1 对论文中的核函数做出自己的理解

1.1 引言

在 SVM 中，解决线性不可分问题的主要手段是使用合适的映射将特征从低维向量空间映射至更高维空间，以转化为更高维空间的线性可分问题。但更高的维度虽带来了更好的线性可分的概率，但也随之引发了计算复杂度的升高，尤其是一些映射会将特征向量映射至无穷维空间中,这会直接导致问题不可解。而我们在并不能直接求解高维特征空间的内积的情况下，则需要用到核函数的技巧。

核函数解决线性不可分问题，除了能将低维向量空间映射至更高维空间，更重要的是还实现了简化计算的作用。

1.2 相关先修知识

在精读论文的过程中，我发现 SVM 和核函数的概念都被反复提及，因此，欲更深入地掌握核函数的知识，还需要对 SVM 的知识进行全面地了解。

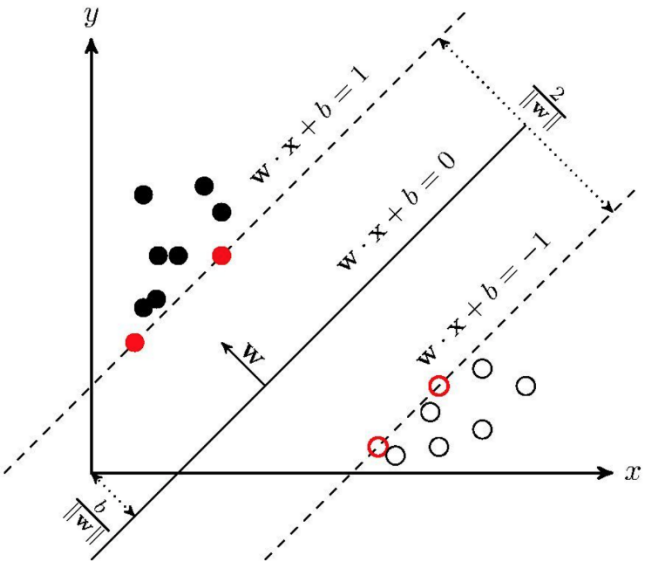
1.2.1 SVM（支持向量机）

（1）SVM 概念

SVM 是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，而间隔最大这个要求使它区别于感知机，因为感知机有无数个，间隔最大只有一个。

SVM 试图寻找一个最优的决策边界，即距离两个类别最近的样本最远。当训练数据不可分时，可以通过核函数学习非线性支持向量机。

SVM 的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的的学习算法就是求解凸二次规划的最优化算法。



（2）线性算法原理

已知 SVM 优化的主问题为

$$\min_{w} \frac{1}{2} \| w \|^2$$

$$\text{s.t.} \quad g_i(w,b) = 1 - y_i(w^T x_i + b) \le 0, i = 1,2,\dots,n$$

首先构造拉格朗日函数

$$\min_{w,b} \max_{\lambda} L(w,b,\lambda) = \frac{1}{2} \| w \|^2 + \sum_{i=1}^{n} \lambda_i [1 - y_i(w^T x_i + b)]$$

$$s.t. \lambda_i \ge 0$$

再利用强对偶性可转化为

$$\max_{\lambda} \min_{w,b} L(w,b,\lambda)$$

现对参数w与b求偏导

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \lambda_i x_i y_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0$$

得到

$$\sum_{i=1}^{n} \lambda_i x_i y_i = w$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

最终可得到

$$\min_{w,b} L(w,b,\lambda) = \sum_{j=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j (x_i \cdot x_j)$$

其次利用 SMO（序列最小优化算法）求解上式，可得

$$\max_{\lambda} \left[\sum_{j=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j (x_i \cdot x_j)\right]$$

$$\text{s.t.} \sum_{i=1}^{n} \lambda_i y_i = 0 \lambda_i \geq 0$$

计算w与b，构造超平面，最终得到分类决策函数

$$f(x) = \text{sign}\,(w^T x + b)$$

$$\text{sign}\,(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

（2）非线性算法原理：核函数

对于线性不可分的数据集，SVM 通常会选择一个核函数处理。处理思路一般是将二维线性不可分样本映射到高维空间中，让样本点在高维空间线性可分。

用x表示输入空间的样本点，首先将x映射到新的特征空间，相应地得到分割超平面函数

$$f(x) = w\phi(x) + b$$

后续的计算则根据新的向量进行对偶计算。

1.3 核函数详细介绍

1.3.1 核函数定义

设$\mathcal{X}$是输入空间（欧式空间R^n的子集或离散集合），又设$\mathcal{H}$为特征空间（希尔伯特空间），如果存在一个从$\mathcal{X}$到$\mathcal{H}$的映射

