華東理工大學

模式识别大作业

题　　目　基于 SVM 的癫痫患者预测

学　　院　信息科学与工程

专　　业　控制科学与工程

组　　员　黄怡涛（Y20190059）

指导教师　赵海涛

完成日期：　**2019** 年 **12** 月 **08** 日

基于 SVM 的癫痫患者预测

组员：黄怡涛

一、实验背景

癫痫病困扰着世界近 1%的人口，自发性是癫痫的一个主要特点。对于许多患者来说，可以服用足够高剂量的抗惊厥药物来预防癫痫发作，但是患者经常会出现副作用。而且对于 20-40%的癫痫患者来说，药物可能是无效的，甚至在手术切除引起癫痫的脑组织后，许多患者仍会继续自发发作。尽管癫痫发作很少发生，但由于癫痫发作的自发性，患有癫痫的患者会持续感到焦虑。

一个有效的癫痫发作预测系统可以帮助到癫痫患者。为了使基于脑电信号（Electroencephalogram，EEG）的癫痫发作预测系统有效工作，计算算法必须可靠地识别出癫痫发作可能性增加的时期。如果可以识别出这些预示癫痫发作的大脑状态，则有可能设计出可以提醒患者癫痫发作的设备。患者可以避免驾驶或游泳等潜在危险活动，并且可以只在需要时使用药物，来防止即将来临的癫痫发作，从而降低药物的长期副作用。

二、整体方案

2.1 数据介绍

本次实验采用的是经过预处理和特征提取后得到的滤除了噪声的功率谱特征数据，每个样本数据是由 4096 个采样点 3 个电极信号的平均经过快速傅立叶变换（Fast Fourier transform，FFT）得到的，共分成了 20 个子频带（0-1Hz, 1-2 Hz, … , 19-20 Hz）。总共有 50 个带标签的癫痫数据样本和 50 个带标签的非癫痫数据样本作为训练集，还有另外 50 个癫痫数据样本和 50 个非癫痫数据样本作为验证集。

2.2 分类方法

对于这一个二分类问题，样本比较少，支持向量机被认为是在小样本训练集上最常用、效果最好的分类器之一，且具有良好的泛化能力，故本次实验选择了三种方法对测试集中的 100 个数据样本进行了分类，分别是一种作为对比的简单分类器、线性支持向量机和非线性支持向量机，下面将对其逐一介绍。

2.3.1 一种作为对比的简单分类器

该分类器首先计算了两个训练集中各子频带的平均功率谱密度值，然后比较测试集的某一样本点与这两个平均值的欧氏距离，与之距离较小的那个平均值所对应的类别即为所预测的测试样本的类别。该分类器原理简单，其实只是我们课上所讲的 KNN 算法和 Kmeans 聚类中的一个计算步骤，相比较于上述两种方法其缺少自我修正的机制，因而在本次实验中该简单分类器只作为一个对比，来评估支持向量机的实际分类效果。

2.3.2 线性支持向量机

线性支持向量机处理的是线性不可分的数据集。对于线性支持向量机的优化问题，就是在线性可分支持向量机的基础上加了一个松弛变量。

下面先来介绍一下线性可分支持向量机，即硬间隔支持向量机。我们定义一个**广义线性模型**，

$$f(x)=sign(wx+b) \tag{2-1}$$

在数据集中，各点离超平面的远近，可以表明该点分类预测的确信程度。离超平面越远，可以说这个点更“确切”地分类到对应的一类。假如超平面确定，$|w\cdot x+b|$ 可以相对地表示点 x 和超平面之间的远近，而其与标签是否同号，则可表示分类是否正确。由此可以定义**函数间隔**

$$\hat{\gamma}_i = y_i(w\cdot x_i + b) \tag{2-2}$$

一组数据集的函数间隔，可以用这组样本里离超平面最近的样本点的函数间隔来表示，这个样本点称为支持向量，即

$$\hat{\gamma} = \min_{i=1,\dots,N} \hat{\gamma}_i \tag{2-3}$$

在决定分离超平面时，只有支持向量起作用，其他样本点并不起作用。

如果成比例地缩放 w 和 b，超平面并没有改变，但是函数间隔却会发生变化。因此需要给函数间隔加上约束，使其变为**几何间隔**。其中 ||w|| 是 w 的 L2 范数。

