Hans Journal of Data Mining 数据挖掘, 2020, 10(3), 163-175
Published Online July 2020 in Hans. http://www.hanspub.org/journal/hjdm
https://doi.org/10.12677/hjdm.2020.103017

Hans汉斯

Understanding of Theory and Experiments in Support Vector Machine Classification

Fangfang Du, Lingchun Xiong

School of Science, Beijing Technology and Business University, Beijing
Email: dff_btbu@163.com

Received: Apr. 20th, 2020; accepted: May 12th, 2020; published: May 21st, 2020

Abstract

In the era of big data, machine learning is indispensable for data processing. Various colleges and universities are carrying out the course of machine learning in many majors. As one of the important contents, support vector machine algorithm is widely favored by students. But in the process of teaching, we find that its theory is rather obscure in some aspects. After summarizing the theory of support vector machine classification, this paper summarizes several key points' understanding, such as, the understanding of maximizing margin in support vector machine linear classifier model; the understanding of some knowledge points in the derivation of objective functions and constraints in linear classifier model; in the process of solving linear classifier model, why is the data point corresponding to Lagrange multiplier greater than 0 support vector? Is the support vector machine classification face unique? Then, the experimental codes of several softwares such as MATLAB, R language and python on iris data and the understanding of relevant results are expounded. The understanding of these details and the software experiment have carried on the thorough analysis and the fusion to the support vector machine model, which is helpful for the preliminary exploration and in-depth study of SVM classification learning, having provided the important reference for the broad support vector machine learner and the user.

Keywords

Support Vector Machine Classification, Understanding of Maximizing Margin, Knowledge Points Understanding of Linear Classifier Model, Support Vectors' Determination, Iris Data Experiments on Matlab, R and Python

支持向量机分类理论中的几个细节理解及实验

杜芳芳，熊令纯

北京工商大学数学与统计学院，北京
Email: dff_btbu@163.com

收稿日期：2020年4月20日；录用日期：2020年5月12日；发布日期：2020年5月21日

文章引用: 杜芳芳, 熊令纯. 支持向量机分类理论中的几个细节理解及实验[J]. 数据挖掘, 2020, 10(3): 163-175.
DOI: 10.12677/hjdm.2020.103017

摘　要

目前的大数据时代，机器学习作为数据处理的关键技术，不可缺少。各个高校在很多专业都在开展机器学习这门课程，其中的支持向量机算法作为重要内容之一，广受学生青睐。但在教学过程中，我们发现其理论在某些方面比较晦涩难懂，本文在简要概述完支持向量机分类的理论后，总结了几个关键细节的理解，比如：支持向量机线性分类器模型中的最大化Margin的理解；线性分类器模型中目标函数和约束条件的推导过程中某些知识点的理解；线性分类器模型求解过程中，为什么拉格朗日乘子大于0对应的数据点就是支持向量？支持向量机的分类面唯一吗？最后，给出了Matlab、R语言、python等几种软件在iris数据上的实验代码及实验中得到的支持向量机模型的获取方法。这些细节的理解及软件实验，对支持向量机模型都进行了透彻的剖析和融会贯通，有助于支持向量机分类学习中的初步探索和深入的研究，给广大支持向量机的学习者和使用者提供了重要的参考。

关键词

支持向量机分类，最大化Margin的理解，线性分类器模型的相关知识点理解，支持向量的确定，基于Matlab、R和Python的Iris数据实验

Copyright © 2020 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 支持向量机的基本理论框架

处在大数据的时代，从数据中挖掘各种信息和关联关系，机器学习方法在其中的应用显得非常重要。统计学习理论基础上发展起来的支持向量机(support vector machine, SVM)，有一套严实的统计学习理论作为支撑[1]，是机器学习的重要算法之一，在处理有限样本的问题上很有优势，已经广泛应用于各行各业，比如：图像识别与分类、文本分类、股票预测、生物医学等等[2] [3] [4] [5] [6]。其理论过程的一般形式如下面几个方面[7] [8] [9] [10]，从这些知识点中，我们给出一些细节的精细理解。

1) **SVM 的线性分类器模型**

一般，先从感知机的二分类问题进行引入。当感知机就一个神经元用于处理信息时，其结构图如图 1。

当其中的激活函数 f 取符号函数 sign 时，设 $W=(w_1,w_2)^{\mathrm{T}}$，$x=(x_1,x_2)^{\mathrm{T}}$，则这个神经元处理完信息以后的输出是：

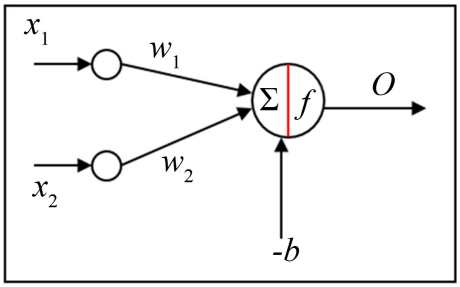


Figure 1. Information processing of a neuron
图 1. 单神经元的信息处理情况

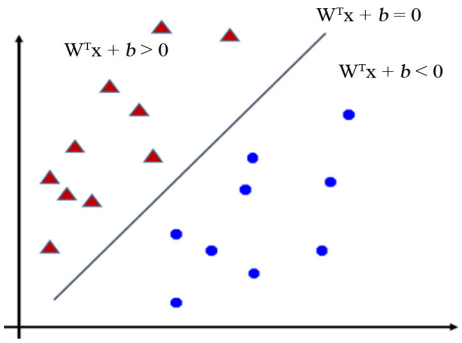


Figure 2. Binary classification problem
图 2. 二分类问题

$$o = sign\left(w_1 * x_1 + w_2 * x_2 + b\right) = sign\left(W^T x + b\right)，\text{其中 } sign\left(x\right) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} 。$$

