华南理工大学

South China University of Technology

本科毕业设计（论文）说明书

基于损失函数的不平衡分类问题的研究

学　　院　计算机学院

专　　业　计算机科学与应用（双语）

学生姓名　张智卓

指导教师　陈琼

提交日期　2008 年 6 月 9 日

华 南 理 工 大 学

毕 业 设 计 （论文） 任 务 书

兹发给_____班学生_____毕业设计（论文）任务书，内容如下：

1.毕业设计（论文）题目：____基于损失函数的不平衡分类问题的研究____

2.应完成的项目：

（1） 分析各种凸损失函数在不平衡数据下的表现

（2） 实现各种损失函数的对应正则化算法

（3） 定义“不平衡不敏感”，并给出充分条件

（4） 设计出“不平衡不敏感”损失函数 ，并实现算法

（5） 推广“不平衡不敏感”到多分类和代价敏感

（6） 分析“采样不平衡”

3.参考资料以及说明：

（1）Bennett, K. P. & Parrado-Hernández, E. (2006), 'The Interplay of Optimization and Machine Learning Research'

（2）Xu, L.; Crammer, K. & Schuurmans, D. (2006), 'Robust support vector machine training via convex outlier ablation'

（3）Rosset, S.; Zhu, J. & Hastie, T., 'Margin Maximizing Loss Functions'

（4）Liu, X. & Zhou, Z. (2006), 'The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study',

（5）Japkowicz, N. (2003), 'Class imbalances: are we focusing on the right issue'

（6） Zhou, Z. & Liu, X. , 'On Multi-Class Cost-Sensitive Learning'

（7） Belkin, M.; Niyogi, P. & Sindhwani, V. (2005), 'On manifold regularization'

4.本毕业设计（论文）任务书于 年 月 日发出，应于 年 月 日前完成，然后提交毕业考试委员会进行答辩。

专业教研组（系）、研究所负责人_____审核_____年___月___日

指导教师_____签发 _____年___月___日

毕业设计（论文）评语：

（*应包括平时表现、论文质量、答辩表现等内容*）

毕业设计（论文）总评成绩：

毕业设计（论文）答辩负责人签字：

年　　月　　日

摘　要

数据不平衡一直被认为是影响分类器性能的一个重要原因，很多学者试图通过重采样，组合方法，改变评价指标等方法来研究数据不平衡问题。而本文从 Tikhonov 正则化框架下机器学习问题转化为最优化问题的角度入手，分析在数据不平衡下采用不同损失函数对优化问题的最优解的影响。本文通过对凸损失函数分三种情况讨论，深刻指出数据不平衡情况下分类器性能下降的根本原因是实施凸优化的结果。另外，本文还给出“不平衡不敏感”的定义和“不平衡不敏感”损失函数的充分条件。同时，由于“不平衡不敏感”损失函数是非凸函数，机器学习问题对应为一个非凸优化问题，本文对采用随机梯度下降方法和半定规划（SDP）方法解决这样的非凸优化问题进行分析和讨论。最后，本文还对“不平衡不敏感”损失函数在多分类情况和代价敏感情况下进行了推广并且讨论了“采样不平衡”的对应解决策略。

关键词：数据不平衡，Tikhonov 正则化，非凸优化，不平衡不敏感，损失函数，半定规划

Abstract

Data imbalance is considered as an important factor affecting the performance of classifiers. Many Meta methods like Re-sampling, classifiers ensemble, various evaluation, have been tried to handle the imbalance problem. This paper takes the machine learning problem as an optimization problem based on Tikhonov Regularization framework, and discusses the effect of different loss function on the optimized solutions in the imbalance situation. In this paper, Classified Discussion of three type convex loss function further points out that the essential cause is the convex optimization, which leads to the performance degradation of the classifiers in the imbalance situation. Moreover, the “imbalance insensitive” and “imbalance insensitive” loss function are defined in this paper with their sufficient condition. However, the “imbalance insensitive” loss function is non-convex function, which turns the machine learning problem to a non-convex optimizing problem. Hence, the further analysis on the non-convex problem solving methods like random gradient decreasing and semi-define programming approximating are given in this paper too. Finally, the paper generalizes the “imbalance insensitive” theory in the multi-class case and cost-sensitive case and makes some discussion on the issue “sampling imbalance”.

Keyword: data imbalance, Tikhonov Regularization, non-convex optimizing, imbalance insensitive, loss function, semi-define programming

目录

第一章　绪论

1.1 不平衡问题研究概述

具有不平衡类分布的数据集在许多实际应用中是很常见的，很多学者的研究实验反映不平衡数据会令大多数已知分类器的性能下降，因此在诸多实际应用中，数据不平衡成为了分类学习的困难和挑战。

Japkowicz 等人[40][41]通过实验的方法研究了数据不平衡对不同分类方法的影响。其中包括基于决策树的 C45、 BP 神经网络以及支持向量机(SVM)等。实验结果表明，相对而言 ，SVM 对数据不平衡带来的影响较不敏感。[32]指出各种重采样方法可以有助于改善不平衡的情况，而[29]进一步指出“下采样”相比“过采样”可以更好有效解决不平衡问题。[26]提出了基于专家框架下解决不平衡问题。[4][7]提出可以改进 Adaboost 算法来解决不平衡问题，而[60]通过模型集成的方法解决客户数据不平衡问题。

周志华，刘胥影等人对在数据不平衡条件下的代价敏感问题进行了大量的研究。[47]利用 C45 决策树在多个不平衡的 UCI 数据集上，对在多分类问题中数据不平衡和代价敏感的关系作了主观性的研究。[46][63][27]分别对“过采样法”，“下采样法”，“阈值移动法”，“软集成”，“硬集成”等等代价敏感方法对大量不平衡数据集进行实验，并总结不同方法的效果。周志华等人提出用代价敏感方法可以很好地处理不平衡数据[63]，并提出一个对多分类样本赋权的方法[62]。

Jo 和 Japkowicz[39]进一步比较研究了类间不平衡和小析取项对分类学习的影响。小析取项即类内不平衡，从概念学习的角度来说，它反映了同一类的若干子概念之间学习样本分布的不平衡性。小析取项就是那些所涵盖的学习样本数量偏少的子概念，它们是容易被错误学习进而影响分类器整体性能的一个重要因素。类间和类内不平衡是不平衡问题的两个不同侧面，它们可能会同时出现，均会影响分类器的性能。