$$\gamma_i = y_i(\frac{w}{\|w\|}\cdot x_i + \frac{b}{\|w\|}) \tag{2-4}$$

和函数间隔一样，数据集 T 的几何间隔为

$$\gamma = \min_{i=1,\dots,N} \gamma_i \tag{2-5}$$

几何间隔可以视作实例点到超平面的带符号距离。支持向量机的基本思想就是求解能够正确划分数据集并且几何间隔最大的分离超平面。对于线性可分的数据集而言，这个最大间隔分离超平面是唯一的，这里的几何间隔最大化称为硬间隔最

大化。

于是得到如下的约束最优化问题：

$$\begin{aligned} &\max_{w,b} \gamma \\ &s.t. \qquad y_i\left(\frac{w}{\|w\|}\cdot x_i+\frac{b}{\|w\|}\right)\geq\gamma \end{aligned} \tag{2-6}$$

相当于

$$\begin{aligned} &\max_{w,b} \frac{\hat{\gamma}}{\|w\|} \\ &s.t. \qquad y_i(w\cdot x_i+b)\geq\hat{\gamma} \end{aligned} \tag{2-7}$$

函数间隔的取值并不影响最优化问题的解，也就是说这是一个等价的最优化问题，于是我们可以将整个模型缩放到函数间隔为 1 的程度。另外注意到最大化 $\frac{1}{\|w\|}$ 等价于最小化 $\frac{1}{2}\|w\|^2$，于是得到最终的最优化问题

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &s.t. \qquad 1-y_i(w\cdot x_i+b)\leq 0 \end{aligned} \tag{2-8}$$

这是一个凸二次规划问题。对于这种有约束最优化问题，可以构建拉格朗日函数

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i(w^T\cdot x_i+b)-1) \tag{2-9}$$

我们可以将原始问题写成拉格朗日函数形式，再将其转化为对偶问题。而且原始问题和对偶问题的最优解，由于满足 Slater 条件，所以是一致的。写出原始问题的对偶问题

$$\max_{\alpha}\min_{w,b} L(w,b,\alpha) \tag{2-10}$$

要求解这个对偶问题，需要先对 L(w,b,α) 对 w 和 b 求极小，再对 α 求极大。首先求解极小问题，分别求拉格朗日函数对 w 和 b 的偏导，并令其等于 0，解得

$$\begin{aligned} &w=\sum_{i=1}^{N}\alpha_i y_i x_i \\ &\sum_{i=1}^{N}\alpha_i y_i=0 \end{aligned} \tag{2-11}$$

将其代入拉格朗日函数原公式，得到

$$L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i \tag{2-12}$$

接着求解极大问题。根据上面的推导，这个极大问题就是对偶问题，等价于以下

求解极小化问题

$$\begin{aligned}&\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i\\&s.t.\qquad\sum_{i=1}^{N}\alpha_i y_i=0\qquad\alpha_i\geq 0\end{aligned}\tag{2-13}$$

我们利用 KKT 条件继续推导，

$$\begin{aligned}&\nabla_w L(w^*,b^*,\alpha^*)=w^*-\sum_{i=1}^{N}\alpha_i^* y_i x_i=0\\&\nabla_b L(w^*,b^*,\alpha^*)=-\sum_{i=1}^{N}\alpha_i^* y_i=0\\&\alpha_i^*(y_i(w^*\cdot x_i+b^*)-1)=0,i=1,2,...,N\\&y_i(w^*\cdot x_i+b^*)-1\geq 0,i=1,2,...,N\\&\alpha_i^*\geq 0,i=1,2,...,N\end{aligned}\tag{2-14}$$

利用 KKT 条件的第一条，可以求得 w 的最优解

$$w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i\tag{2-15}$$

利用 KKT 条件的第四条，可以求得 b 的最优解

$$b^*=y_k-\sum_{i=1}^{N}\alpha_i^* y_i x_i x_k\tag{2-16}$$

于是得到了最终的分类决策函数

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i(x_i\cdot x_k)+b^*)\tag{2-17}$$

