仅研究的是样本点距离超平面的相对距离，并不考虑绝对距离。因此用函数间隔更加合理。

2.2.3. 怎样定义最大间隔

SVM 的模型是让所有点到超平面的距离大于一定的距离，也就是所有的分类点要在各自类别的支持向量两边。我们只需要定义支持向量到超平面的距离最大化即可。先上最大间隔的定义：

$$max\ \frac{2}{||w||_2}\ \ s.t\ \ y_i(w^Tx_i+b)\geq 1(i=1,2,...m)$$

思考一下上面我们的讲的，当分子 $\gamma' = 1$ 时，几何间隔就变成了 $\gamma = \frac{1}{||w||_2}$ ，是不是很相近？这个公式怎么来的呢？为什么分子是 2 而不是 1 呢？下面我们一一说明。



如上图所示，我们定义超平面, H_1 和 H_2 的数学表达式为：

$Hyperplane : w'^T x_+ + b' = 0$

$H_1 : w'^T x_+ + b' = -\epsilon$

$H_2 : w'^T x_- + b' = \epsilon$

方程两边同时除以 ϵ，令 $w = \frac{w'}{\epsilon}$，$b = \frac{b'}{\epsilon}$，简化方程组我们得到：

超平面： $w^T x_+ + b = 0$

$H_1 : w^T x_+ + b = -1$

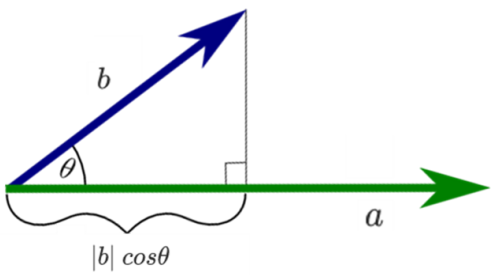
$H_2 : w^T x_- + b = 1$

$H_1 - H_2$ 我们得到

$(-w)^T (x_+ - x_-) = 2$

w 表示超平面的法向量，$(x_+ - x_-)$ 表示沿 x 轴方向的向量，那两个向量的内积是？我们先来看看向量内积的计算过程：

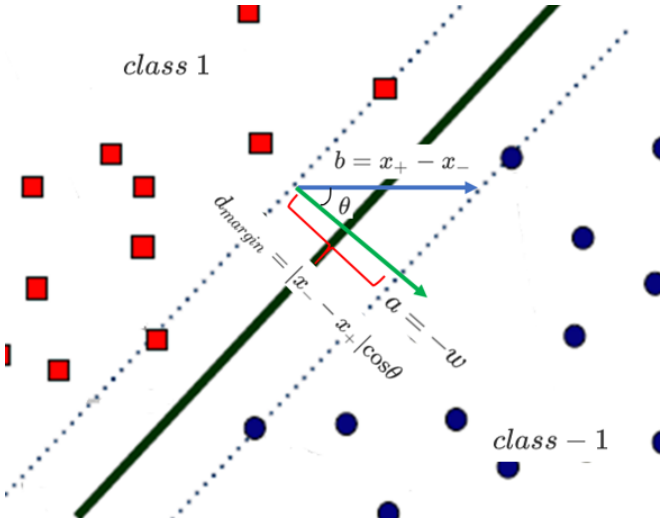
$a \bullet b = a^T b = |a|\ |b|\ cos\theta = |a|(|b|\ cos\theta)$



从上面的公式可以推出：

$|w|\ |x_+ - x_-| \cos\theta = 2$

我们把上面的三角形移植到SVM分类问题中：



则 $d_{margin} = |x_+ - x_-|\cos\theta$， 继续化简：

$|w|d_{margin} = 2$

$d_{margin} = \frac{2}{||w||_2}$

注意：$|w| = ||w||_2 = \sqrt{w_1^2 + w_2^2 + \cdots}$

这样我们的优化函数定义为：

$$max\ \frac{2}{||w||_2}\ \ s.t\ \ y_i(w^T x_i + b) \geq 1(i = 1, 2, \ldots m)$$

也就是说，我们要在约束条件 $y_i(w^T x_i + b) \geq 1(i = 1, 2, \ldots m)$ 下，最大化 $\frac{2}{||w||_2}$ 。翻译成汉语，优化条件：保证其他样本点在 H_1 和 H_2 （支持向量）以外，通俗一点就是找到下限，优化目标：最大化支持向量和超平面之间的距离，通俗一点就是最大化下限和决策边界的距离。大家仔细想想有没有学过类似的思想？EM 算法里面提到过，E步是……，M步是……，想起来了没？是不是有异曲同工之妙？（有一种天下算法一大抄的感觉）

