异常 & 新奇检测：一类支持向量机

Part 1：理论

支持向量机（ Support Vector Machine，以下简称 **SVM** ）相信接触过机器学习和深度学习的读者一定不陌生了，是按监督学习（ Supervised Learning ）方式对数据进行二分类的广义线性分类器。笔者在这系列文章要介绍的是SVM的延申：**一类支持向量机**（ One Class SVM，简称 **OCSVM** ）。OCSVM的主要用途就是新奇检测（ Novelty Detection ）和异常检测（ Outlier Detection ）。

系列文章分为两部分：

- **Part 1** 主要围绕着OCSVM的一些理论进行讲解。充分理解理论才能将它放入你的武器库，做到在合适的场合使用。
- **Part 2** 会用一些玩具例子和实际例子演示OCSVM的具体实施和效果

Part 1 分为两部分：

1. 一类支持向量机 (One Class SVM)

1.1 一类问题

我们先来讨论什么是一类问题。举一个非常简单易懂的例子。

- *二分类问题*：你从小一直吃苹果和香蕉，现在有一盘水果，里有分补较为均匀的苹果和香蕉，要求对其中每一个水果进行判断：它是苹果还是香蕉。
- *一类问题*：　你从小只吃过各种各样的苹果这一种水果。现在有一盘水果摆在你面前，要求对其中每一个水果进行判断：它是苹果还是“不是苹果”。

注意了，在第二个场景中，如果它不是苹果，我们并不在意它到底属于哪一种水果，我们只在意它 **是不是苹果**！其实这第二个场景也是一类SVM的典型运用场景：在一个分类问题中，只有一种类型的样本A，或有两种类型样本，但其中一类型样本数目远少于A类型样本数目。我们的目标就是判断：属于/不属于A。基于这种场景，我们可以引出两种概念：

1. **新奇检测**：训练集中只有一种类型A的样本，但测试集中的每一个数据属不属于A。
2. **异常检测**：训练集中有一种类型A的样本和其他数量远小于A的样本，要求检测出不属于A的样本。

用一句话来概括：**一类学习** 是指训练过程中，只有一类数据，通过算法寻求出可以代表这部分数据的模型，从而在检测过程中，输出数据样本是否属于该类别。

为什么不能用二分类算法来解决这种一类问题?

在新奇检测和异常检测中，我们最忌讳的是**假正例**（False Positive, FP)，即我们将不属于A的样本判断成属于A了。这在实际应用中会造成非常大的隐患。而用二分类算法来解决一类问题就会造成非常高的FP值，其原因在于在训练的过程中算法会对数量一边倒的样本产生偏倚影响，固在预测的过程中也会频频失误。

这篇文章中，作者具体分析了这种现象。作者尝试使用逻辑回归算法对一个极具倾向性的数据集进行分类。在不改变任何权重的前提下，虽然真正例（True Positive，TP）达到了100%，但FP值达到了惊人的72%。在对数据集的权重进行调整后，我们仍会有40%的FP值。OCSVM非常完美地解决了这个问题，将FP值降到了3%，同时还保持着100%的TP值!

1.2 回顾SVM

假设我们有一组数据集 $\Omega = \{(x_i, y_i) : i = 1, ..n\}$； $y_i = \{-1, 1\}$。 SVM学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的**分离超平面** $w \cdot x + b = 0$。支持向量就是最大间隔边界上的样本点。同时，引入“软间隔”的概念防止模型过适应。

核函数:
SVM具有一个非常友好的性质。如果数据在目前维度 I 中线性不可分，我们可以将数据点 x 通过一个非线性函数 $\phi(x)$ 映射到新的特征空间 F，在这个更高的维度中达成线性可分。在 F 中找到分离超平面 $w \cdot \phi(x) + b = 0$ 后再将其映射回 I 即可。而在实际运用中，**并不需要直接指名某种映射**。SVM真正在意的是在空间 F 中所有数据的两两距离，即数据点之间的内积 $\phi(x) \cdot \phi(x')$。所以我们用一个**核函数** $K(x, x') = \phi(x)\dot{\phi}(x')$ 取代内积。比较典型的核函数是高斯函数:

$$K(x, x') = exp(-\frac{\|x - x'\|^2}{2\sigma^2})$$

SVM的本质就是求解以下二次规划：

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}\|w\|^2 + C\sum_i \xi_i, \\ s.t. \quad & y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, 2, ...n \end{aligned}$$

其中 C 是一个惩罚参数，用来惩罚松弛变量 ξ_i 。通过求解对偶问题，α 为拉格朗日算子,决策函数就可以表示为: $f(x) = sign(\sum_i \alpha_i^* y_i K(x, x_i) + b^*)$。

以上就是基本SVM的内容，如果不理解具体推导过程的同学可以参考这篇知乎文章，讲解得十分详细。其实，后文提到的两种OCSVM的求解方式和普通SVM大同小异，如果明白SVM的求解原理，那理解OCSVM就并不会非常困难。我们接下来就来看两种最为泛用的OCSVM。