硬间隔支持向量机是一个严格的线性模型，即用一个超平面将数据分隔为两部分。但数据一般情况下很少是严格线性可分的，难以找到一个超平面可以完美地隔开这些数据，即使在训练模型的时候隔开了，当训练集或实际数据落入特征空间后，也很难说会被正确隔开（过拟合）。而且，如果模型把一些噪声当成了支持向量，那也会得到错误的结果。

缓解这个问题的一个方法是允许模型在一些样本上出错，由此可引出线性支持向量机即软间隔支持向量机的概念。

我们允许部分样本不满足约束：$y_i(w\cdot x_i+b)<1$。当然，在最大化间隔的同时，还是要让不满足约束的样本尽可能少。所以可以把优化目标写成

$$\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{m}L_{0/1}(y_i(w\cdot x_i+b))\tag{2-18}$$

C 是一个常数，C＞0。而 $L_{0/1}$ 代表 0/1 损失函数，如果括号里的值小于 1，则损失函数输出 1（代表有一个样本出错了），否则输出 0。所以 C 值控制了间隔有多“软”，当 C 无穷大时，优化目标会迫使所有样本都满足约束，而 C 越小，则可以允许越多样本不满足约束。在 SVM 的实际使用中，C 是一个相当重要的超参数。

而 0/1 损失函数由于不连续且非凸，所以在实际应用中经常用 **hinge 损失函数**（也被称为合页函数）代替，即

$$L_{hinge}(z)=\max(0,1-z) \tag{2-19}$$

当样本正确分类时，hinge 损失输出 0；而当样本被错误分类时，hinge 损失可以输出它和正确值之间的绝对值距离。

于是优化目标变成

$$\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{m}\max\{0,1-y_i(w\cdot x_i+b)\} \tag{2-20}$$

引入**松弛变量** $\xi_i=1-y_i(w\cdot x_i+b)$ 且令 $\xi_i\geq 0$，就能将上式改写为

$$\begin{aligned}&\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{m}\xi_i\\ &s.t.\qquad y_i(w\cdot x_i+b)\geq 1-\xi_i,\xi_i\geq 0\end{aligned} \tag{2-21}$$

从约束条件可以看出，松弛变量使得[公式]不一定非要为 1，也就是不一定要处在支持向量之外，也算正确分类，如下图所示。

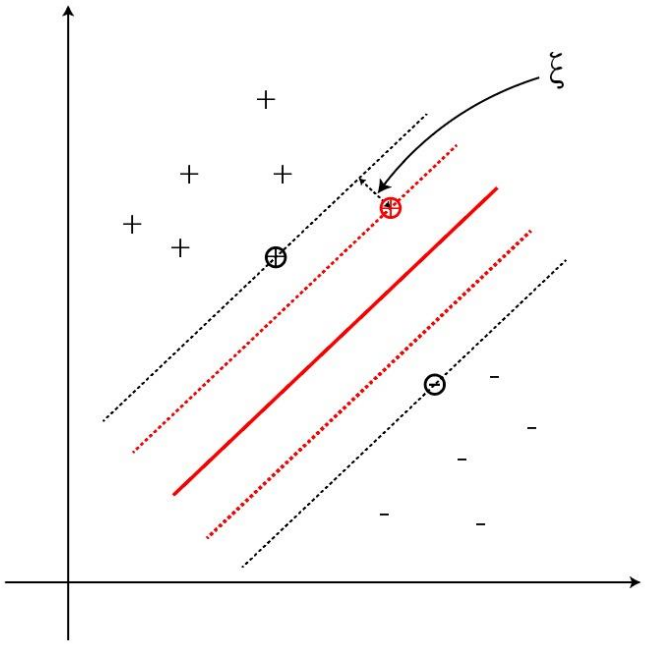

图 2-1 软间隔支持向量机超平面示例

同理可得对偶问题

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i \qquad (2\text{-}22)$$
$$s.t. \qquad 0 \le \alpha_i \le C$$

利用 KKT 条件，可求得

$$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i \qquad (2\text{-}23)$$
$$b^* = y_j - \sum_{i=1}^{N} y_i \alpha_i^* (x_i \cdot x_j)$$