郑恩辉等人[9]提出了支持向量率的概念，并指出小类的支持向量率和边界支持向量率分别依概率大于大类的支持向量率和边界支持向量率。[3][7][10][13][22]在基于支持向量机的框架下分析不平衡问题，并提出多种改进的支持向量机算法，其中[2]对基于 SVM 方法在不平衡数据的研究作了综述。而[5]提出如何利用数据不平衡环境提高基于差别矩阵的核算法效率的方法。[11][22]指出单分类问题的支持向量数据描述算法有助于解决不平衡问题。

1.2 机器学习中的最优化问题概述

机器学习的核心就是最优化规划问题，目标无疑是最小化泛化误差。Vapnik[24][56]、Smale[25]、Zhou[61]在不同的函数空间下得出泛化误差的界与假设空间大小和主观误差、样本数目等参数有关，因而最优化的目标函数通常表述为主观误差与假设复杂之间的折中。最优化规划问题有分为凸优化问题和非凸优化问题，大多数算法都是基于凸优化算法下实现，只是搜索的形式不同，例如贪心算法（决策树，主成份分析），动态规划（隐马尔科夫模型）[42]等等。当然一个算法可以采用多种形式实现，最典型的是支持向量机[57]可以二次规划，线性规划或者半定规划[27]等等方法实现。可见，凸优化在机器学习中的地方是非常重要的。另外，基于优化理论和实际需求发展出在线学习[52]，主动学习，资源有限学习，模型压缩等等机器学习的子领域。

凸优化由于理论成熟[19]，而且最优收敛可以保证，因而大多数机器学习学者都倾向于把非凸优化的问题近似表述为凸优化问题[23]。然而，人们越来越发现大多数自然界中的学习问题都是非凸的，凸优化的转换显得束缚了解决问题的思路，并且有时候掩盖了问题的本质[43]。从而，非凸优化在机器学习中的应用最近越来越受到重视，尤其是在复杂对象识别和机器视觉上面。

1.3 论文选题的意义

虽然有大量论文已经针对二分类问题或者多分类问题提出了在数据不平衡的情况下的提升算法，例如重采样方法，改变评价指标方法。但到目前为止，学术界还没有对不平衡数据对分类器性能的影响有比较确定理论解释，而多数算法也只是着眼于如何提升小类样本的重要性上面。本文的选题深刻地指出了不平衡数据导致分类器性能下降与损失函数选取的关系，进而指出分类器在数据不平衡下性能下降是由于实施凸优化的结果。本文的成果将会为不平衡问题的研究确立一个定性的研究框架，并指出这类问题的正确研究思路应该是如何实施凸优化与非凸优化之间的折中。 这无论对理论分析还是实际应用都具有重大意义，为更深入理解数据不平衡问题跨出重要的一大步。

1.4 本文的内容安排

在第二章中，我将会简单介绍本文需要的基础知识；第三章将会列举出一些常见的代理损失函数；第四章定性分析了凸损失函数可能导致数据不平衡情况下分类性能下降的原因；第五章严格定义了什么是“不平衡不敏感”损失函数，并给出了其充分条件，最后给出“不平衡不敏感”损失函数的实现算法和代价敏感版本并进行实验；第六章介绍了“不

平衡不敏感”损失函数在多分类下的推广；第七章提出“采样不平衡”的概率并指出其与传统不平衡的关系；最后一部分会对本文的内容进行总结并对没解决的问题进行讨论。

第二章　基础知识介绍

2.1 凸函数的定义

凸函数是一个定义在某个向量空间的凸子集 C（区间）上的实值函数 f。设 f 为定义在区间 I 上的函数，若对 I 上的任意两点x_1, x_2和任意的实数$\lambda \in (0,1)$，总有：

$$f(\lambda \cdot x_1 + (1-\lambda) \cdot x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

则 f 称为 I 上的凸函数。

除了定义外，我们还可以通过凸函数的上方点集为凸集或者凸函数的二次偏导数大于 0 的性质来判断。

2.2 凸优化迭代算法

常见的凸优化迭代算法有牛顿梯度下降法、共轭梯度下降法、BFGS 方法和 Quasi-Newton 方法等。这些方法的共同之处是都经过两个估计阶段：一是梯度的估计，即梯度下降的方向；二是线搜索，即沿该方向应该走的距离。这两个阶段估计的方法不同就导致不同的迭代优化算法。

假设目标函数是 *f(x)*，则第一阶段的估计重点就是其一阶偏导∇f，而第二阶段的估计重点是 f 的 Hessian 矩阵，即二阶偏导$\nabla\nabla f$。

具体的详细算法这里不给出了，需要知道的是对凸优化问题来说，只要这两个估计准确的话就可以得到最优解，甚至有时候二阶偏导未知的情况下通过其他作近似估计也可以得到最优解（但收敛速度会大大减慢）。在 matlab 有专门提供“fminunc”函数接口，只要定义出 *f(x)*，∇f和$\nabla\nabla f$（后者是非必须的），并指定迭代算法就可以自动求解。

2.3 再生核希尔伯特空间（RKHS）

希尔伯特空间就是通过内积定义的范数的完备赋范线性空间。因为有了内积的定义，所以该空间就有了几何的性质，从而一些最优化问题以及其对偶问题都有了几何上直观的意义。而 RKHS 就是通过再生核 ***K*** 定义内积的希尔伯特空间。其内积具有再生特性，即：

$$\langle K_t, f \rangle_K = f(t)$$

其中 $K_t(x) =< K_t, K_x >_K = K(x,t) = K(t,x) = K_x(t)$

可见，*K* 是一个自对称的二元算子，且要求 *K* 正定。*K* 又通常被成为核函数。常见的核函数有：

线性核： $K(\mathbf{x},\mathbf{x}') = \mathbf{x}\cdot\mathbf{x}'$

高斯核（RBF）： $K(\mathbf{x},\mathbf{x}') = \mathrm{e}^{\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma^2}}, \qquad \sigma > 0$

多项式核： $K(\mathbf{x},\mathbf{x}') = (\mathbf{x}\cdot\mathbf{x}'+1)^d, \qquad d \in \mathbb{N}$