SVM 的代价函数等价于：

$$min\ \ \frac{1}{2}||w||_2^2 \qquad s.t\ \ y_i(w^T x_i + b) \geq 1(i = 1, 2, \ldots m)$$

总结一下，我们再来看看回头看看几何间隔：

$$\gamma = \frac{y(w^T x + b)}{||w||_2}$$

感知机的代价函数是分母为 1 的几何间隔，SVM 的优化函数是分子为 2 几何间隔。

Q8：本小节遗留问题，为什么分子取2，而不是1呢？

大家想想逻辑回归的代价函数：

$J(\theta) = -\sum y_i \log \hat{y}_i - (1 - y_i)\log(1 - \hat{y}_i) + \frac{1}{2}\alpha||w||_2^2$

大家回忆一下为什么逻辑回归的正则化系数是 $\frac{1}{2}\alpha$ 而不是 α 。

2.2.4 求解最大间隔

线性可分SVM算法的优化过程分为以下五步：

- 转化为拉格朗日函数
- 转化为对偶问题
- 简化对偶问题
- SMO 算法求解 α
- 根据 α 求解出 w 和 b

转化为拉格朗日函数

根据凸优化理论，代价函数满足KKT条件，我们可以通过拉格朗日函数将我们的优化目标转化为无约束的优化函数：

$$L(w,b,\alpha)=\frac{1}{2}||w||_2^2-\sum_{i=1}^{m}\alpha_i[y_i(w^Tx_i+b)-1] \qquad s.t\ \alpha_i\geq 0$$

Q10：这个家伙 $\frac{1}{2}||w||_2^2$ 是不是有点面熟？想想我们在哪儿见过它？

想想我们的逻辑回归代价函数：

$J(\theta)=-\sum y_i\log\hat{y}_i-(1-y_i)\log(1-\hat{y}_i)+\frac{1}{2}\alpha||w||_2^2$

不就是 $\frac{1}{2}\alpha||w||_2^2$ 吗？妙哉，妙哉，一个公式，两种解释，可以理解为正则化，也可以理解为距离的倒数！！！（来自Ivan老师）

由于引入了拉格朗日函数，我们的优化目标变成：

$$\underbrace{min}_{w,b}\ \underbrace{max}_{\alpha_i\geq 0}L(w,b,\alpha)$$

转化为对偶问题

这个拉格朗日函数满足KKT条件，我们可以通过拉格朗日对偶将该问题转化为等价的对偶问题来求解。我们可以先求优化函数对于 w 和 b 的极小值。接着再求拉格朗日乘子 α 的极大值，即：

$$\underbrace{max}_{\alpha_i\geq 0}\ \underbrace{min}_{w,b}L(w,b,\alpha)$$

简化对偶问题

首先我们来求 w 和 b 的极小值，即 $\underbrace{min}_{w,b}\ L(w,b,\alpha)$ 。这个极值我们可以通过对 w 和 b 分别求偏导数得到：

$$\frac{\partial L}{\partial w}=0\ \Rightarrow w=\sum_{i=1}^{m}\alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b}=0\ \Rightarrow \sum_{i=1}^{m}\alpha_i y_i=0$$

我们已经求得了 w 和 α 的关系，带入优化函数 $L(w,b,\alpha)$ 消去 w 。我们定义:

$$\psi(\alpha)=\underbrace{min}_{w,b}\ L(w,b,\alpha)$$

现在我们来看将 w 替换为 α 的表达式以后的优化函数 $\psi(\alpha)$ 的表达式：

好长的等式推导，好恐怖，好可怕！！！在开始推导之前先让我们预习一下相关的知识点：

- 首先我们需要知道公式里面的 w 和 x 均为向量，α 和 b 为实数；
- $||w||_2^2 = w^T w$
- 根据上面推导： $w = \sum_{i=1}^{m} \alpha_i y_i x_i$
- w^T和样本 i 没关系： $\sum_{i=1}^{m} \alpha_i y_i w^T x_i = w^T \sum_{i=1}^{m} \alpha_i y_i x_i$
- 提取常数项： $\sum_{i=1}^{m} \alpha_i y_i b = b \sum_{i=1}^{m} \alpha_i y_i$
- 根据上面推导： $\sum_{i=1}^{m} \alpha_i y_i = 0$
- $-\frac{1}{2}(\sum_{i=1}^{m} \alpha_i y_i x_i)^T (\sum_{i=1}^{m} \alpha_i y_i x_i) = -\frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i x_i^T \sum_{i=1}^{m} \alpha_i y_i x_i$
- $$\begin{aligned} \sum_{i=1}^{m} a_i * \sum_{i=1}^{m} b_i &= (a_1 + a_2 + a_3 + \ldots)(b_1 + b_2 + b_3 + \ldots) \\ &= a_1 b_1 + a_1 b_2 + \cdots + a_2 b_1 + a_2 b_2 + \cdots \\ &= \sum_{i=1,j=1}^{m} a_i b_j \end{aligned}$$