这实际上相当于图 2 中的二分类问题，圆点数据点这一类对应输出是−1，三角数据点一类对应输出是 1。也就相当于在两类样本点中间找一条直线把它们分开。这样的直线有很多，如下图 3，支持向量机就是围绕着去寻找中间那条最好的直线而进行推导。二维空间里是分类线；三维空间里是分类平面；四维空间以上，则称为分类超平面，为了便于理解和观察，以下主要在二维空间上作图和进行阐述。

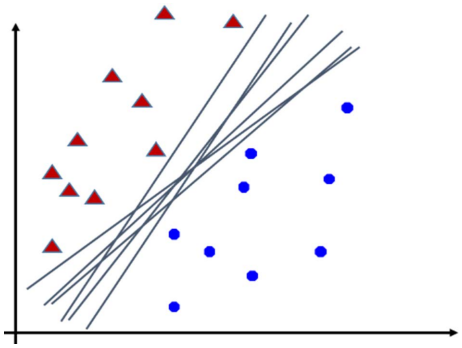


Figure 3. Classification lines in binary classification problem
图 3. 二分类问题的多条分类直线

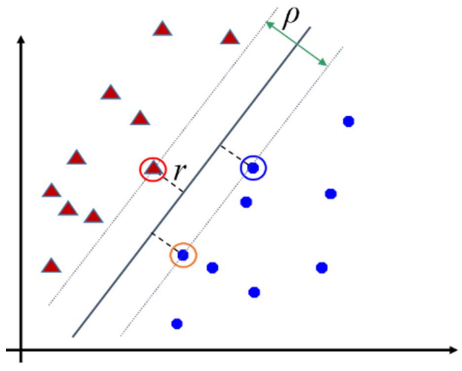


Figure 4. The best classification line in binary classification problem
图 4. 二分类问题中最好的分类线情况

分类线距离哪一侧的样本点越远，那么这类样本点就被分类得更好，并且对于这类中新来的测试样本点也会被分得更好，从而具有更好的泛化性。两侧的样本点都按照这样的原则，那么就会出现图 4 的情形，中间那条分类线只能是距离两边最近的样本点都一样远，最近的样本点叫支持向量，并且过两边的支持向量分别作平行于分类线的直线，那么最好的分类线就是要使得过两边支持向量的平行线之间的距离最大，即图 4 中的 ρ 最大。这是从两条平行线的“函数间隔”来分析出 ρ 的最大化。那么，得到初步的线性分类器模型可以是这样：

$$\begin{aligned}&\max\ \ \rho\\&\text{s.t.}\ \ y_i\left(W^{\mathrm{T}}x_i+b\right)\geq\rho/2, i=1,2,\cdots,n\end{aligned}\tag{1}$$

于是，中间那条分类线相当于 $W^{\mathrm{T}}x_i+b=0$；对于圆点的数据点有 $W^{\mathrm{T}}x_i+b\leq-\rho/2$，对于三角的数据点有 $W^{\mathrm{T}}x_i+b\geq\rho/2$。

再根据支持向量数据点到分类线的距离，推导出关于“几何间隔”最大化的模型：

$$\begin{aligned}&\max\ \ \rho/\|W\|\\&\text{s.t.}\ \ y_i\left(W^{\mathrm{T}}x_i+b\right)\geq\rho/2, i=1,2,\cdots,n\end{aligned}\tag{2}$$

最后，经过重新量化，去掉 ρ，得到最终的线性分类器模型如下：

$$\begin{aligned}&\min\frac{1}{2}W^{\mathrm{T}}W\\&\text{s.t.}\ \ y_i\left(W^{\mathrm{T}}x_i+b\right)\geq1, i=1,2,\cdots,n\end{aligned}\tag{3}$$

这里的约束条件，相当于，对于圆点的数据点有 $W^{\mathrm{T}}x_i+b\leq-1$，对于三角的数据点有 $W^{\mathrm{T}}x_i+b\geq1$。

模型(3)的求解过程是通过***拉格朗日乘子法***，为每一个训练数据点添加拉格朗日乘子 $\alpha_i\geq0, i=1,2,\cdots,n$，原问题转化为

$$L_P\left(W,b,\alpha_i\right)=\min_{w,b}\max_{\alpha_i\geq0}\left\{\frac{1}{2}W^{\mathrm{T}}W-\sum_{i=1}^{n}\alpha_i\left(y_i\left(W^{\mathrm{T}}x_i+b\right)-1\right)\right\},$$

再通过 L_P 对 W,b 求偏导等于 0，得到

$$\begin{aligned}&\frac{\partial L(W,b)}{\partial W}=0\Rightarrow W-\sum_{i=1}^{n}\alpha_i y_i X_i=0\\&\frac{\partial L(W,b)}{\partial b}=0\Rightarrow\sum_{i=1}^{n}\alpha_i y_i=0\end{aligned}\tag{4}$$