另外，核函数 K 对应的积分算子$A_k(f) = \iint_{x,y\in X} k(x,y)f(x)f(y)dxdy$是一个自共轭算子。而自共轭的算子的特征元素是 L2 空间（平方可积空间）的一组完全正交基。假设A_k对应的特征元素是$\{\Phi_i\}$,则在 L2 空间中任一元素 f 都可以表示为$\{\Phi_i\}$的线性组合，即：

$$f(x) = \sum_i w_i \Phi_i(x)$$

注意上面的求和可以是无穷的，我们通常把$x \to [\Phi_1(x),\Phi_2(x),\dots,\Phi_N(x)]$称为特征映射，*N* 为特征元素数目（可能是无穷）。可见，任何平方可积函数为分类界面的分类问题都可以转化为特征空间中线性判别问题，即：

$$f'(z) = \sum_i w_i z_i$$

因为本文对不平衡问题的讨论对样本维数没有任何假定，只是对其分类超平面的性质进行讨论，所以本文的下面所有的章节的讨论都直接在特征空间中进行，即假定分类函数的形式：

$$f(x) = \sum_i w_i x_i = \boldsymbol{w}^T x$$

当然从上面分析知道，这样讨论对任何平方可积的分类函数都适用。

2.4 Tikhonov 正则化框架

正则化（正则化）的基本思想是恢复在假设空间$\mathcal{H}$的最小化主观误差(ERM)的适定性。而一个直接的方法是，把最小化主观误差限制在赋范泛函空间$\mathcal{H}$的一个球中进行。即 Ivanov 正则化：

$$min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i),y_i)$$

且满足 $\|f\|_{\mathcal{H}}^2 \le A$

其中 L 被称为代价函数或者**损失函数**。

通过拉朗格日乘子技术，我们可以把约束放到目标函数中去，从而得到 Tikhonov 正则化：

$$min_{f\in\mathcal{H}}\ \frac{1}{n}\sum_{i=1}^{n}L(f(x_i),y_i)+\lambda\|f\|_{\mathcal{H}}^2$$

在实际分析中，假设空间$\mathcal{H}$通常选取再生核希尔伯特空间（RKHS），由上一小节知道，目标函数可以表述为：

$$min_{f\in\mathcal{H}}\ \frac{1}{n}\sum_{i=1}^{n}L(w^Tx_i,y_i)+\lambda\|w\|^2$$

另外上式中的损失函数，通常可以表示为基于“间隔（margin）”的形式，即：

$$min_{f\in\mathcal{H}}\ \frac{1}{n}\sum_{i=1}^{n}L(y_i\cdot w^Tx_i)+\lambda\|w\|^2 \quad (1)$$

间隔定义为 $m=y_i\cdot\boldsymbol{w}^T\boldsymbol{x_i}$，而可以用间隔表示输入的损失函数称为基于间隔的损失，本文主要研究的就是这一大类损失函数。

第三章　代理损失函数

3.1 0-1 损失

在二分类问题中，最自然的损失函数就是 0-1 损失。假设样本 label 为{-1,1}，预测函数为 f，得：

$$L_{0-1}(y,f)=\begin{cases}1 & yf\le 0\\ 0 & yf>0\end{cases}$$

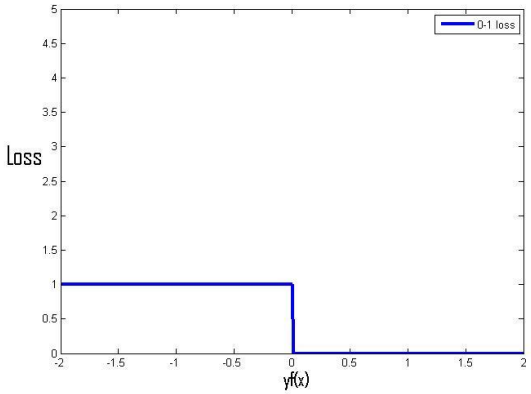


图 1　0-1 损失函数图像

然而，在实际训练分类器的过程，这种 0-1 损失基本上不被使用， 因为L_{0-1}不是一个凸的函数，即是局部最优不同于全局最优，导致很多优化方法如梯度下降，共轭梯度法的使用遇到困难。因此，各种各样的代理损失函数就被发明出来，代替L_{0-1} 成为被优化的目标函数。显然，它们都有一个共同的特点，就是它们都是凸函数。下面介绍一些常用的代理损失函数。

3.2 几个常见代理损失函数

3.2.1 Hinge 损失（SVM）

Hinge 损失函数是支持向量机（SVM）的使用的损失函数，随着支持向量机的流行，Hinge 损失函数也被深入研究并出现了多个变种，下面是 hinge 损失函数的定义：

$$L_{hinge}(y,f)=max(1-yf,0)$$

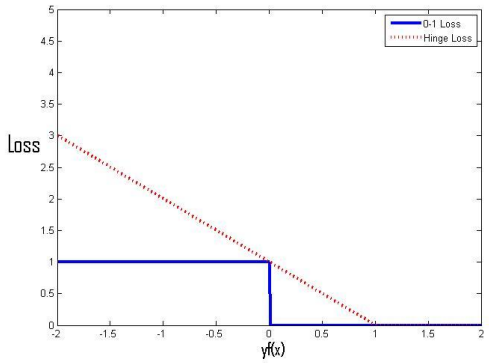


图 2 hinge 损失函数图像

3.2.2 指数损失(Logistic Regression ， AdaBoost)

指数损失函数是 Logistic 回归和 AdaBoost 等算法隐含的损失函数，其定义如下：

$$L_{exp}(y,f) = e^{-yf}$$

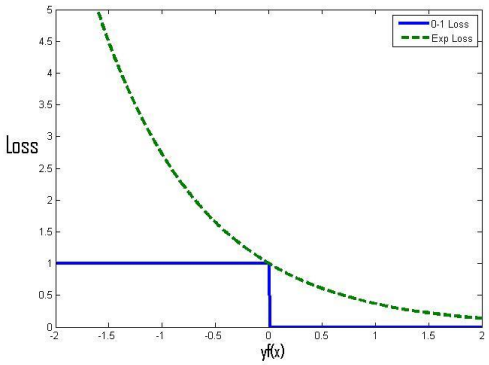


图 3 指数损失函数图像

3.2.3 似然损失(最大熵模型，LogitBoost)