经过上述推导，将线性支持向量机分类问题转换为了求解非线性规划问题，即约束条件为线性、目标函数为二次函数的优化问题，程序中采用了 quadprog 函数来进行求解。其一般调用格式为 x=quadprog（H,f,A,b,Aeq,beq,lb,ub,x0,options），其中各字段说明如下表所示。

表 2.1 quadprog 函数字段名说明

字段	说明	字段	说明
H	二次项矩阵	beq	线性等式约束的右端向量
f	一次项向量	lb	自变量下界约束
A	线性不等约束的系数矩阵	ub	自变量上界约束
b	线性不等约束的右端向量	X0	初始点
Aeq	线性等式约束的系数矩阵	options	Options 结构

2.3.3 非线性支持向量机

在线性不可分的情况下，SVM 通过某种事先选择的非线性映射（核函数）将输入变量映到一个高维特征空间，将其变成在高维空间线性可分，在这个高维空间中构造最优分类超平面。

如上已经得到，线性可分的情况下，可得最终的超平面方程为：

$$f(x) = \sum_{i=1}^{N}\alpha_i y_i \langle x_i, x\rangle + b \qquad (2\text{-}24)$$

对于线性不可分，我们使用一个非线性映射，将数据映射到特征空间，在特征空间中使用线性学习器，分类函数变形如下：

$$f(x) = \sum_{i=1}^{N}\alpha_i y_i \langle \phi(x_i), \phi(x)\rangle + b \qquad (2\text{-}25)$$

其中 ϕ 是从输入空间(X)到某个特征空间(F)的映射，这意味着建立非线性学习器分为两步：

（1）首先使用一个非线性映射将数据变换到一个特征空间 F；

（2）然后在特征空间使用线性学习器分类。

这里直接给出一个核技巧的定义：核函数是一个函数 k，对所有 x,z∈X，满足 k(x,z)=<ϕ(xi),ϕ(x)>，这里 ϕ(·)是从原始输入空间 X 到内积空间 F 的映射。

简而言之：如果不用核技术，就要先计算线性映 ϕ(x1)和 ϕ(x2)，然后计算这它们的内积；使用了核技术之后，可直接把 ϕ(x1)和 ϕ(x2)的一般表达式<ϕ(x1),ϕ(x2)>=k(x1, x2)计算出来，这里的<·，>表示内积，k(·，)就是对应的核函数，这个表达式往往非常简单，所以计算非常方便。

通过核技巧的转变，我们的分类函数变为：

$$f(x)=\sum_{i=1}^{N}\alpha_i y_i K(x_i,x)+b \tag{2-26}$$

我们的对偶问题变成了：

$$\begin{aligned}&\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i\\&s.t.\qquad \sum_{i=1}^{N}\alpha_i y_i=0\qquad \alpha_i\geq 0\end{aligned} \tag{2-27}$$

实际算法实现中，选择了核函数 $K(x,z)=(x\cdot z)^2$，并采用 MATLAB 内置 fmincon 函数求解，fmincon 函数可以求解带约束的非线性多变量函数(Constrained nonlinear multivariable function)的最小值,即可以用来求解非线性规划问题，其一般调用格式为[x,fval]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)，区别于上述 quadprog 函数，其中 fun 为定义的函数法 f(x)，代表了非线性目标函数；nonlcon 是定义的非线性向量函数约束

2.4 验证方法

所谓验证（Validation），是把数据集随机分成训练集，验证集，测试集（互斥）。用训练集训练出模型，然后用验证集验证模型，根据情况不断调整模型，选出其中最好的模型。

2.4.1 留出法

留出法(hold-out)：直接将数据集 D 划分为两个互斥的集合，其中一个集合作为训练集 S，另外一个作为测试集 T，即 D=S∪T,S∩T=0.在 S 上训练出模型后，用 T 来评估其测试误差，作为对泛化误差的评估。

我们希望评估的是用 D 训练出的模型的性能，但是留出法需划分训练/测试集，这就会导致一个窘境：若另训练集 S 包含大多数的样本，则训练出的模型可能更接近于 D 训练出的模型，但是由于 T 比较小，评估结果可能不够稳定准确；若另测试集 T 包含多一些样本，则训练集 S 与 D 的差别更大，被评估的模型与