1.2 Schölkopf 的 OCSVM

Support Vector Method For Novelty Detection by Schölkopf et al. 介绍了这种被称作 ν - OCSVM的模型。其核心思想是在特种空间 F 中将所有数据点和**原点**分开，并最大化原点和超平面之间的距离。将该超平面映射回原特征空间 I 中，会形成一个将所有数据点进行包裹的区域，捕捉原始数据的密度分布。如果一个数据点落在这片区域内则决策函数返回 $+1$，反之返回 -1。

假设该超平面的表达式为 $w \cdot \phi(x) - \rho = 0$ ，目标是在分类基础的情况下最大化原点和超平面之间的距离：

$$\begin{aligned} \max_{w \in F, \rho} \quad & \frac{|\rho|}{\|w\|^2} \\ s.t. \quad & w \cdot \phi(x_i) \geq \rho, \quad i = 1, ..n \end{aligned}$$

化简，加入松弛变量 ξ_i得（具体推导过程可以参考这篇博客）：

$$\begin{aligned} \min_{w, \rho, \xi_i} \quad & \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho \\ s.t. \quad & w \cdot \phi(x_i) \geq \rho - \xi_i, \quad i = 1, ..n \end{aligned}$$

这就是 Schölkopf 的 OCSVM，与SVM相比最大的不同之处就是 ν 参数。源论文中，定义 $\nu \in (0, 1)$ ，而 n 是数据量，所以 ν 表示数据集的一个比例。同时，$\frac{1}{\nu n}$ 作为松弛变量和的常数，影响超平面外数据点的数量。

- **观察1**：当 ν 趋向于0时，松弛变量的惩罚趋向于无穷大，ν - OCSVM 就会变成“硬间隔”问题。并且，如果所有数据点和原点之间在特征空间 F 中线性可分，可以找出最大间隔超平面唯一解，所有数据点都在超平面的一侧（皆非异常值）。

我们来看看 ν 在对偶问题中扮演了什么养的角色。ν - OCSVM的对偶问题写作：

$$\min_{\alpha} \quad \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j) \tag{1}$$

$$s.t. \quad \sum_i \alpha_i = 1, \tag{2}$$

$$0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad i = 1, 2, ...n \tag{3}$$

- **观察2**：当 ν 趋向于 1 时，拉格朗日算子 α_i 的上限就是 $1/n$，结合（2）得出对偶问题的唯一解： $\alpha_i = 1/n$，即每一个 α_i 都取上限值 。此时，所有的数据点都是支持向量。

从以上两点观测，我们发现 ν 在整个OCSVM中是一个校准参数，他用来平衡支持向量和异常值之间的数量。论文中给出了以下结论：

1. ν 为异常值的分数设置了一个上限（训练数据集里面被认为是异常的）
2. ν 是训练数据集里面做为支持向量的样例数量的下届

这是两个非常重要的性质，也是在实际运用中调整 ν 参数的依据。没错，在 ν - OCSVM 模型中，ν 是一个超参数。

最后，我们给出 ν - OCSVM的决策函数：

$$f(x) = sign(\sum_i \alpha_i K(x_i, x_j) - \rho)$$

注意：在决策函数中，是没有 y_i 项的，也就是我们不需要知道标签来进行学习。

1.3 SVDD

刚才我们看到 ν - OCSVM和SVM非常类似，都是寻找一个超平面。而Support Vector Data Description by Tax and Duin的方法却是寻找一个超球面，十分有趣。这个算法试图在数据的特征空间 I 中寻找一个超球面边界，包裹住所有的数据点，同时，也要最小化这个球体的体积，减少异常点被容纳进球体的可能性。

将超球体的圆心写作 $\mathbf{a}$, 是所有支持向量的线性组合，超球体的半径写作 R。引入松弛变量 ξ_i，我们得出以下优化问题：

$$\begin{aligned} \min_{R,\mathbf{a}} \quad & R^2 + C\sum_i \xi_i \\ s.t. \quad & \|x_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad i = 1, ..., n \\ & \xi_i \geq 0 \quad i = 1, ...n \end{aligned}$$

SVDD的决策函数便是两个点之间的距离是否小于超球体的半径，十分易懂。唯一个超参数 C 和SVM中的 C 具有相同的含义。值得一提的是，SVDD也可以用核函数来映射到高纬度的特征空间。这样一来在 I 中的分类边界就不再是个超球面，而是更加复杂且严密的的边界。

2. 调参 & Cross Validation (CV) ?