似然损失函数是最大熵模型、LogitBoost 和其他基于概率的模型常用的损失函数，其定义如下：

$$L_{log\ -lik}(y,f) = log_2(1 + e^{-2yf})$$

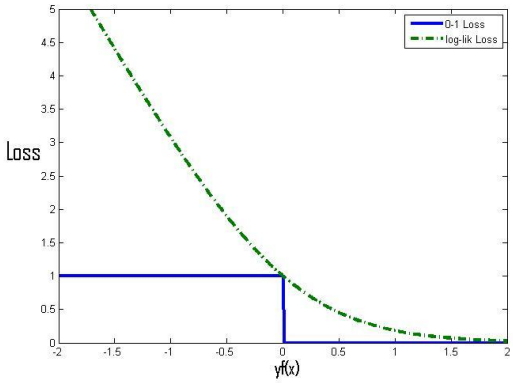


图 4 似然损失函数图像

3.2.4 L2 损失(RLSC)

L2 损失函数是回归类算法常用的损失函数，但也可以用于分类，典型例子是用于分类的正则化最小平方误差算法（Regularized Least Square Error，RLSC），其定义如下：

$$L_{L2}(y,f) = (1-yf)^2$$

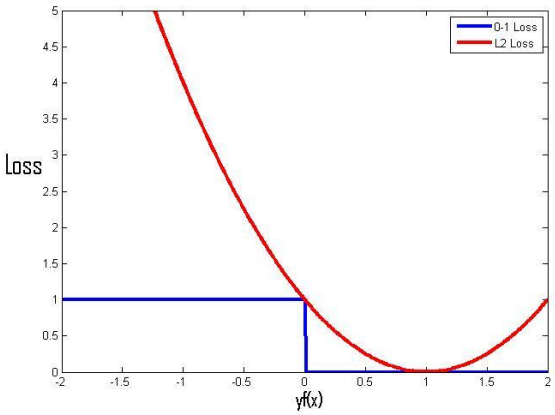


图 5 L2 损失函数图像

3.2.5 L1 损失

L1 损失函数也是回归类算法常用的损失函数，但因为其非处处可微，所以实现上常用 L2 损失函数近似，其定义如下：

$$L_{L1}(y,f) = |1-yf|$$

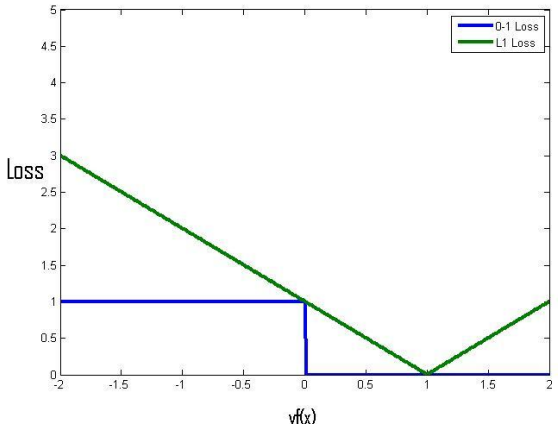


图 6 L1 损失函数图像

人们通常把 yf(x)的值称为“margin”（间隔），上面前 3 个代理损失函数都是“Margin Maximizing”（间隔最大化）类的损失函数。我们看到它们在区间[0,1]上的损失不为 0，而且越接近 0 的损失值就越大，而 0-1 损失则只要大于 0，其损失值就为 0。“Margin Maximizing”类损失函数的特点就是让优化得到的预测函数在已知样本上的 margin 尽量大，其作用的结果就是最小化结构性误差。如下面左图是使用没有“Margin Maximizing”特点的损失函数，右图是用了“Margin Maximizing”特点的损失函数。虽然两者都成功分类，但明显右图的泛化能力要强些。

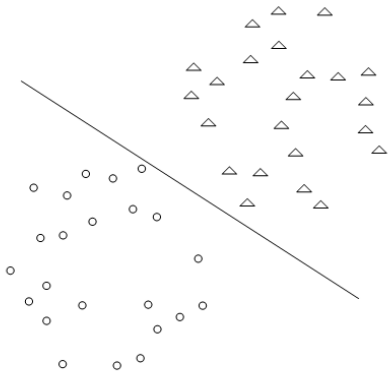


图 7 非间隔最大化分类

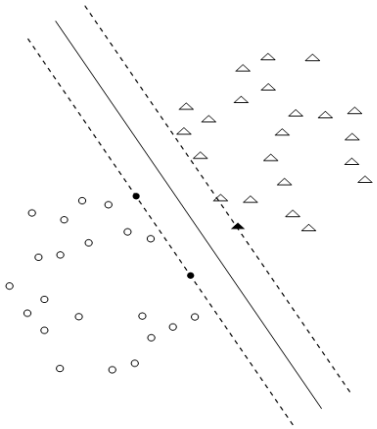


图 8 间隔最大化分类

第四章　不平衡问题产生的根源

我们看到因为凸性的要求，间隔（margin）值越小，几个代理损失函数的值就越大，而且增长的速度也是越来越快的(hinge 损失和 L1 损失的增长速度不变）。这样的好处是惩罚预测函数所犯的“严重错误”，而坏处是对那些离群点（outliner）特别敏感，而且这个坏处在不平衡的情况下被放大。另外，像 L2 损失不是一个单调递减的函数，它在区间[1,+∞) 中是递增的，就是说太过“肯定”也是不赞成的，这个限制对分类来说会带来些麻烦，也是不平衡问题的一个原因。还有的损失函数像指数损失和似然损失在区间[1，+∞) 严格下降，也是可能是做成分类性能下降的原因。下面将对凸损失函数分三种情况进行讨论。

4.1 [0，+∞）非单调下降的损失函数

这类的损失函数通常用在回归问题上面，不过也有人把二分类问题看成值域为{-1，1}的回归问题来处理。L2 损失和 L1 损失都属于这类损失函数，而使用这类损失函数的最常见的分类器是 RLSC(Regularized Least Squares for Classification)。下图是 RLSC 对平衡数据进行分类的结果。可以看到有不错的分类结果。

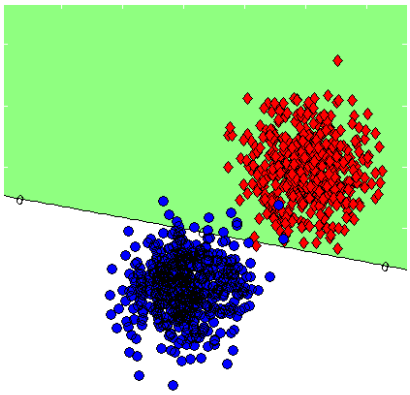


图 9　RLSC 数据平衡的分类情况

但是如果对不平衡数据分类的话，其结果就不让人满意了。RLSC 把所有样本都分为较大的一类。

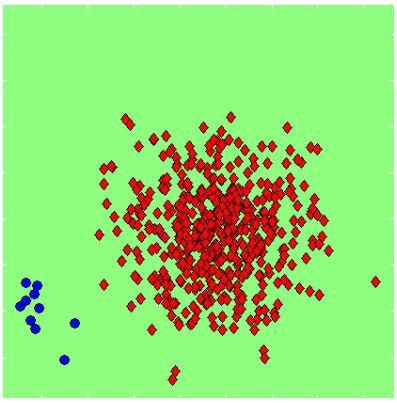


图 10 RLSC数据不平衡的分类情况

下面我们来分析一下出现这个现象的原因。

基于这类损失函数的算法可以等价于下面这个最优化问题的表述：

$$min_{w\in\mathcal{H}}\ \frac{1}{n}\sum_{i=1}^{n}L(\varepsilon_i)+\lambda\|\boldsymbol{w}\|^2 \tag{2}$$