$$\phi(x): \mathcal{X} \to \mathcal{H}$$

使得对所有$x, z \in \mathcal{X}$，函数$K(x,z)$满足条件

$$K(x,z) = \phi(x) \cdot \phi(z)$$

则称$K(x,z)$为核函数，$\phi(x)$为映射函数，式中$\phi(x) \cdot \phi(z)$为$\phi(x)$和$\phi(z)$的内积。

1.3.2 核函数原理

SVM 中将一个二次规划问题经过拉格朗日乘子法、对偶问题转化等技巧，将优化变量 ω 的确定转化为高维空间向量内积的求解问题，由此可以通过核函数解决线性不可分问题。

核函数在解决线性不可分问题的时候，采取的方式是：使用低维特征空间上的计算来避免在高维特征空间中向量内积的巨大计算量；也就是说此时 SVM 模型

可以应用在高维特征空间中数据可线性分割的优点，同时又避免了引入这个高维特征空间巨大的内积计算量。

核函数的作用是简化映射到高维空间后的内积运算，直接通过低维空间的内积计算得到高维空间的内积，是一种简化计算的技巧。

本质：核函数是一个低维的计算结果，并没有采用低维到高维的映射。只不过核函数低维运算的结果等价于映射到高维时向量点积的值。低维映射到高维的操作实际上是由$\phi(x)$映射函数完成的，并不是核函数。

1.3.3 常见的核函数及选择

lasso 回归相比于岭回归，会比较极端。它不仅可以解决过拟合问题，而且可以在参

（1）线性核函数（Linear kernel function）

$$k(x,y)=x^Ty+c$$

线性核函数是多项式核函数的特例，优点是简洁，缺点是对线性不可分数据集没有解决办法。主要用于线性可分的情况，特征空间到输入空间的维度是相同的，其参数少速度快，对于线性可分数据，其分类效果很理想，因此通常首先尝试用线性核函数来做分类。

（2）多项式核函数（polynomial kernel function）

$$K(x,z)=(x\cdot z+1)^p$$

对应的支持向量机是一个p次项多项式分类器。在此情形下，分类决策函数成为

$$f(x)=\text{sign}\left(\sum_{i=1}^{N_s}a_i^*y_i(x_i\cdot x+1)^p+b^*\right)$$

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，但是多项式核函数的参数多，当多项式的阶数比较高的时候，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。

（3）高斯核函数（Gaussian kernel function）

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

对应的支持向量机是高斯径向基函数分类器。在此情形下，分类决策函数成为

$$f(x)=\text{sign}\left(\sum_{i=1}^{N_s}a_i^*y_i\exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right)+b^*\right)$$

高斯核函数也称径向基核函数，是一种局部性强的核函数，该函数的形状为钟形曲线，参数σ控制曲线的宽度。可以把输入特征向量扩展到无限维度的空间里。高斯核函数计算出来的值永远在 0 到 1 之间。其可以将一个样本映射到一个更高维的空间内，该核函数是应用最广的一个，无论大样本还是小样本都有比较好的性能，而且其相对于多项式核函数参数要少，因此大多数情况下在不知道用什么核函数的时候，优先使用高斯核函数。

（4）在核函数的选择上

如果特征维数很高，往往线性可分（SVM 解决非线性分类问题的思路就是将样本映射到更高维的特征空间中），可以采用线性核函数。

如果样本数量很多，由于求解最优化问题的时候，目标函数涉及两两样本计算内积，使用高斯核明显计算量会大于线性核，所以手动添加一些特征，使得线性可分，然后使用线性核函数。

如果不满足上述两点，即特征维数少，样本数量正常，可以使用高斯核函数。

1.4 难点及心路历程

1.4.1 论文概括

《Advanced support vector machinesand kernel methods》在我的理解是一篇综述性论文，提出了综合泛化改进、模型选择、超参数调整和先验知识的先进支持向量机和核方法。提出了基本方法，提供了一个共同的基础，包括原始/对偶问题和数值优化问题。并且说明了 SVM 在各细分领域得到了广泛的应用。