理解了OCSVM的基本原理之后，接下来就来聊聊调参。笔者是直接用sklearn包的，目前sklearn还没有实装SVDD，所以以下讨论皆基于 ν - OCSVM 。

在开始之前，笔者想要强调一个问题：**异常检测和新奇检测能不能用CV来调参?**

在监督学习的场景中，通常会被要求对算法模型的超参数做一个Cross Validation。这已经是现在训练过程中非常相信大家对这个词汇不会很陌生。用分类场景举例，由于训练数据集中每个数据的分类标签是已知的，可以帮助我们轻松估算出模型的分类精准性。又因为我们可以定义模型的精准性，我们才能通过Cross Validation选取超参数的最优解。

然而，异常和新奇检测都是属于**非监督学习**的范畴，即我们并不知道训练数据集中的标签。多年以来，非监督学习的CV一直是机器学习研究领域的一个讨论话题。非监督学习的CV不像监督学习的那些模型有泛用的打分标准，例如分类任务里的准确度、NLL（负相似对数）；回归任务里的MAE（平均绝对误差）、Euclidean距离函数等。非监督学习调参的论文讨论的方法往往与任务本身息息相关。也就是说，**不同的非监督学习任务可能有着完全不同的调参方法**。

那我们如何对 ν - OCSVM 进行调参?

2.1 核函数选择

`sklearn.svm.ocsvm(*)` 提供了以下几种核函数：

1. 径向基核函数（Radial Basis Function）/ 高斯核（Gaussian Kernel）

$$K(x, x') = exp(-\frac{\|x - x'\|^2}{2\sigma^2})$$

系统默认的核函数，也是应用范围最广的核函数。不知道用什么的时候优先使用它，无论大小样本都有非常好的性能。其优点是**参数较少**，且对噪声有着较好的**抗干扰能力**。

2. 线性核（Linear Kernel）

$$K(x, x') = x \cdot x'$$

主要用于**线性可分**的情况，参数少，速度快，效果显著。可以看出特征空间 F 的维度和原数据维度是一样的。也可以先从线性核入手，如果效果不好再另辟蹊径。

3. 多项式核（Polynomial Kernal）

$$K(x, x') = ((x \cdot x') + 1)^d$$

多项式核函数可以实现将低维的输入空间映射到高纬的特征空间。适合于正交归一化数据。参数 d 越大，映射的维度越高，计算量就会越大。

但是多项式核函数的**参数多**，当多项式的阶数d比较高的时候，由于学习复杂性也会过高，易出现**过拟合**现象，核矩阵的元素值将趋于无穷大或者无穷小，计算复杂度会大到无法计算。

2.2 超参数 ν

回顾一下 ν 的性质：

1. ν 为异常值的分数设置了一个上限（训练数据集里面被认为是异常的）
2. ν 是训练数据集里面做为支持向量的样例数量的下届

可以说 ν 是对整个模型**影响最大**的参数了，直接决定了决策边界的样子。和SVM中的 C 类似， 都是为了在**模型准确率**与**模型复杂度**之间取得一个平衡。

我们可以结合以上两条和一些实际情况需要来选取 ν，或是根据以下这个原则：$\nu \approx$ **异常值占训练集里的比例**。这个值通常会给我们比较好的结果，在新奇检测中可以有效地将FP值控制得很低，且能保证 $TN \leq \nu$ (性质1)。

可问题也出现了：在非监督学习的情景下我们如何得知异常值的比例？笔者这里能给出的答案就是根据实际业务场景进行估算了。我们在阅读相关文献的时候经常会看到作者把OCSVM说成监督学习，原因可能就是来源于此。

2.4 超参数 gamma

gamma是属于高斯核函数的参数：$\gamma = -1/2\sigma^2$。scikit-learn中默认值是 $1/(n * var(X))$。这部分和SVM中的gamma是一样的，引用其他作者的博客里的一段话：

主要定义了单个样本对整个分类超平面的影响，当gamma比较小时，单个样本对整个分类超平面的影响比较小，不容易被选择为支持向量；反之，当gamma比较大时，单个样本对整个分类超平面的影响比较大，更容易被选择为支持向量，或者说整个模型的支持向量也会多。

gamma的调参要和 ν 一起进行，是一个相互平衡的过程。

2.3 其他调参方法

在这里给读者们分享一篇论文 Hyperparameter selection of one-class support vector machine by self-adaptive data shifting。主要说明了 ν - OCSVM 如何不用CV进行调参。文中用了很多数据转换的方法生成数量可控的异常值和正常值。和其他调参方法不同的是，它不会形成额外的超参数，是一个完全基于训练集本身的自动化步骤。同时，它也不限于高斯核函数，泛用性较广。

以后笔者可能会专门出一篇文章讲述这种方法。

Reference

1. 外国友人博客1

2. 外国友人博客2
3. 知乎：gamma解释
4. Hyperparameter selection of one-class support vector machine by self-adaptive data shifting
5. Support Vector Data Description by Tax and Duin
6. OCSVM推导过程
7. Support Vector Method For Novelty Detection by Schölkopf et al.
8. SVM推导过程