且满足：

$$|\delta-y_i\boldsymbol{w}^T\boldsymbol{x}_i|\le\varepsilon_i \qquad i=1,\cdots,n$$

$$\varepsilon_i\ \ge 0 \qquad\qquad i=1,\cdots,n$$

其中$L(x)$是在$[0,+\infty)$区间上的递增函数，δ 为零损点，例如 L2-Loss 对应的就是$L(x)=x^2$ 且 $\delta=1$；而ε_i 通常被称为“软间隔”。

显然为了最小化目标函数就必须令每个“软间隔”的值尽量小，换句话说，由(2)得到的最优解 $\boldsymbol{w}$，必须令$y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i$ (margin)尽量接近零损点 δ。

接着上面 RLSC 的例子，在平衡数据下，这时候(2)的最优解 $\boldsymbol{w}$ 令超平面$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}=1$ 和超平面$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}=-1$分别基本贯穿正类和负类的最大密度地区，从而令$y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i$的值可以尽量接近零损点 $\delta=1$。如下图：

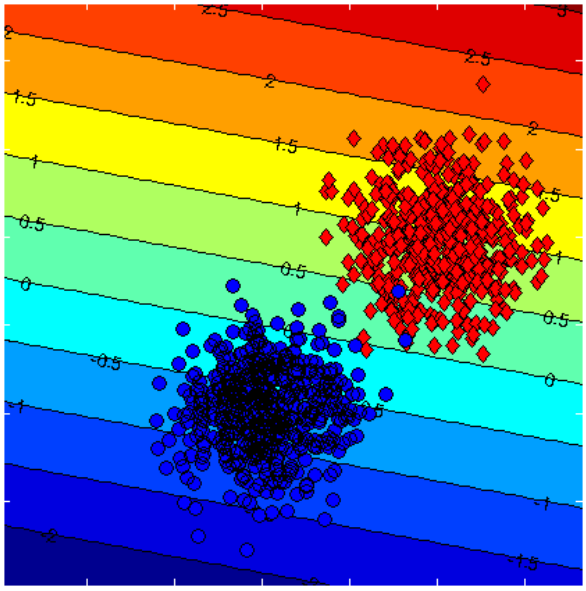


图 11 RLSC数据平衡的分类面梯度图

然而如上图二维情况所示，两个超平面的距离是由 $\boldsymbol{w}$ 决定，更准确地说，$\boldsymbol{w}$ 决定了这些超平面垂直方向的梯度。在平衡数据的情况（如图 1），正类（红色）被$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = 0$ 到$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = 2$的区间覆盖，距离为 2 个梯度。而在不平衡的情况，对于同样的正类数据，$\mathbf{w}$ 变化了，正类只需要用 0.8 个梯度覆盖（0.6~1.4）。于是，对应正类的损失值：

$$\sum|\delta - y_i \boldsymbol{w}^T \boldsymbol{x}_i|, \qquad 其中 i \in \{\alpha | y_\alpha = 1\}$$

不平衡的情况要远远小于平衡的情况，当然代价是负类的损失值增大，不过由于在不平衡下，负类样本数远少于正类，因此损失值的增加量小于减少量。

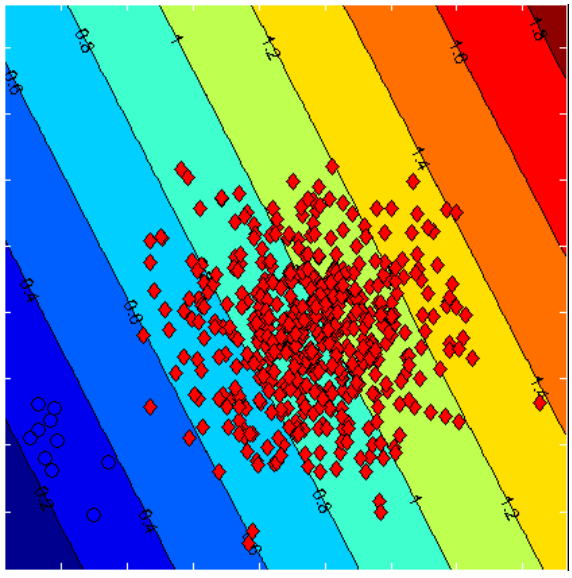


图 12 RLSC数据不平衡的分类面梯度图

总的来说，这类损失函数不太适合用于分类任务，因为它加了一个很不必要的约束就是间隔（margin）必须落在零损点附近，而在不平衡的情况下，由(2)式得到的最优模型会倾向让大类的样本落在零损点附近而忽视小类样本。

4.2［0，+∞）严格单调递减的损失函数

指数损失和似然损失就属于这一类的损失函数，这两个损失函数引起了人们的广泛兴趣。例如似然损失是最著名的 AdaBoost 的隐含损失函数，以及指数损失被用于 logistic 回归和最近用在 boosting[4]等等。基于这类损失函数的算法可以等价于下面这个最优化问题的表述：

$$\min_{\mathbf{w}\in\mathcal{H}} \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} L(\varepsilon_{\mathrm{i}}) + \lambda\|\boldsymbol{w}\|^2$$

(3)

且满足：$$y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq \varepsilon_i \qquad i = 1, \cdots, n$$

其中 $L(x)$是在$(-\infty, +\infty)$区间上的严格单调递减的函数。

现在假设在数据可分的情况，即(3)存在解（超平面，不一定是最优）使$\forall i, \varepsilon_i \geq 0$， 给其一个垂直于超平面方向的微小扰动$\Delta\varepsilon$, 则目标值的变化为：

$$\Delta\mathrm{F} = \sum_{\mathrm{i}\in\Lambda_-} -L'(\varepsilon_{\mathrm{i}})\Delta\varepsilon + \sum_{\mathrm{i}\in\Lambda_+} L'(\varepsilon_{\mathrm{i}})\,\Delta\varepsilon$$

