支持向量机

编写者：杨旭瑜 陈泽 屠文翔

一、支持向量机简介:

支持向量机（Support Vector Machines简称SVM）是Vapnik根据统计学理论提出的一种机器学习算法。这种基于结构风险最小化原则的机器学习算法，能够在有限训练样本的基础上,求解实际问题时尽量提高它的推广能力并能够得到准确度较高的解。同时,由于支持向量机算法是一个凸二次规划问题，故其最终求得的解是全局最优解，是一种全局机器学习算法。相比较其它机器学习算法，它不但有强大的统计学理论作为基础，而且在训练过程中所需调整的参数比较少，有较高的推广性。在模式识别领域，具有较好推广能力分类器的设计一直备受关注。传统模式识别或者人工神经网络基本上都是以大样本统计理论为指导，但是很多实际问题往往面对的是小样本，基于小样本并具有优良推广能力统计学模型的研究是模式识别领域中一个新的研究热点。神经网络等新兴机器学习方法的研究和应用，目前已经遇到一些难题，例如网络结构的确定问题、欠拟合与过拟合问题以及局部极值问题等。相比较神经网络等新兴机器学习方法而言，Vapnik等人研究的有限样本条件下的支持向量机理论逐渐成熟和完善，并以完备的统计学理论为指导形成了一个基础机器学习算法，它在解决小样本、非线性及高维模式识别问题中表现出极大的优势和良好的研究与发展前景。尤为重要的是在解决非线性小样本问题时，支持向量机通过引入核函数，将原始输入空间映射到高维空间，将原始输入空间的非线性问题转化为映射后高维特征空间的线性问题来求解，从理论上满足了统计学习的一致性条件和经验风险最小化原则，并基于统计学习方法泛化边界的基础上建立小样本条件下归纳推理统计学习算法。

总的来说：支持向量机是一种通用的机器学习方法，属于有监督的那种。（因为在输入特征值的时候需要给出改特征值的正负性，即要明确告诉机器，该样本是正样本还是负样本。)相比于神经网络，支持向量机相对没那么容易过度拟合，其中有好些机制可以尽量避免过度拟合。另外，支持向量机是用来“二分”的，即，一个样本要么是正的要么是负的。不过，通过一些处理办法，可以用支持向量机来区分“多类”的问题。一下详细阐述了支持向量机的理论基础。

二、支持向量机的理论基础

支持向量机处理的数据有两种情况，分别是线性可分的数据集和非线性可分的数据集。对于线性可分的数据集处理起来就比较简单了，可以保证绝对不会出错。然而如果要处理非线性可分的数据集就要有一定的容错了，即加入一些参数，允许少量的错误产生，牺牲准确性来用线性办法区分非线性可分的数据。（支持向量机使用的是线性的区分办法，即变量的次数最多是一次。）

1. 线性可分情况下的SVM

如果用一个线性函数（如2维空间中的直线，3维空间中的平面以及更高维度空间中的超平面)可以将两类样本完全分开，就称这些样本是线性可分(linearly separable）的。反之，如果找不到一个线性函数能够将两类样本分开，则成这些样本是非线性可分的。

已知一个线性可分的数据集$\{(\overrightarrow{x_1},y_1),(\overrightarrow{x_2},y_2),\dots,(\overrightarrow{x_N},y_N)\}$，样本特征向量，$\vec{x}\in R^D$,即$\vec{\mathrm{x}}$是D维实数空间中的向量；类标签$y\in\{-1,+1\}$,即只有两类样本，此时通常称类标签为确定最大分类间隔的分割超平面，设最优超平面方程为$\vec{w}^T\vec{x}+b=0$（一个线性方程）,根据点到平面的距离公式：样本$\vec{\mathrm{x}}$与最佳超平面$(\vec{w},b)$之间的距离为$\frac{|\vec{w}^T\vec{x}+b|}{\|\vec{w}\|}$，注意通过等比例地缩放权矢量$\vec{\mathrm{w}}$和偏差项b最佳超平面存在着许多解，我们对超平面进行规范化，选择使得距超平面最近的样本$\vec{\mathrm{x}}_{\mathrm{k}}$满足$|\vec{w}^T\vec{x}+b|=1$的$\vec{\mathrm{w}}$和b，即得到规范化超平面。（个人感觉规范化是为了计算简便，反正最优解有无数种，那么给系数加上不影响它们形状的约束，更是方便了下面的求解。其实就像一个平面可以写出无数个方程，但是如果要求它们的系数相互之间没有除1外的公约数，那么一个平面就只有一个方程了，不过这并不影响平面的表示。）