  好啦，剩下的时间让我们来愉快的玩耍吧：

$$\begin{aligned} \psi(\alpha) &= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{m} \alpha_i [y_i (w^T x_i + b) - 1] \\ &= \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i y_i w^T x_i - \sum_{i=1}^{m} \alpha_i y_i b + \sum_{i=1}^{m} \alpha_i \\ &= \frac{1}{2} w^T \sum_{i=1}^{m} \alpha_i y_i x_i - \sum_{i=1}^{m} \alpha_i y_i w^T x_i - \sum_{i=1}^{m} \alpha_i y_i b + \sum_{i=1}^{m} \alpha_i \\ &= \frac{1}{2} w^T \sum_{i=1}^{m} \alpha_i y_i x_i - w^T \sum_{i=1}^{m} \alpha_i y_i x_i - b \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} w^T \sum_{i=1}^{m} \alpha_i y_i x_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} (\sum_{i=1}^{m} \alpha_i y_i x_i)^T (\sum_{i=1}^{m} \alpha_i y_i x_i) + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i x_i^T \sum_{i=1}^{m} \alpha_i y_i x_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i x_i^T \sum_{i=1}^{m} \alpha_i y_i x_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i y_i x_i^T \alpha_j y_j x_j + \sum_{i=1}^{m} \alpha_i \end{aligned}$$

从上面可以看出，通过对 w, b 极小化以后，我们的优化函数 $\psi(\alpha)$ 仅仅只有 α 向量做参数。只要我们能够极大化 $\psi(\alpha)$ ，就可以求出此时对应的 α ，进而求出 w, b 。

对偶问题的简化结果如下：

$$\underbrace{max}_{\alpha} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j (x_i^T x_j) + \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0 \; i = 1, 2, \ldots m$$

可以去掉负号，即为等价的极小化问题如下：

$$\underbrace{min}_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j (x_i^T x_j) - \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0 \; i = 1, 2, \ldots m$$

SMO算法求解 α

只要我们可以求出上式极小化时对应的 α 向量就可以求出 w 和 b 了 (需要用到SMO算法)。在这里，我们假设通过SMO算法，我们得到了对应的 α 的值 α^*。

根据 α 求解出 w 和 b

那么我们根据 $w = \sum_{i=1}^{m}\alpha_i y_i x_i$ ， 可以求出对应的 w 的值

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

再求出 b 我们就大功告成啦。注意到，对于支持向量 (x_s, y_s)，都有

$y_s(w^{*T}x_s + b) = 1$

将 $w^* = \sum_{i=1}^{m}\alpha_i^* y_i x_i$ 带入上式可以得到：

$y_s(\sum_{i=1}^{m}\alpha_i^* y_i x_i^T x_s + b) = 1$

求得：

$b_s^* = y_s - \sum_{i=1}^{m}\alpha_i y_i x_i^T x_s$

假设我们有 S 个支持向量，则对应我们求出 S 个 b_s^*，然后将其平均值作为最后的结果。

Q10：怎么得到支持向量呢？

根据KKT条件中的对偶互补条件 $\alpha_i^*(y_i(w^T x_i + b) - 1) = 0$，如果 $\alpha_i > 0$ 则有 $y_i(w^T x_i + b) = 1$ 即点在支持向量上，否则如果 $\alpha_i = 0$ 则有 $y_i(w^T x_i + b) \geq 1$，即样本在支持向量上或者已经被正确分类。

3. 线性可分SVM的算法过程

输入是 m 个线性可分的样本 $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$,

其中 x 为 n 维特征向量。y 为二元分类结果 1或 -1。

输出是分离超平面的参数 w^* 和 b^* 和分类决策函数。

算法过程如下：

1) 构造代价函数：

$$\underbrace{min}_{\alpha}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j(x_i^T x_j)-\sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0\ i=1,2,\dots m$$

2) 用 SMO 算法求出 α 向量的值 α^* 。

3) 计算 $w^*=\sum_{i=1}^{m}\alpha_i^* y_i x_i$。

4) 找出所有的 S 个支持向量(x_s,y_s)，计算出每个支持向量 (x_s,y_s) 对应的 b_s^*，取平均得到 b

这样最终的分类超平面为：$w^{*T}x+b^*=0$

最终的分类决策函数为：$f(x)=sign(w^{*T}x+b^*)$

参考文献：

支持向量机原理(一) 线性支持向量机 http://www.cnblogs.com/pinard/p/6097604.html