1.4.2 心路历程

在通读论文以后，我对其中的公式定理，尤其是 SVM 的各个概念非常模糊，因此去网上参考了许多资料，其中最有用的还是李航的《统计学习方法》，结合 b 站 up 主讲解以后，我对 SVM 基本掌握了。核函数是一个非常神奇的方法，尤其是在解决非线性分类问题上，但核函数其实是一个更加通用性的概念，不仅仅是用于 SVM 上。

1.4.3 难点

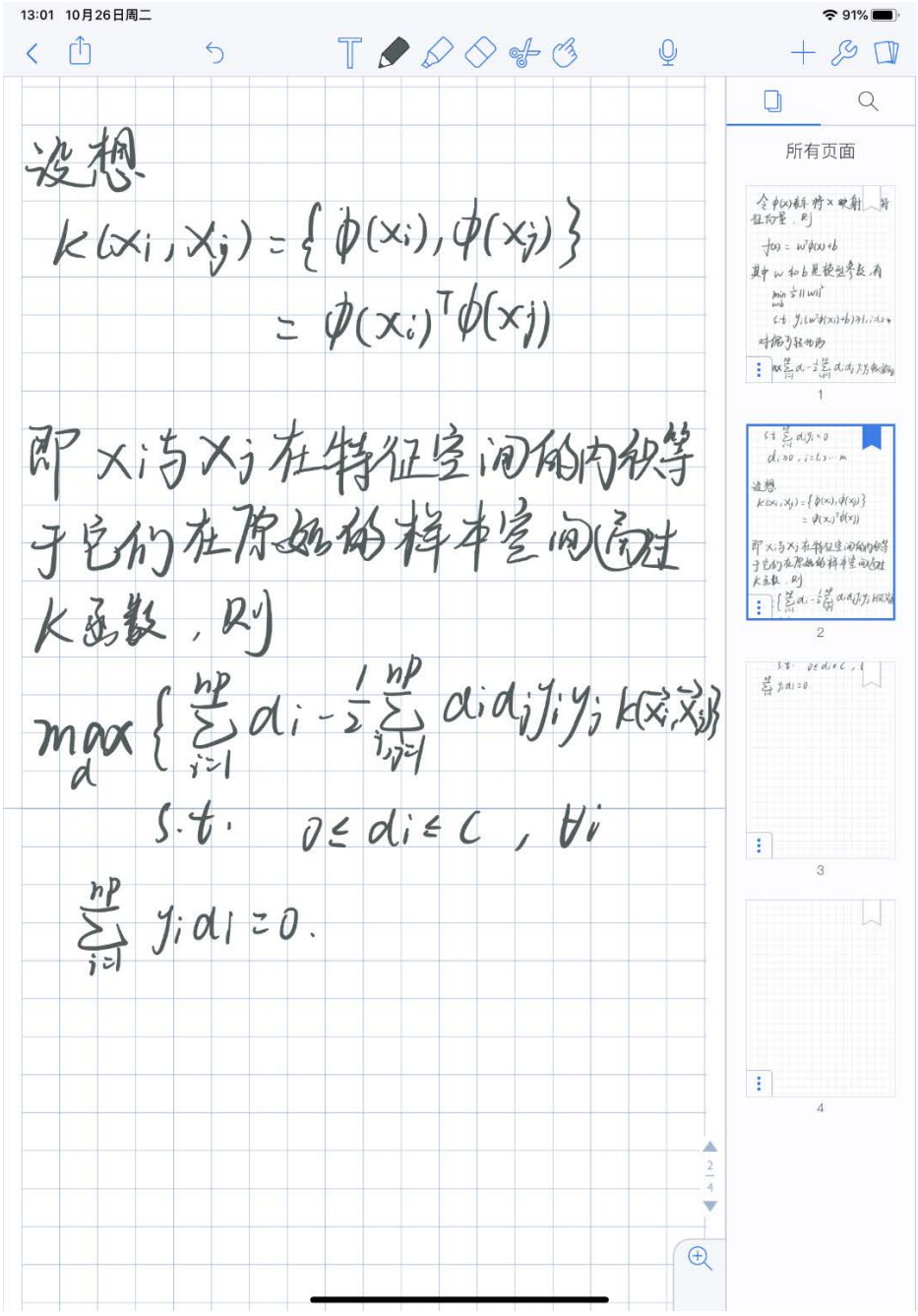
目前对核函数的了解还不算彻底，例如在“核函数为什么能简化计算上”以及 SMO 算法的掌握上还存在一些没有明白的点，还需要公式推导或是通过编程的方式才能完全掌握。