(4)

其中Λ_-和Λ_+分别代表负例和正例的指标集，L'为损失函数的一阶导数，且$L'(x)$恒小于 0。若L'为常数 $k<0$,不平衡情况有$\mathrm{n}_- < \mathrm{n}_+$,这时候，根据（4）式得：

$$\Delta\mathrm{F} = k(\mathrm{n}_+ - \mathrm{n}_-)\Delta\varepsilon$$

(5)

当$\Delta\varepsilon > 0$的时候，$\Delta\mathrm{F} < 0$；换句话说，超平面向负类方向移动可以获得更小的目标函数的值，且当L'为常数的话，这样的移动会不断进行下去。

虽然上面L'为常数的假设并不出现在常见的损失函数中，但说明了一点，在不平衡的情况下要从一个可行解超平面（$\forall i, \varepsilon_i \geq 0$）移动达到一个平衡点（$\Delta\mathrm{F} = 0$），$L'(\varepsilon)$必须在

$(0,+\infty)$上随着ε增大而增大（$L'(\varepsilon)<0$,因为是递减函数），这样(4)式中$\Delta F=0$才会存在一组解$\{\varepsilon_i\}$。为了进一步讨论平衡点对分类器性能的影响，我将会通过引入例子“广义指数损失函数”来说明。

由上面假设L'为常数的例子中可以看到，其结果是分类边界一直向小类（负类）方向移动，最后整个空间都被分作正类。直觉上，我们认为L'的性质影响最终平衡点从而影响分类性能。为了更形象地说明问题，我引入“广义指数损失函数”：

$$L_{\mathrm{GExp}}(\mathrm{x})=e^{-kx} \qquad k>0 \tag{6}$$

$$且 L'_{GExp}(x)=-ke^{-kx}$$

从定义知道，当 k=1 的时候，广义指数损失函数就相当于指数损失；当 k->+∞的时候，L_{GExp}就趋向于$L_{0-\infty}$,其定义如下：

$$L_{0-\infty}(x)=\begin{cases}+\infty & x\le 0\\ 0 & x>0\end{cases}$$

由定义知道，在固定 *x* 下，L'_{GExp}的值随着 *k* 增大而增大并趋向于 0，换句话说，L_{GExp} 对 *x* 的增大越来越不敏感，而对 *x* 的减少而来越敏感，如下图所示，对 0-1 损失和几个不同 k 值的广义指数损失函数进行比较：

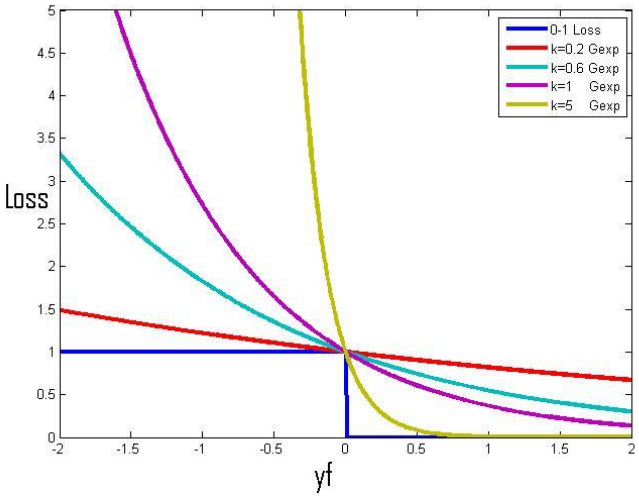


图 13 广义指数损失函数图像

同时，把上面不同 *k* 值的L_{GExp}代入(3)式，对不平衡数据进行实验，得到对应 *k* 值的最优分割平面（即(4)中的平衡点）如下面图表所示：

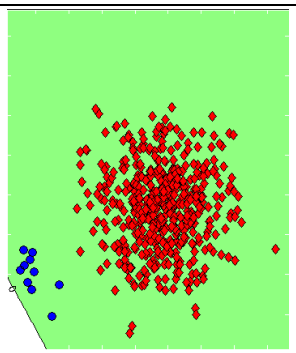
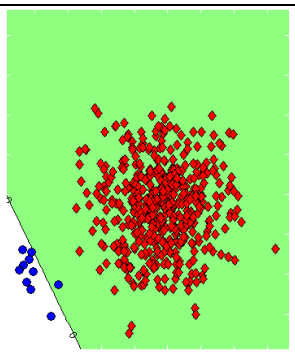
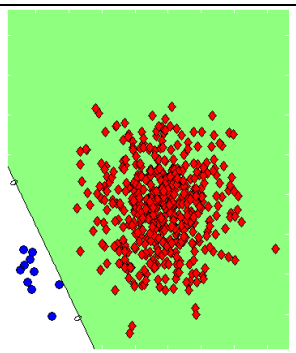
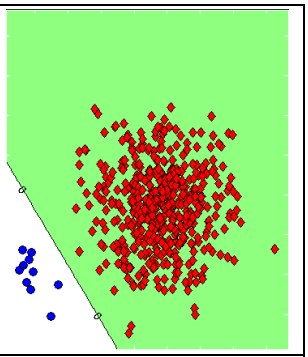
			
k=0.2	k=0.6	k=1	k=5

表 1 分类边界随参数 k 变化情况

实验的结果符合我们理论分析，在可分的数据下，当 k 值较小的时候，最终平衡点离原可行解（0-1 损失为 0 的解）较远，即向小类方向移动了较远的距离，这是因为 L_{GExp} 对 ε_i 的减少带来的损失值上升反应不够，而对 ε_i 的增加带来的损失值下降反应过大，导致需要较大距离才达到平衡，从而导致分类器性能下降；反之，当 k 值较大的时候，最终平衡点距离原解较近，分类能力相对较好。

总的来说，为了避免这种原因做成的分类器性能下降，应该使损失函数在 $(0,+\infty)$ 的导数尽量快地接近 0。

4.3 其他的凸损失函数

上面已经考虑了两种类型的损失函数，且上面的讨论都是基于可分的数据集，而这一小节，将讨论剩下的所有凸损失函数，并且在有噪声的情况下进行讨论。

究竟剩下的损失函数是怎样的呢？既然是损失函数，它在 $(-\infty,0)$ 上面肯定是非递增的，而且因为要求凸性，即要求其二阶导数 L'' 在 $(-\infty,+\infty)$ 上都必须大于或者等于 0。因为上面已经讨论了 [0，+∞) 上非单调递减和严格单调递减的损失函数，现在就来讨论 [0，+∞) 非严格单调递减的损失函数，其中一个例子就是 hinge 损失。基于这类损失函数的算法可以等价于下面这个最优化问题的表述：

$$\min_{\mathbf{w}\in\mathcal{H}} \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} L(\varepsilon_{\mathrm{i}}) + \lambda\|\boldsymbol{w}\|^2 \tag{7}$$