此时从最近样本到边缘的距离为：

$$\frac{|\vec{w}^T\vec{x}+b|}{\|\vec{w}\|}=\frac{1}{\|\vec{w}\|} \tag{1}$$

且分类间隔（余地）变为：$m=\frac{2}{\|\vec{w}\|}$

(2)

至此，问题逐渐明朗化，我们的目标是寻找使得式(2)最大化的法向量$\vec{\mathrm{w}}$，之后我们将$\vec{\mathrm{w}}$代入关系式$|\vec{\mathrm{w}}^{\mathrm{T}}\vec{\mathrm{x}}+\mathrm{b}|=1$，即可得到b。

最大化式（2）等价于最小化：

$$J(\vec{w})=\frac{1}{2}\|\vec{w}\|^2 \tag{3}$$

除此之外，还有以下的约束条件：

$$y(\vec{w}^T\vec{x}_i+b)\geq 1,\forall i\in\{1,2,\dots,N\} \tag{4}$$

通过对式子（4）中的每一个约束条件乘上一个拉格朗日乘数α_i，然后代入式子（3）中，可以将此条件极值问题转化为下面的不受约束的优化问题，即关于$\vec{\mathrm{w}}$，b和$\alpha_i(i=1,2,\dots,N)$来最小化L。

$$L(\vec{w},b,\alpha)=\frac{1}{2}\|\vec{w}\|^2-\sum_{i=1}^{N}\alpha_i[y_i(\vec{w}^T\vec{x}_i+b)-1],\ \alpha_i\geq 0 \tag{5}$$

经过一系列计算，得到如下的最优分类函数：

$$h(\vec{x})=sgn\big((\vec{w}^*\vec{x})+b^*\big)=sgn(\sum_{i=1}^{n}\alpha_i{}^*y_i(\vec{x}_i\cdot\vec{x})+b^*) \tag{6}$$

这里向量$\vec{x}$是待分类的测试样本，向量$\vec{x}_i(i=1,2,\dots,N)$ 是全部N个训练样本。注意在式（6）中测试样本$\vec{x}$与训练样本$\vec{x}_i$也是以点积的形式出现。

2. 非线性可分情况下的C-SVM

(1).约束条件

为处理样本非线性可分的情况，我们放宽约束，引入松弛变量$\varepsilon_i>0$（加入一个容错因子，本来$y(\vec{w}\cdot\vec{x}_i+b)\geq 1$应该要成立才对，可惜如果数据线性不可分，那么上面的式子不能成立，所以加入一个变量来缓和它们的误差。）此时约束条件变为：

$$y(\vec{w}\cdot\vec{x}_i+b)\geq 1-\varepsilon_i,\ \ \varepsilon_i\geq 0(i=1,2,\dots,N) \tag{7}$$

即：

$$\begin{cases} y(\vec{w}\cdot\vec{x}_i+b)\geq+1-\varepsilon_i，\ y_i=+1(\vec{x}_i\text{为正例}) \\ y(\vec{w}\cdot\vec{x}_i+b)\leq-1+\varepsilon_i，\ y_i=+1(\vec{x}_i\text{为反例}) \end{cases}$$

具体可以分为以下三种情况来考虑：

①当$\varepsilon_i=0$,约束条件退化为线性可分的情况——$y_i(\vec{w}^T\vec{x}_i+b)\geq1$

②当$0<\varepsilon_i<1$, $y_i(\vec{w}^T\vec{x}_i+b)$是一个0,1之间的数，小于1意味着我们对约束条件放宽到允许样本落在分类间隔之内，大于0则说明样本仍可被分割超平面正确分类。

③最终当$\varepsilon_i>1$, $y_i(\vec{w}^T\vec{x}_i+b)<0$，此时约束条件已放宽到可以允许有分类错误的样本。

(2). 目标函数

利用一个附加错误代价系数C后，目标函数变为：

$$f(\vec{w},b,\varepsilon)=\frac{1}{2}\|\vec{w}^2\|+C\sum_{i=1}^{N}\varepsilon_i \tag{8}$$

我们的目标是在式（7）的约束下，最小化目标函数（式（8））。最小化目标函数的第一项也就等同于最大化分类间隔，这在介绍线性可分情况时已经阐述过了，而目标函数的第2项是分类造成的错误代价，只有对应于$\varepsilon_i>0$的那些“错误”样本才会产生代价（这里所说的“错误”并不仅仅指被错误分类的样本，也包括那些空白间隔之内的样本）。事实上最小化此目标函数体现了最大分类间隔（式（8）中的第1项）与最小化训练错误（式（8）中的第2项）之间的权衡。