于是，得到对偶问题

$$\begin{aligned}&\max\ \ L_D(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j X_i^{\mathrm{T}}X_j\\&\text{s.t.}\ \ \sum_{i=1}^{n}\alpha_i y_i=0,\ \alpha_i\geq0\end{aligned}$$

然后，用 SMO 算法进行对偶问题求解，求出各个 $\alpha_i, i=1,2,\cdots,n$，再带回(4)式，就得到 W 的值。这里求出来的 α_i 中大于 0 的就对应支持向量 i。再由各个支持向量满足的表达式 $\alpha_i\left[y_i\left(W^{\mathrm{T}}X_i+b\right)-1\right]=0$ 求出 b、并进行平均得到最终的 $\hat{b}$，从而得到最终的分类线如下

$$f(X)=\sum_{i=1}^{n_s}\hat{\alpha}_i y_i X_i^{\mathrm{T}}\cdot X+\hat{b}.$$

2) SVM 的软边界模型

实际生产和生活中，很多数据往往是线性不可分的，但，如果还想用线性分类器去分，又想尽量减少分类错误，那么可以像下图 5 那样，为每个数据点加上松弛变量 $\xi_i \geq 0$，使得，对于三角的数据点，由原来的 $W^{\mathrm{T}}X+b \geq 1$ 松弛到 $W^{\mathrm{T}}X+b \geq 1-\xi_i$；对于圆点的数据点，由原来的 $W^{\mathrm{T}}X+b \leq -1$ 松弛到 $W^{\mathrm{T}}X+b \leq -1+\xi_i$。

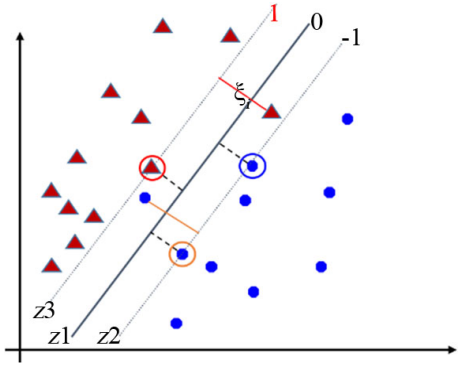


Figure 5. Soft margin classification problem
图 5. 软边界分类问题

得到的相应“软边界”模型如下：

$$\min \ \Phi(W)=1/2W^{\mathrm{T}}W+C\sum\xi_i$$
$$\text{s.t.} \ \ y_i\left(W^{\mathrm{T}}x_i+b\right)\geq 1-\xi_i,\xi_i\geq 0,i=1,2,\cdots,n$$

对于这个模型，把松弛变量 ξ_i 加进目标函数里作为惩罚，惩罚的力度由参数 C 来决定。其求解过程和上面线性分类器模型的求解过程类似。

3) SVM 中的核函数

通过函数映射 $\varphi(X)$，把数据从低维空间映射到高维空间，使得低维空间里线性不可分的数据到了高维空间里变成线性可分，从而构造高维空间里的线性 SVM 分类器进行解决，如下图 6。

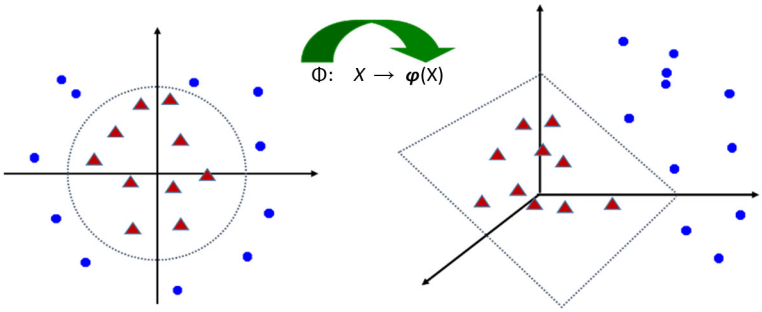


Figure 6. The mapping of linearly inseparable data in two-dimensional space to three-dimensional space to make it linearly separable
图 6. 把二维空间线性不可分的数据映射到三维空间使其线性可分

但那样的映射，一般很难找，而在 SVM 的模型求解过程的对偶问题模型中以及求解的结果中，主要涉及到数据点的内积形式，比如下面两个式子中类似 $X_i^{\mathrm{T}}X$ 形式，

$$\max L_D(\lambda)=\sum_{i=1}^{Q}\lambda_i-\frac{1}{2}\sum_{i=1}^{Q}\sum_{j=1}^{Q}\lambda_i\lambda_j y_i y_j\left(X_i^{\mathrm{T}}X_j\right)$$

$$f(X)=\sum_{i=1}^{n_s}\hat{\alpha}_i y_i X_i^{\mathrm{T}}\cdot X+\hat{b}$$

如果想象一下，数据点映射到高维空间，上面两个式子里也就对应 $\varphi(X_i)^{\mathrm{T}}\varphi(X_j)$ 形式，于是研究者们想到去找这样的函数 $K(X_i,X_j)=\varphi(X_i)^{\mathrm{T}}\varphi(X_j)$，即找 K 函数，这就是核函数，而不是去找 φ。于是，在引入核函数以后，得到 SVM 的分类超平面为：