2 对论文中的重要过程的推导

对于原论文中

$$\begin{aligned} \max_{\alpha} \quad & \left\{\sum_{i=1}^{n_p}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n_p}\alpha_i\alpha_j y_i y_j K(\vec{x}_i,\vec{x}_j)\right\}, \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant c, \forall i \\ & \sum_{i=1}^{n_p} y_i\alpha_i = 0 \end{aligned}$$

公式的推导如下：



令 $\phi(x)$ 表示将 x 映射后的特征向量，则

$$f(x) = w^T\phi(x) + b$$

其中 w 和 b 是模型参数，有

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

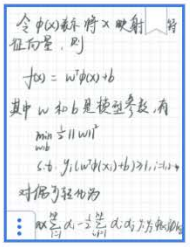
$$s.t.\ y_i(w^T\phi(x_i)+b) \geq 1, i=1,2\cdots m$$

对偶为转化为

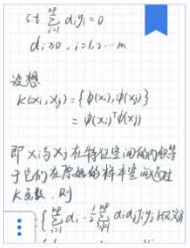
$$\max \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j \phi(x_i)\phi(x_j)$$

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

$$\alpha_i \geq 0, i=1,2\cdots m.$$



1



2

3

4

3 相关进一步引用该论文并作出了有效改进的追踪（两个）

3.1 SVM 的缺陷

虽然 SVM 具备统计学习理论基础，以及优良的泛化能力，并且在多个领域已经得到了广泛的应用，但同时也存在一些缺陷。例如：计算量大、速度慢；参数选择经验性强；不能很好地解决多分类问题等。

3.2 FCSVM

3.2.1 FCSVM 介绍

针对训练速度慢的问题，刘向东等于 2004 年提出一种快速的支持向量机分类算法 FCSVM。

其对支持向量集采用变换的方式,用少量的支持向量代替全部支持向量进行分类计算,在保证不损失分类精度的前提下使得分类速度有较大提高。

在一定程度上克服了传统的支持向量机分类速度较慢的缺点、尤其在训练集规模庞大、支持向量数量较多的情况下,采用该算法能够较大幅度地减小计算复杂度,提高分类速度。

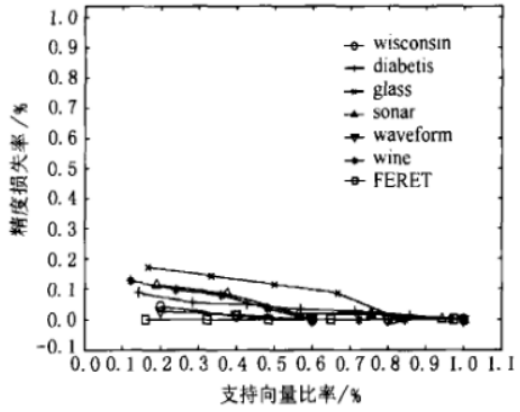
3.2.2 FCSVM 算法思想

对于 SVM 分类器,当支持向量数量较多时计算量较大,其分类速度必然较慢,为此减少分类函数中支持向量的数量是提高分类速度的关键,但支持向量集是决定最优超平面的关键子集，随意减少支持向量会影响超平面的形式，从而影响分类精度。FCSVM 的核心思想是减少分类函数中的支持向量数量，同时又不影响分类精度，