直观上我们自然希望“错误”样本越少越好，然而不要忘记这里的错误是训练错误。如果我们一味追求最小化训练错误，则代价可能是导致得到一个小余地的超平面，这无疑会影响分类器的推广能力，在对测试样本分类时就很难得到满意的结果，这也属于一种过渡拟合。通过调整代价系C的值可以实现两者之间的权衡，找到一个最佳的C使得分类超平面兼顾训练错误和推广能力。

如果在训练过程中选择一个适当小一些的C值，此时最小化式（8）将兼顾训练错误与分类间隔。由此可知选择合适的错误代价系数C的重要性。正因为在这种处理非线性可分问题的方法中引入了错误代价系数C，这种支持向量机被称为C-SVM。

（3）优化求解

类似于线性可分情况下的推导，最终得到下面的对偶问题。

在如下的约束条件下：

$$\sum_{i=1}^{N}\alpha_i y_i=0，\ 0\leq\alpha_i\leq C，\ i=1,2,\dots,N \tag{9}$$

最大化$L(\alpha)$:

$$L(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\vec{x}_i\cdot\vec{x}_j) \tag{10}$$

同样在利用二次规划技术解得最优的的值之后，可以计算出和的值，最终的决策函数与式（6）相同。

3. 需要核函数映射情况下的SVM

线性分类器的分类性能毕竟有限，而对于非线性问题一味放宽约束条件只能导致大量样本的错分，这时可以通过非线性变换将其转化为某个高维空间中的

线性问题，在变换后的空间求得最佳分类超平面（根据自己的理解，这其实是将每一特征点按照相同的规则映射到高维空间，维数越高取得理想的线性分割超平面会比较容易，因为变量增加了，方程的可能数量大大增多了，即可以选择的多了很多。）。

（1） 非线性映射

实际上样本可能映射到高维空间R^D后仍非线性可分，这时只需在R^D中采用上面所介绍的非线性可分情况下的方法训练SVM。还有一点要说明的是，在分类时我们永远不需要将R^D中的分割超平面再映射回R^D当中，而是应让分类样本$\vec{\mathrm{x}}$也经非线性变换ψ映射到空间R^D中，然后将$\psi(\vec{\mathrm{x}})$送入R^D中的SVM分类器即可。

（2） 优化求解

经过一系列推导，最终得到了下面的对偶问题。

在如下的约束条件下：

$$\sum_{i=1}^{N} a_i y = 0 \quad 0 \le a_i \le C \ \mathrm{i} = 1,2,\ldots\ldots,\mathrm{N}$$

最大化目标函数：

$$\mathrm{L(a)} = \sum_{i=1}^{N} a_i \ - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j \left(\psi(\vec{\mathrm{x}}_i) * \psi(\vec{\mathrm{x}}_j)\right)$$

此时，因为有

$$\vec{w} = \sum_{i=1}^{N} a_i \, y_i \psi(\vec{\mathrm{x}}_i)$$

最后通过简化得到式子：

$$\mathrm{h(x)} = sgn\left(\sum_{i \in SV} a_i^* y_i (\psi(\vec{\mathrm{x}}_i) * \psi(\vec{\mathrm{x}}_j)) \ + \ b^*\right)$$

其中SV表示支持向量的集合。

（3） 核函数

计算所有样本的非线性映射并在高维空间中计算其点积常常是困难的。幸运的是，在一般情况下我们都不必如此，甚至不需要去关心映射ψ的具体形式。注意到上面的对偶问题中，都函数都只涉及样本特征向量的点积运算$\psi(\vec{\mathrm{x}}_i) * \psi(\vec{\mathrm{x}}_j)$。因此，在高维空间上实际只需要进行点积运算，而再高维数的向量的点积结果也是一个常数，那么能否抛开映射$\psi(\vec{\mathrm{x}}_i)$和$\psi(\vec{\mathrm{x}}_j)$的具体形式而直接根据$\vec{\mathrm{x}}_i$和$\vec{\mathrm{x}}_j$在原特征空间中得到$\psi(\vec{\mathrm{x}}_i) * \psi(\vec{\mathrm{x}}_j)$的常数结果呢？答案是肯定的，应为这种点积运算是可以用原特征空间中的核函数实现的。

根据泛函的有关理论，只要有一种核函数$K(\vec{x}_i*\vec{x}_j)$满足Mercer条件，它就对应某一变换空间中的内积。

核函数是一个对称函数K：$R^n \times R^n \rightarrow R$,它将两个n维空间的n维向量映射为一个实数。

$$K(\vec{x}_i*\vec{x}_j) = \psi(\vec{x}_i) * \psi(\vec{x}_j)$$

其中ψ将特征向量从n维空间映射到更高为空间。

总的来说：

核函数就是为了减少高维空间点积的运算量的，怎么减少呢？其实就是通过结合高维映射函数，写出计算公式，减少计算量罢了。这个问题就好像：计算1 + 2 + 3 + ……n 和计算 $\frac{(1+n)\,n}{2}$ 一个道理。通过找出其中的函数关系，减少计算量。