$$f(X)=\sum_{i=1}^{n_s}\hat{\alpha}_i y_i K(X_i^{\mathrm{T}}\cdot X)+\hat{b}$$

对于新来的测试样本 X，可以带入上式，算出其决策函数值 $f(X)$，从而决定其类属。这样的 K 函数在一般的软件里通常有：

线性核函数：$K(X_i,X_j)=X_i'*X_j$；

多项式核函数：$K(X_i,X_j)=(\sigma*X_i'*X_j+b)^p$；

径向基核函数(或者叫高斯核函数)：$K(X_i,X_j)=\exp\left(-\frac{\|X_i-X_j\|^2}{2\sigma^2}\right)$；

sigmoid 核函数：$K(X_i,X_j)=\tanh(\sigma*X_i'*X_j+b)$。

2. 理论教学中的几个关键细节理解

在上面描述的支持向量机理论过程中，有很多关键细节，在教学过程中，理解起来非常有难度，我们给出详细的解释。

理解 1：“函数间隔”与“几何间隔”，以及 SVM 最大化的 Margin 究竟是什么？

解释：将图 4 进一步细化，把其中的直线方程补充上，如下图 7。

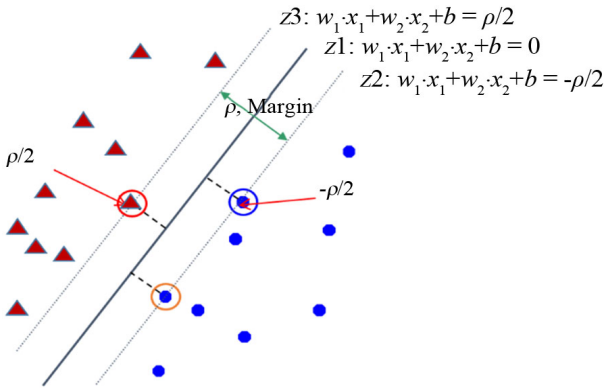


Figure 7. Three lines and margin in binary classification problem
图 7. 二分类中三条直线及 Margin 等情况

在上面线性分类器模型的推导过程中，图 7 中的 $z2$ 和 $z3$ 线就分别是两类样本点的边界，最大化两类样本的分类边界间的间隔，这个间隔，比较难理解，模型的推导过程，目标函数是怎么从最大化 ρ，到了最大化 $\frac{1}{2}W^{\mathrm{T}}W$ 的？

事实上，$z2$ 和 $z3$ 线之间的“函数间隔”就是这两条平行线的表达式做差的绝对值，就是 ρ，所以得到式(1)的模型；而它们之间的“几何间隔”就是其中一条线上的点到另一条线的距离，比如 $z2$ 上的点 $X=(x_1,x_2)$，首先满足方程 $w_1\cdot x_1+w_2\cdot x_2+b=-\rho/2$，然后到 $z3$ ($w_1\cdot x_1+w_2\cdot x_2+b-\rho/2=0$)距离为

$$r=\frac{\left|w_1\cdot x_1+w_2\cdot x_2+b-\rho/2\right|}{\sqrt{w_1^2+w_2^2}}=\frac{\left|-\rho/2-\rho/2\right|}{\sqrt{w_1^2+w_2^2}}=\frac{\rho}{\sqrt{w_1^2+w_2^2}}=\frac{\rho}{\|W\|},$$

所以，得到式(2)的模型。这样，**最大化的“Margin”从代数上讲是 ρ，从几何上讲是** $\frac{\rho}{\|W\|}$。那么，最大化哪一个更好呢？

实际上，要讲“几何间隔”最大，而不讲“函数间隔”最大。因为同一条直线的方程，将其各个项都乘以同一个倍数，它还是这条直线，但两条平行直线之间的“函数间隔”就变了，因为各自的常数项被乘以了倍数，做的差也比原来多了个倍数！比如，$z2$ 和 $z3$ 线都变成：

$$z2: 2w_1\cdot x_1+2w_2\cdot x_2+2b+\rho=0$$
$$z3: 2w_1\cdot x_1+2w_2\cdot x_2+2b-\rho=0$$

那么，“函数间隔”就变成 2ρ。而“几何间隔” $\frac{\rho}{\|W\|}$ 没变，因为求距离式子的分子分母都增大了同一个倍数，

$$r=\frac{\left|2w_1\cdot x_1+2w_2\cdot x_2+2b-\rho\right|}{\sqrt{\left(2w_1\right)^2+\left(2w_2\right)^2}}=\frac{\left|-\rho-\rho\right|}{\sqrt{\left(2w_1\right)^2+\left(2w_2\right)^2}}=\frac{2\rho}{\sqrt{\left(2w_1\right)^2+\left(2w_2\right)^2}}=\frac{\rho}{\|W\|}$$

所以要讲“几何间隔” $\frac{\rho}{\|W\|}$ 最大。当然，在下面一条理解里，当把图 7 中的三条直线一起重新量化后，发现要最大化的 Margin 实际上是量化后的 $\frac{2}{\|W\|}$。**解释完毕！**

理解 2：线性分类器模型中的目标函数如何由 $\frac{\rho}{\|W\|}$ **变成** $\frac{1}{2}W^{\mathrm{T}}W$ **的？约束条件又如何从** $y_i\left(W^{\mathrm{T}}x_i+b\right)\geq\rho/2$ **变成** $y_i\left(W^{\mathrm{T}}x_i+b\right)\geq 1$ **的？**