用 D 训练出的模型相比可能就会有较大的误差，从而降低了评估结果的保真性（fidelity）。因此，常见的做法是：将大约 2/3~4/5 的样本用于训练，剩余样本作为测试

2.4.2 交叉验证

交叉验证（Cross Validation）也是一种模型验证技术。简单来说就是重复使用数据。除去测试集，把剩余数据进行划分，组合成多组不同的训练集和验证集，某次在训练集中出现的样本下次可能成为验证集中的样本，这就是所谓的“交叉”。最后用各次验证误差的平均值作为模型最终的验证误差。

（1）留一法（Leave One Out Cross Validation，LOOCV）

假设数据集一共有 m 个样本，依次从数据集中选出 1 个样本作为验证集，其余 m-1 个样本作为训练集，这样进行 m 次单独的模型训练和验证，最后将 m 次验证结果取平均值，作为此模型的验证误差。（注：这里说的数据集都是指已经把测试集划分出去的数据集，下同）

留一法的优点是结果近似无偏，这是因为几乎所有的样本都用于模型的拟合。缺点是计算量大。假如 m=1000，那么就需要训练 1000 个模型，计算 1000 次验证误差。因此，当数据集很大时，计算量是巨大的，很耗费时间。除非数据特别少，一般在实际运用中我们不太用留一法。

（2）K 折交叉验证（K-Fold Cross Validation）

把数据集分成 K 份，每个子集互不相交且大小相同，依次从 K 份中选出 1 份作为验证集，其余 K-1 份作为训练集，这样进行 K 次单独的模型训练和验证，最后将 K 次验证结果取平均值，作为此模型的验证误差，如下图 2-2 为 5 折交叉验证的示例图。当 K=m 时，就变为留一法。可见留一法是 K 折交叉验证的特例。

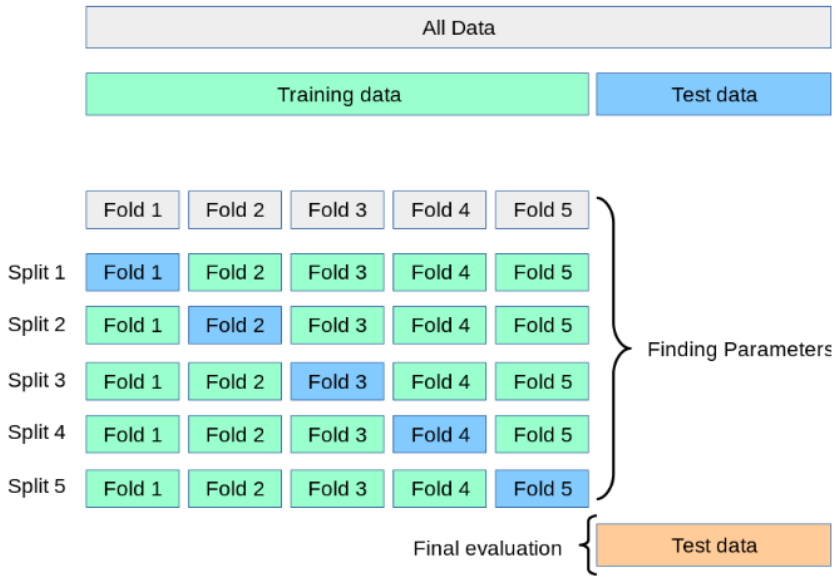

图 2-2 5 折交叉验证示例图

（3）多次 K 折交叉验证（Repeated K-Fold Cross Validation）

每次用不同的划分方式划分数据集，每次划分完后的其他步骤和 K 折交叉验证一样。例如：10 次 10 折交叉验证，即每次进行 10 次模型训练和验证，这样一共做 10 次，也就是总共做 100 次模型训练和验证，最后将结果平均。这样做的目的是让结果更精确一些。（研究发现，重复 K 折交叉验证可以提高模型评估的精确度，同时保持较小的偏差。）