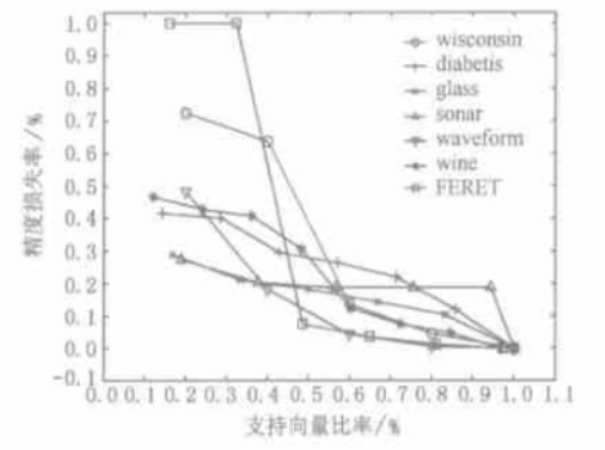
FCSVM 方法是通过优化的方式适当变换特征空间中的内积矩阵，并进一步变换分类函数的形式，以减少分类函数中的支持向量数量，提高分类速度，同时保留其他支持向量的信息，使全部支持向量的信息得以保留在分类函数中，因此分类精度并没有损失,从而实现了在不损失分类精度的前提下通过减少支持向量数量而提高分类速度。

3.2.3 FCSVM 与标准 SVM 的对比

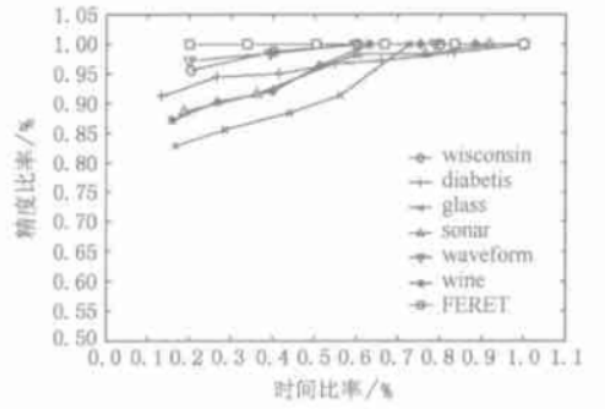
根据下图，观察 FCSVM 支持向量比率与精读损失比率的关系。随着使用的支持向量比例的增大，其精度快速逼近最佳值，而精度损失较小且衰减很快，另外在使用较少支持向量时就能够提前达到最佳值,其原因在于未使用的支持向量的信息没有被抛弃而是融合到分类函数中了。



根据下图，观察标准 SVM 方法中支持向量比率与相应精度损失率的关系。可以看到在使用较少支持向量时其精度损失很大，只有使用高比率或全部支持向量时才能达到最佳精度值，原因就在于标准 SVM 在使用部分支持向量的同时简单抛弃了另一些支持向量，导致精度损失较大。



根据下图，观察 FCSVM 分类时间比率与精度比率之间的关系。可以看到其在使用较少分类时间的同时能够达到最佳的分类精度，分类速度可以提高 16.7%～80.0%。从另一个意义上讲，如果对分类精度的要求不是很高，那么可以适当损失很小的精度而换取速度的大幅提高。



3.3 DAGSVM

3.3.1 DAGSVM 介绍

针对 SVM 不能很好地解决多分类问题，John C. Platt 等于 1999 年提出 DAGSVM。

DAGSVM 是根据有向无环图(DAG）提出的另外一种基于 SVM 的多类别分类器，它引入了 OVO 分类器中利用每两个类别作为基础二类分类器的方法，保证了分类的准确率，而且采用了有向无环图结构，使得每次分类只需要$k-1$个分类器，大大提升了分类的效率。但是由于采用了层次结构，也保留了层次结构固有的的缺陷:误差累积，在上层的节点产生的错误会一直保留下来，因此，距离根节点越近的节点，对整个结构的分类结果影响越大。而 DAGSVM 的节点选取方式采用了随机的方式，这就使得最终的分类结果十分不稳定。

3.3.2 DAGSVM 原理

DAG-SVM 是一种以 SVM 分类器作为基础分类器，以有向无环图作为拓扑结构的组合多类别分类器，它通过从全体类别集合中不断的删除不可能的类别，得到最终的结果。

DAG-SVM 的具体分类过程，根节点 a 表示当前类别集合为全体类别{1，2，3,4}，经过分类器 1-vs-4 作用后，节点走到第二层，由于排除了一个类别 4，因此当前类别集合为{1,2,3}，经过分类器 1-vs-2 作用后，节点走到了第三层 d{1，3}，再经过分类器 1-vs-3 作用后，走到最后的叶节点类别 1，因此，类别 1 即为最终的结果。