解释：如果把图 7 中的三条直线的方程分别都除以 $\rho/2$，以改变比例、重新量化，得到

$$z3: w_1/(\rho/2)\cdot x_1+w_2/(\rho/2)\cdot x_2+b/(\rho/2)=1$$
$$z1: w_1/(\rho/2)\cdot x_1+w_2/(\rho/2)\cdot x_2+b/(\rho/2)=0$$
$$z2: w_1/(\rho/2)\cdot x_1+w_2/(\rho/2)\cdot x_2+b/(\rho/2)=-1$$

或者，写成

$$z3: \overline{w}_1\cdot x_1+\overline{w}_2\cdot x_2+\overline{b}=1$$
$$z1: \overline{w}_1\cdot x_1+\overline{w}_2\cdot x_2+\overline{b}=0$$
$$z2: \overline{w}_1\cdot x_1+\overline{w}_2\cdot x_2+\overline{b}=-1$$

或者，再写成

$$\begin{aligned} z3&: w_1\cdot x_1+w_2\cdot x_2+b=1\\ z1&: w_1\cdot x_1+w_2\cdot x_2+b=0\\ z2&: w_1\cdot x_1+w_2\cdot x_2+b=-1 \end{aligned} \tag{5}$$

则，一个支持向量到分类线的距离就变成 $r=\frac{1}{\|W\|}$，那么分类线两侧的距离之和，也就是 $z2$ 和 $z3$ 线之间

的距离就是$\frac{2}{\|W\|}$，于是，**应该追求$\frac{2}{\|W\|}$的最大化，即这就是支持向量机最终要最大化的 Margin。**同时，为了求解的方便，也相当于追求$\frac{1}{2}\|W\|^2$的最小化，或者追求$\frac{1}{2}W^{\mathrm{T}}W$的最小化。

另外，根据(5)式，*z*2 和 *z*3 线两边的数据点必须满足约束条件$y_i\left(W^{\mathrm{T}}X_i+b\right)\geq 1$。于是，最终的线性分类器模型就是：

$$\min \frac{1}{2}W^{\mathrm{T}}W$$
$$\text{s.t. } y_i\left(W^{\mathrm{T}}X_i+b\right)\geq 1,\ i=1,2,\cdots,n.$$

解释完毕！

理解 3：把线性分类器模型转换成对偶问题求解以后，得到的$\hat{\alpha}=(\hat{\alpha}_1,\cdots,\hat{\alpha}_n)^{\mathrm{T}}$，为什么其中$\hat{\alpha}_i>0$对应的样本 *i* 就是支持向量[11]？

解释：在下面式子中

$$L_P(W,b,\alpha_i)=\min_{w,b}\max_{\alpha_i\geq 0}\left\{\frac{1}{2}W^{\mathrm{T}}W-\sum_{i=1}^{n}\alpha_i\left(y_i\left(W^{\mathrm{T}}X_i+b\right)-1\right)\right\},$$

关于α_i求最大时(把 W 和 b 当常数看)，得到

$$\max_{\alpha_i\geq 0}L_P(W,b,\alpha)=\max_{\alpha_i\geq 0}\frac{1}{2}W^{\mathrm{T}}W-\sum_{i=1}^{n}\alpha_i\left(y_i\left(W^{\mathrm{T}}X_i+b\right)-1\right),\tag{6}$$

如果X_i是支持向量，那么就有$y_i\left(W^{\mathrm{T}}X_i+b\right)-1=0$，即$y_i\left(W^{\mathrm{T}}X_i+b\right)=1$，则(6)式的最大化就只由$\frac{1}{2}W^{\mathrm{T}}W$决定，而$\alpha_i$取任何值都对目标函数的最大化没有影响，其中就可以取大于 0 的值；而对于非支持向量，则有$y_i\left(W^{\mathrm{T}}X_i+b\right)-1>0$，即$y_i\left(W^{\mathrm{T}}X_i+b\right)>1$，而$\alpha_i\geq 0$，则有$\sum_{i=1}^{n}\alpha_i\left(y_i\left(W^{\mathrm{T}}X_i+b\right)-1\right)\geq 0$，为了满足最大化的要求，必须$\alpha_i=0$，才能使得$\sum_{i=1}^{n}\alpha_i\left(y_i\left(W^{\mathrm{T}}X_i+b\right)-1\right)=0$，从而使得(3)式最大化。这样一来，只有对应支持向量的α_i有可能大于 0，而非支持向量对应的α_i一定等于 0；当然，α_i等于 0 的不一定是非支持向量。综上，$\alpha_i>0$对应的数据点X_i就一定是支持向量。**解释完毕！**

理解 4：对于同一个训练集，得到的分类超平面，唯一吗？

解释：分类超平面，在二维平面里，就是分类直线。从下面图 8 中的三个图来看，图(1)中实线是分类线，在保持两条虚线之间距离不变的同时(即保持 Margin 不变)，三条线一起做适当的旋转，得到图(2)中的三条粗一点的线，此时，图(1)中的支持向量和图(2)中的支持向量稍微有些变化。旋转以后形成的支