（4）蒙特卡洛交叉验证（Monte Carlo Cross Validation）

即将留出法(holdout)进行多次。每次将数据集随机划分为训练集和验证集，这样进行多次单独的模型训练和验证，最后将这些验证结果取平均值，作为此模型的验证误差。与单次验证（holdout）相比，这种方法可以更好地衡量模型的性能。与 K 折交叉验证相比，这种方法能够更好地控制模型训练和验证的次数，以及训练集和验证集的比例。缺点是有些观测值可能从未被选入验证子样本，而有些观测值可能不止一次被选中。（偏差大，方差小）

在数据较少的情况下，使用 K 折交叉验证来对模型进行评估是一个不错的选择。如果数据特别少，那么可以考虑用留一法。当数据较多时，使用留出法则非常合适。如果我们需要更精确一些的结果，则可以使用蒙特卡洛交叉验证。

2.5 模型评价指标——混淆矩阵

混淆矩阵(confusion matrix):是展示分类学习算法的一种性能矩阵(方阵)，包括分类器预测结果真正(true positive)、真负(true negative)、假正(false positive)、假负(false negative)的数量，如下图 2-3

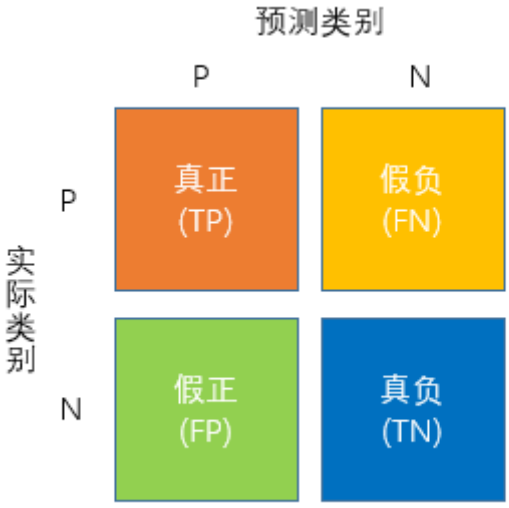

图 2-3 混淆矩阵

真正（TP）：实际是正类别，预测也是正类别；
假负（FN）：实际是正类别，预测成了负类别；
真负（TN）：实际是负类别，预测也是负类别；
假正（FP）：实际是负类别，预测成了正类别。

预测性分类模型，肯定是希望越准越好。那么，对应到混淆矩阵中，那肯定是希望 TP 与 TN 的数量大，而 FP 与 FN 的数量小。所以当我们得到了模型的混淆矩阵后，就需要去看有多少观测值在第二、四象限对应的位置，这里的数值越多越好；反之，在第一、三象限对应位置出现的观测值肯定是越少越好。

2.5.1 二级指标

（1）预测误差(error,ERR)和准确率(accuracy,ACC)

两者都表示了误分类样本数量的相关信息，其中 ERR=1-ACC。预测误差，为预测错误样本的数量与所有样本数量的比值。准确率，为预测正确样本的数量与所有样本数量的比值。计算公式如下：

$$\begin{aligned} ACC &= \frac{TN+TP}{TP+FN+TN+FP} \\ ERR &= \frac{FN+FP}{TP+FN+TN+FP} \end{aligned} \tag{2-28}$$

（2）真正率(TPR)与假正率(FPR)

对于类标数量不均衡的分类问题来说，真正率(TPR)与假正率(FPR)也是非常有用的性能指标。真正率表示预测与实际都为正类别的样本数量与实际正样本数量的比值，假正率表示预测为正类别实际为负类别的样本数量与实际负类别的样本数量的比值。计算公式如下:

$$\begin{aligned} TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned} \tag{2-29}$$

（3）精准率(precision,PRE)和召回率(recall,REC)

精准率是在模型预测是正的所有结果中，模型预测对的比重，召回率和真正率的含义相同，计算公式如下：

$$\begin{aligned} PRE &= \frac{TP}{TP+FP} \\ REC &= \frac{TP}{TP+FN} \end{aligned} \tag{2-30}$$