三、用SVM区分多类问题

1.一对多的响应策略

假设有A、B、C、D四类样本需要划分。在抽取训练集的时候，分别按照以下4种方式划分：

- A所对应的样本特征向量作为正样本集合（+1），B、C、D均作为负样本集合（-1）。
- B所对应的样本特征向量作为正样本集合（+1），A、C、D均作为负样本集合（-1）。
- C所对应的样本特征向量作为正样本集合（+1），B、A、D均作为负样本集合（-1）。
- D所对应的样本特征向量作为正样本集合（+1），B、C、A均作为负样本集合（-1）。

将上述4个训练集分别进行训练，得到4个SVM的分类器。在测试的时候，把未知类别的测试样本$\vec{x}$分别送入这4非分类器进行判决，最后每个分类器都有1个响应，分别为$f_1(\vec{x})$，$f_2(\vec{x})$，$f_3(\vec{x})$，$f_4(\vec{x})$，最终的决策结果就是max($f_1(\vec{x})$，$f_2(\vec{x})$，$f_3(\vec{x})$，$f_4(\vec{x})$),即4个响应中的最大者。（这里为什么用的是$f(\vec{x})$而不是$h(\vec{x})$呢？其实用$h(\vec{x})$也是可以的，不过错误率会增加，除非上面4个样本分类全部分对。使用$f(\vec{x})$则不仅仅考虑到分的样本的正负性，还考虑到了样本距离分割面的距离，距离越远则该分类面分错它的几率就越小；反之，距离越是近，则有可能该样本被分错。）

2.一对一的投票策略

将A、B、C、D四类样本两类两类地分成几组，即（A，B），（A，C），（A，D），（B，C），（B，D），（C，D），得到6个二分类器，采用投票的形式，最后得到结果。投票过程如下：

初始化vote(A) = vote(B) = vote(C) = vote(D) = 0。

每个需要被判断的特征依次跑这6个二分的分类器。每个分类器得到结果后进行投票。比如第一个分类器只能对A和B投票。一个样本是哪一类就对那个集合投票。假设$\vec{x}$被分为A类，则vote(A) += 1;最后，取vote(A) = vote(B) = vote(C) = vote(D)中的最大值，即$\vec{x}$属于max(vote(A) , vote(B) , vote(C) , vote(D))那一类。如果有两个最大值，则可以随便取一个。

仔细考虑这种投票方法，其实还是蛮有道理的。假如$\vec{x}$是属于B类的特征。那么在（A，B），（B，C），（B，D）的分类器上很大几率会被分对，这样B的票数自然会比较高。

3.一对一的淘汰策略

将A、B、C、D四类样本两类两类地分成几组，即（A，B），（A，C），（A，D），（B，C），（B，D），（C，D），得到6个二分类器。对六个分类器进行标号：一号（A，B），二号（A，C），三号（A，D），四号（B，C），五号（B，D），六号（C，D）。标号越前表示该分类器的置信度越高。那么如果判断两个分类器的置信度呢？很简单，如果分割超平面的分类间隔越是大，那么其置信度越高。为什么呢？间隔越大，就表明该分类器更有把握区分这两类。即这个分类器出错的概率是最小的。所以当然要首选出错概率最小的分类器来分啦！假设现在有一个测试样本A进来。假设分类器排序就是上面的样子。那么一号分类器有很大的把握（是所有分类器把握最大的）将A分对，同时淘汰B。说明输入的样本不可能是B。那么和B有关的分类器就不需要参与搅和了！那么直接淘汰掉四号（B，C），五号（B，D）分类器。一号分类器用过了，现在剩下二号（A，C），三号（A，D）分类器。按照上面的方法继续后面两个分类器判断。最后三号分类器则可以给出结果。

总的来所，这种策略使用的思想是，以最有把握的方法淘汰掉不可能的情况，最后剩下的一个就很有可能被分对，因为干扰的分类器少了。其实仔细考虑上面的算法，我自己还发现，有可能一个分类器算出来既不是+1也不是-1，而是0。这种情况还是有可能出现的。所以我觉得上面的算法应该增加对这种情况的处理。我觉得最简单的办法就是直接淘汰放弃当前使用的分类器。既然它不能区分，那么就不用了呗。

附：

看了上面3中区分方法，发现他们不仅仅可以用在SVM，它们适合于各种二分类器。比如老师详细讲过的Adaboost也可以使用。即它们适合宇宙中的所有二分器。