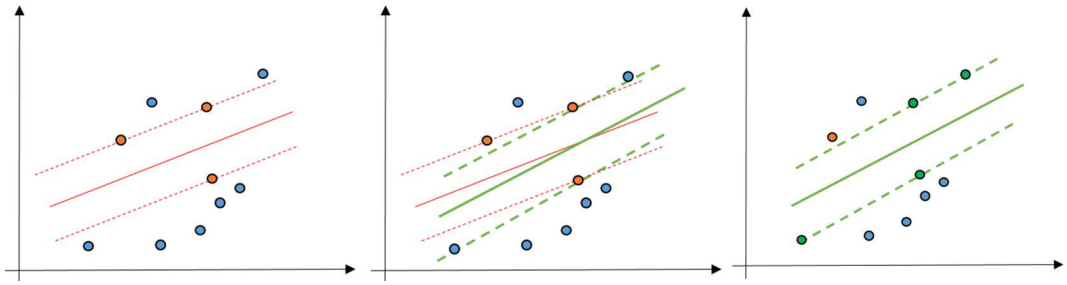


Figure 8. Non-unique case of support vector machine classification line
图 8. 支持向量机分类直线的不唯一情况

持向量机分类情况如图(3)。由此可以看出，从理论上讲，对于同一个训练集，得到分类直线可以不唯一，或者得到的分类超平面可以不唯一，但从实验出发，很多数据点都只能算出来一个分类超平面，而不是恰好能有两个分类超平面的情况，实验方面还需要恰好能产生这样的数据点来进行实践。

解释完毕！

3. Matlab、R 语言和 Python 在 Iris 数据上的实验

下面给出三个软件上的实验代码及得到的实验结果中如何获取支持向量机模型。

1) Matlab 实验代码

我们的实验，使用了台湾林志仁教授他们开发 Matlab 工具包 libsvm [12]。这个包，需要按照一定的流程在 matlab 里进行安装，才可以使用。

① 代码

```
load fisheriris
%工作空间中就出现了 meas 和 species 变量，前者是 150*4 的特征，后者 150 个类别标签
[iris_scale,ps]=mapminmax(meas',0,1); iris_scale=iris_scale'; %进行数据归一化
%从 species 看到，总共三类，每类 50 个样本，随机选择 60%当训练样本，其余样本做测试
%相当于每类随机选择 30 个样本用于训练，其余 20 个用于测试。下面开始准备训练样本和测试样本
Num=[randperm(50), 50+randperm(50), 100+randperm(50)];
train_data=iris_scale(Num([1:30, 51:80, 101:130]), :);
test_data=iris_scale(Num([31:50, 81:100, 131:150]), :);
labels=[ones(50,1);2*ones(50,1);3*ones(50,1)];
train_labels=labels(Num([1:30, 51:80, 101:130]));
test_labels=labels(Num([31:50, 81:100, 131:150]));

%C-SVC 中的参数 C 和核函数中的参数 gamma(简记为 g)的寻优[13]
[bestCVaccuracy,bestc,bestg]=SVMcgForClass(train_labels,train_data,-6,-1,-8,8,5,0.01,0.1,10)

%SVM 模型的训练与测试[14]
model = svmtrain(train_labels, train_data,    '-s 0 -t 2 -g 1 -c 2    -b 1');
[predict_labels, accuracy,prob] = svmpredict(test_labels, test_data, model,'-b 1');
```

② 实验中得到的 SVM 模型的获取

当 model 这个变量生成以后，可以在 matlab 命令窗口执行它，得到下面结果

```
model =
    Parameters: [5x1 double]
       nr_class: 3
        totalSV: 36
            rho: [3x1 double]
          Label: [3x1 double]
    sv_indices: [36x1 double]
          ProbA: [3x1 double]
          ProbB: [3x1 double]
            nSV: [3x1 double]
      sv_coef: [36x2 double]
            SVs: [36x4 double]
```

要获得最后的分类超平面 $f(X)=\sum_{i=1}^{n_s}\hat{\lambda}_i y_i K\left(X_i^{\mathrm{T}}\cdot X\right)+\hat{b}$，需要针对每个支持向量 i 获得核函数前面的系数 $\hat{\lambda}_i y_i$，以及分类面的截距 $\hat{b}$。这个实验，分三类，构造了三个分类超平面，即第一类和第二类之间的一个分类面，第一类和第三类之间一个分类面，第二类和第三类之间一个分类面，可以分别记为 1/2，1/3，2/3 的三个分类面。那么需要找到每个分类面涉及到的系数和截距。

首先，截距的获得比较简单，在 Matlab 命令窗口执行 model.rho，就得到三个数：0.0745、0.2416、0.0592，这就是那三分类面截距的相反数。

其次，在命令窗口执行 model.sv_coef 就可以得到 36 × 2 的向量，这是那 36 个支持向量中每个支持向量对应的两个系数，为什么是两个系数呢？因为我们的分类面是 1/2，1/3 和 2/3，那么第一类的样本就有可能在 1/2 分类面里是支持向量，也有可能在 1/3 里是支持向量；第二类的样本有可能在 1/2 中是支持向量，也有可能在 2/3 中是支持向量；同理，第三类的样本有可能在 1/3 里是支持向量，也有可能在 2/3 里是支持向量，且这些系数值的正负情况，就表明它们在一个分类面的相应正负类里当作支持向量的。具体地，我们取出第 2、3 号样本支持向量对应的系数、以及第 41、90 号样本对应的系数出来稍做分析，它们的系数值见下面的表 1，为了节省空间，其他支持向量的系数就不再列入表格。