且满足：

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq \delta - \varepsilon_i \qquad i = 1,\cdots,n$$

$$\varepsilon_i \geq 0 \qquad i = 1,\cdots,n$$

其中 $L(x)$是在$(0,+\infty)$区间上的非递减函数，例如 hinge 损失对应 $L(x)=x$ 和$\delta=1$，从(7)得到著名的 SVM 的最优化问题表述：

$$\min_{\mathbf{w}\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i + \lambda\|\boldsymbol{w}\|^2$$

(8)

且满足：

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq 1-\varepsilon_i \qquad i=1,\cdots,n$$

$$\varepsilon_i \geq 0 \qquad i=1,\cdots,n$$

类似上一节，为了更具体研究这类函数的性质，我将引入“广义 hinge 损失函数”，其定义如下：

$$L_{GHinge}(x) = (max(\delta - x, 0))^{k} \qquad k>0, \delta>0$$

(9)

从定义知道，当 k=1 的时候，广义 hinge 损失函数就相当于 hinge 损失；当 k=2 的时候，广义 hinge 损失函数就相当于平方 hinge 损失；当 k->+∞的时候，L_{GHinge} 就趋向于$L_{0-\infty}$。

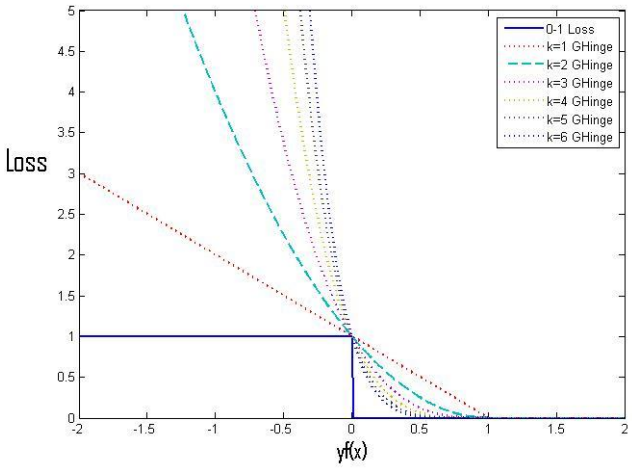


图 14 广义 hinge 损失函数图像

因此得到 Tikhonov 正则化框架下基于广义 hinge 损失函数的最优化问题的表述如下：

$$\min_{\mathbf{w}\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^{\,k} + \lambda\|\boldsymbol{w}\|^2$$

(10)

且满足:

$$y_i \boldsymbol{w}^T \boldsymbol{x_i} \geq \delta - \varepsilon_i \qquad i=1,\cdots,n$$

$$\varepsilon_i \geq 0 \qquad\qquad i = 1,\cdots,n$$

从定义我们知道，不同 k 值的广义 hinge 损失函数对相同 margin 的错分样本的惩罚不同，k 越大惩罚的力度越大而且随着 margin 的减少而增加的速度更快。在实际数据中，噪声普遍存在，而且不平衡的情况下，大类的样本噪声出现在小类样本中间是一个普遍存在的现象。因此，我猜想在 k 不断增大的情况下，L_{GHinge} 对L_{0-1}的偏离会越来越厉害，从而导致分类器对大类的噪声更加敏感。

为了研究大类噪声的影响，我在以前的不平衡可分的数据集上加了一个大类噪声样本，固定$\delta = 1$并把不同 k 值的代入(10)式，计算对应的最优分割超平面，如下图表所示：