2.5.2 三级指标

这个指标叫做 F1 Score，它的计算公式是：

$$F1Score = \frac{2PRE * REC}{PRE + REC} \tag{2-31}$$

F1-Score 指标综合了 Precision 与 Recall 的产出的结果。F1-Score 的取值范围从 0 到 1 的，1 代表模型的输出最好，0 代表模型的输出结果最差。

2.5.3 ROC 曲线

受试者工作特征曲线(receiver operator characteristic,ROC)是基于模型的假正率和真正率等性能指标进行分类模型选择的有用工具，假正率和真正率可以移动分类器的分类阈值来计算。ROC 曲线的对角线表示的是随机猜测，比如说某件事情发生或者不发生，那么我们随机猜中的概率为 0.5。如果 ROC 曲线在对角线下，就表示分类器的性能比随机猜测还差。最好的分类器，其真正率是 1，假正率是 0，对应的 ROC 曲线是一条横轴为 0 与纵轴为 1 组成的折线。ROC 曲线下的区域(area under the curve,AUC),用来表示分类模型的性能。

3 实验结果

实验中分别使用三种方法对给定的训练集和验证集进行了分类，记录了各实验的准确率、真正率和假正率，如下表 3.1 所示。

表 3.1 给定训练集和验证集分类结果

分类器	准确率	真正率	假正率
简单分类器	0.9300	1.0000	0.1228
线性支持向量机	0.9100	0.8727	0.0444
非线性支持向量机	0.9960	0.9256	0.0000

然后再使用 5 折交叉验证法对三种分类方法进行了验证，记录了各实验的准确率、真正率和假正率，如下表 3.2 所示。

表 3.2 5 折交叉验证分类结果

分类器	准确率	真正率	假正率
简单分类器	0.9150	0.9889	0.1382
线性支持向量机	1.0000	1.0000	0.0000
非线性支持向量机	0.9650	0.9359	0.0000

从以上两张表中可以得到，在给定训练集和验证集的测试中，线性支持向量机的表现最差，非线性支持向量机的表现最优；而在 5 折交叉验证测试中，线性支持向量机和非线性支持向量机的表现都优于简单分类器，且线性支持向量机更优，此外注意到其准确率达到了 1.0000，其在交叉验证中所有五次测试均达到了

1.0000 的准确率。这似乎表明，在训练样本较多的情况下，线性支持向量机才能获得更好的效果。

4 心得体会

由于脑电信号是一个随机信号，具有非平稳、幅值低（μV 级别）的特性，且通常夹杂着肌电信号、眼电信号、环境干扰、工频干扰等噪声，从而其预处理和特征提取的步骤相对复杂，故为了实践模式识别的 SVM 算法选择了已经进行相关处理后得到的特征数据。针对 SVM 算法，Matlab 中有类似 fitrsvm 的内置函数，还有 Libsvm 工具箱可供下载，为了加深对课程所学知识的理解，后决定自己编写 SVM 算法，当然也是在参考了 MATLAB 中 fitrsvm 的 doc 文件的基础上。其实，整个实验过程只是完成了最基本的 SVM 的编写，没有对线性支持向量机中的超参数 C 和非线性支持向量机的核函数进行选择优化，故实验得出的结果也有一定的局限性。这次大作业除了编写了 SVM，还对分类模型的验证方法和评价标准进行了拓展。对于诸如 SVM、K 近邻、线性判别分析 LDA 等分类算法，以前只是在组会上听师兄师姐介绍过，一些推导我都是浅尝辄止，基本都是没听懂，在上了模式识别这门课之后，特别是自己动手去推导和编程后，发现这些算法其实并没有我之前想象的那么困难（一些基于概率论的分类算法还是没搞懂，比如朴素贝叶斯、三个估计、EM 算法等，概率论这方面知识在后期研究中需要补上）。总之，这次大作业是我对机器学习算法的一次尝试，对我之后关于脑电信号的识别研究很有帮助。

5 参考资料

李航. 统计学习方法[M]. 北京: 清华大学出版社, 2012.

https://zhuanlan.zhihu.com/p/87173581

https://blog.csdn.net/BigData_DK/article/details/83624387

https://www.cnblogs.com/HuZihu/p/9368362.html

https://blog.csdn.net/sinat_29957455/article/details/79718804