2 号样本的两个系数的值都是 2，都非零，且 2 号样本本身是第一类的，说明它在 1/2 和 1/3 中都当了支持向量，并且这两个系数值就相当于它在下面分类面中的 $\hat{\lambda}_2 y_2$ 的值；

3 号样本的两个系数值分别是 0.2580 和 0，后面一个系数值是 0，且 3 号样本本身是第一类的，说明它只在 1/2 分类面中当了支持向量，而在 1/3 中不是支持向量，且在 1/2 中的作用相当于 1/2 分类面上面方程中的 $\hat{\lambda}_3 y_3$ 的值；

41 号样本的两个系数的值是−0.2144 和 2，都非零，根据上面第④项中关于 sv_indices 的分析来看，该样本本身属于第二类，那么它有在 1/2 和 2/3 这两个分类面中都当了支持向量，第一个系数是−0.2144，说明它在 1/2 中当了支持向量，且这个系数值是负数，说明它在 1/2 分类面的负类中当了支持向量；第二个系数值 2，说明它在 2/3 中当了支持向量，且是正类的支持向量；

90 号样本，是训练集中的最后一个样本，且它本身是第三类的，其两个结果系数是−0.4835 和 0，说明它在 1/3 当了支持向量，而在 2/3 中没有当支持向量，且因为它的值为负的，更说明了该样本在 1/3 这个分类面的负类中当作了支持向量的，这个非零系数值就是它在分类面 1/3 中的 $\hat{\lambda}_{90} y_{90}$。

实际上，根据 SVM 的原理、实验所选取的核函数和 model.sv_coef 中的数据，就可以把三个分类面依次写出来。由于这里数据太多，就不再一一分析和整理，留给读者去实验和把三个模型写出来。

Table 1. The coefficients of support vector in the classification plane
表 1. 支持向量在分类面中的系数结果

样本序号	2	3	41	90
系数 1	2	0.2580	−0.2144	−0.4835
系数 2	2	0	2	0

2) 基于 R 语言的实验

R 语言里，使用了 e1071 包，它提供了对 libsvm 的接口。

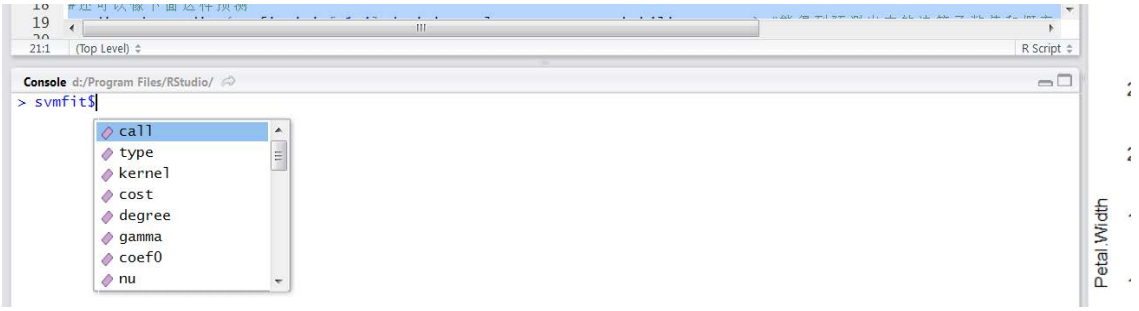
① 代码

```
#install.packages('e1071') #第一次使用时需要安装该包
library(e1071)
inputData <- data.frame(iris[,1:4], response = as.factor(iris$Species))
```

```
# 准备训练集和测试集
set.seed(100) # 为了实验结果的可重复性
rowIndices <- 1 : nrow(inputData) # 准备行标
sampleSize <- 0.6 * length(rowIndices) # 训练集大小
#这里是总体样本随机选择 60%当训练样本
trainingRows <- sample (rowIndices, sampleSize) # 随机抽样
trainingData <- inputData[trainingRows, ] # 训练集
testData <- inputData[-trainingRows, ] # 测试集
tuned <- tune.svm(response ~., data = trainingData, gamma = 10^(-6:-1), cost = 10^(1:2))
# tune 寻找最优参数
summary (tuned) # 看参数寻优的结果并根据最小误差选择最好的 gamma 和 cost
#下面这个语句中的参数要根据上面这个语句的结果而定
svmfit <- svm (response ~ ., data = trainingData, kernel = "radial", cost = 10, gamma=0.1,
scale = FALSE)
table (testData$response, predict(svmfit, testData)) # 预测结果和实际结果的比较
mean(testData$response != predict(svmfit, testData)) # 错分率计算
```

② 实验中得到的 SVM 模型的获取

在 Rstudio 命令窗口(Console)，输入 svmfit$就可以看到如下提示



在提示里，我们就可以选择所提示的各个项并回车，去查看一些实验参数及结果。

接下来，如果在上面图里的界面里$后面选择 rho，即在 Console 窗口执行了 svmfit$rho，就得到三个分类面的截距；同理，执行了 svmfit$coefs，就能得到各个分类面的系数；这二者都和上面 Matlab 实验里的解释类似，这里不再多说。

3) 基于 python 的实验代码

Python 实验里使用了其中的 sklearn 中的 svm 包，sklearn.svm 模块提供了很多模型供我们使用，本文使用的是 svm.SVC，它是基于 libsvm 实现的。.

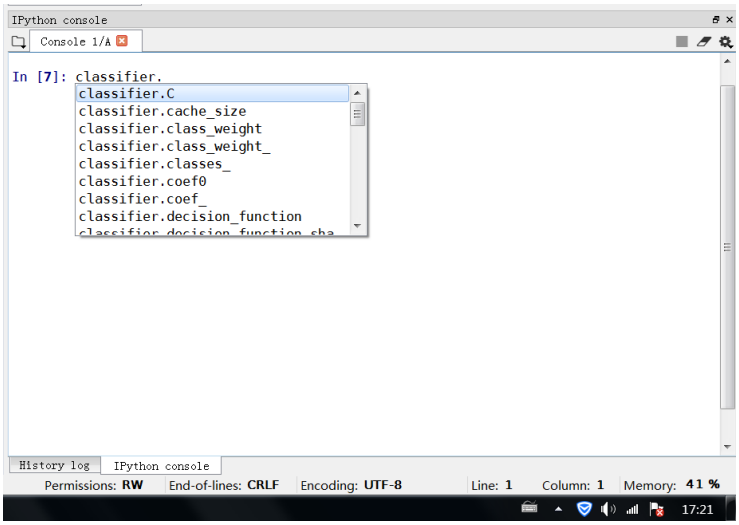
① 代码

```
from sklearn.datasets import load_iris
from sklearn.svm import SVC
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cross_validation import train_test_split
from sklearn.metrics import accuracy_score
iris = load_iris()
x = iris.data[:,0:4]
```

```
y = iris.target
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1, train_size=0.6)
classifier=SVC(C=2,kernel='rbf',gamma=10,decision_function_shape='ovo') # ovr:一对多策略
classifier.fit(x_train,y_train) #ravel 函数在降维时默认是行序优先
train_label=a.predict(x_train)
test_label=a.predict(x_test)
print("训练集：",accuracy_score(y_train,train_label))   #计算 svc 分类器对训练集的准确率
print("测试集：",accuracy_score(y_test,test_label))   #计算 svc 分类器对测试集的准确率
```

② 实验中得到的 SVM 模型的获取

在 Python 的 Console 窗口写 classifier.，然后按 Tab 键，就可以得到下图所示的下拉选项，在选项里选择某项，并回车，就得到了相应的实验结果参数



同样的，可以在上面窗口里进行选择，找到三个分类面的系数和截距。

4. 总结

本文将 SVM 分类的理论知识的主体结构进行了概述，将 SVM 模型本身及其推导过程中的某些细节的理解进行了精细的解释；最后还给出了目前大家常用的三个软件上的实验，即 Matlab、R 语言和 Python 在 iris 数据上的实验，还给出了寻找各个实验中所用到的相应支持向量机模型的方法。这些工作，从不同的角度、不同的计算机语言针对支持向量机的分类进行了理解和分析，这对于 SVM 的学习和使用提供了重要的参考，使得支持向量机的教学过程得到了进一步的提升。

参考文献

[1] Vapnik, V. (1998) Statistical Learning Theory. John Wiley & Sons Inc., New York.

[2] Zhang, X.D., Li, A. and Pan, R. (2016) Stock Trend Prediction Based on a New Status Box Method and Ada Boost Probabilistic Support Vector Machine. *Applied Soft Computing*, **49**, 385-398. https://doi.org/10.1016/j.asoc.2016.08.026

[3] Goudjil, M., Koudil, M., Bedda, M., *et al*. (2018) A Novel Active Learning Method Using SVM for Text Classification. *International Journal of Automation and Computing*, **4**, 1-9. https://doi.org/10.1007/s11633-015-0912-z

[4] Wang, Y. and Duan, H. (2018) Classification of Hyperspectral Images by SVM Using a Composite Kernel by Em-

ploying Spectral, Spatial and Hierarchical Structure Information. *Remote Sensing*, **10**, 26. https://doi.org/10.3390/rs11010026

[5] Singh, G. and Chhabra, I. (2018) Effective and Fast Face Recognition System Using Complementary OC-LBP and HOG Feature Descriptors with SVM Classifier. *Journal of Information Technology Research*, **11**, 91-110. https://doi.org/10.4018/JITR.2018010106

[6] Huang, S., Cai, N., Pacheco, P.P., *et al*. (2018) Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer Genomics Proteomics*, **15**, 41-51. https://doi.org/10.21873/cgp.20063

[7] Hastie, T., Tibshirani, R. and Friedman, J. (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition, Springer, Berlin.

[8] 李航. 统计学习方法[M]. 北京: 清华大学出版社, 2012.

[9] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

[10] Brett Lantz. 机器学习与 R 语言[M]. 李洪成, 许金炜, 李舰, 译. 北京: 机械工业出版社, 2015.

[11] 支持向量机通俗导论(理解 SVM 的三层境界) [Z/OL]. https://blog.csdn.net/macyang/article/details/38782399/, 2014-08-23.

[12] Chang, C.-C. and Lin, C.-J. (2011) LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, Article No. 27. https://doi.org/10.1145/1961189.1961199

[13] 李洋. Matlab 技术论坛[Z/OL]. http://www.matlabsky.com

[14] 王小川, 史峰, 郁磊, 李洋. MATLAB 神经网络 43 个案例分析[M]. 北京: 北京航空航天大学出版社, 2013.