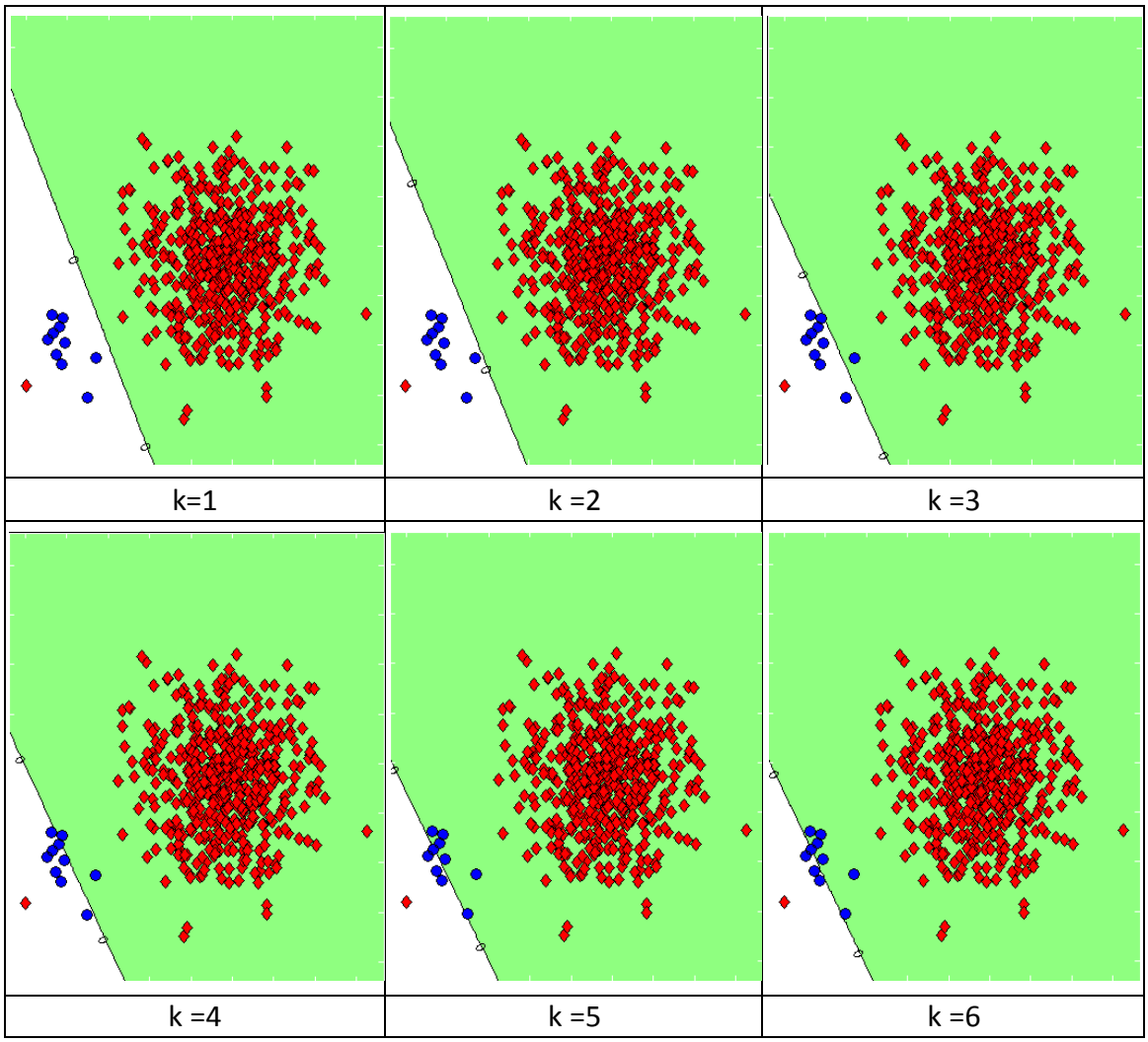


表 2　分类边界随参数 k 变化情况

虽然上面实验采用的是人工数据，但也可以看出这类代理损失函数严重夸大了噪声点的影响，为拟合一个噪声不息牺牲多个小类样本。另外，在上面实验中加入的噪声点还可以被看做为一个离群点([59])，或许可以用一些剔除离群点的方法去改善结果，但实际上，

因为通常来说大类的属性的方差比小类属性的方差大得多，即使在正常方差范围内的点也很有可能散布在小类当中，从而这样的效应会更加明显。

有趣的是，早在 1995，Cortes 和 Vapnik[24]就对广义 hinge 损失进行研究，它们的结论是当 $k>1$ 的时候，分类器会夸大最大的一个损失，并且对离群点比较敏感，但如果 $k<1$，虽然可以提高分类器的鲁棒性，但却成为了非凸的优化问题。

4.4 本章小结

第一节给出了非单调的凸损失函数（绝对值形式）的一般规划形式，并结合实验分析说明在数据不平衡情况下，优化结果是通过调整梯度来倾向大类数据，从而导致性能下降；为了研究严格单调下降的凸损失函数，第二节结合线性单调的例子和引入广义指数损失函数做实验验证，指出为了减少由于不平衡导致的分类平面移动，损失函数的导数应该尽快趋向于 0；第三节研究最后一种情况，即非单调下降情况，通过引入广义 hinge 损失函数指出其指数越大对离群点越敏感而数据不平衡加剧了这个趋势。

综上三个小节所述，每一种凸损失函数在一定的数据分布下都会导致分类性能下降，而根据上述分析，$k=1$ 的广义 hinge 损失函数（当$\delta = 1$时，即 hinge 损失）可能就是对不平衡数据最不敏感的凸损失函数了，但究竟怎样定义“不平衡不敏感”呢？下一章将会继续讨论这个问题。

第五章　不平衡不敏感损失函数（imbalance insensitive loss function ）

5.1 不平衡不敏感的定义

上一章解释了凸损失函数在数据不平衡的情况下导致分类性能下降的各种原因，直觉认为所有凸损失函数都是对数据不平衡敏感的，那么什么损失函数对不平衡不敏感呢？如何定义不敏感呢？这一章将从理论上解答这些问题，并给出了一个损失函数对不平衡不敏感的充分不必要条件。

定义：如果一个损失函数，在(11)式中λ趋向于0的时候得到的最优解$\boldsymbol{w}^*$也是L_{0-1}的最优解，则称它为“不平衡不敏感损失函数”。

$$F_L(X) = \min_{\mathbf{w}\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} L(y_i \boldsymbol{w}^T \boldsymbol{x}_i) + \lambda\|\boldsymbol{w}\|^2 \tag{11}$$

也就是说若 L 是“不平衡不敏感损失函数”，则满足：

$$\lim_{\lambda\to 0}\{F_L \text{的最优解集}\} \subseteq \{F_{L_{0-1}} \text{的最优解集}\} \tag{12}$$

从定义知道，“不平衡不敏感”的定义是相对于“0-1 损失函数”来说的，但这对于分类问题来说也是非常自然的选择，而且后面我们可以看到这个定义对于由“0-1 损失函数”的仿射变换得到的新损失函数都是等价，也很容易推广到代价敏感的损失函数上。

5.2 不平衡不敏感的充分条件

若只根据上面“不平衡不敏感”的定义，我们很难迅速去判断一个损失函数是否满足该定义。为了方便设计出不平衡不敏感的损失函数，我提出了下面的两个条件：

1. 条件一：

$$\forall k > 1, \qquad L(k\cdot m) \le L(m) \tag{13}$$

2. 条件二:

$$\exists k_0 > 1, \forall k \ge k_0, \exists a > 0, b \in \mathbb{R}$$

$$L(k \cdot m) = aL_{0-1}(m) + b$$

(14)

定理 5.1：如果一个损失函数 *L* 满足上面两个条件，则它是“不平衡不敏感损失函数”。

引理 5.2：若 *L* 为满足条件(13)(14)的损失函数，则F_L的最优解 $\boldsymbol{w^*}$必须满足，对于任意样本下标 *i*，有

$$L(y_i \cdot \boldsymbol{w}^{*T}\boldsymbol{x}_i) = aL_{0-1}(y_i \cdot \boldsymbol{w}^{*T}\boldsymbol{x}_i) + b$$

(15)

引理 5.2 的证明：用反证法，假设$\boldsymbol{w_1}$为F_L的最优解但不满足（15）式，令 $\varepsilon_i = y_i \cdot \boldsymbol{w_1}^T\boldsymbol{x_i}$，根据条件（14）则存在一个正数 *k>1*，使得

$$L(k\,\varepsilon_i) = aL_{0-1}(\varepsilon_i) + b \le L(\varepsilon_i)$$

又因为$\boldsymbol{w_1}$不满足（15）式，即右边的等号条件不成立，得：

$$L(k\,\varepsilon_i) = aL_{0-1}(\varepsilon_i) + b < L(\varepsilon_i)$$

那么令$\boldsymbol{w_2}$=*k*·$\boldsymbol{w_1}$，那么$y_i \cdot \boldsymbol{w2}^T\boldsymbol{x_i} = k \cdot \varepsilon_i$，得：

$$\begin{aligned}\lim_{\lambda\to 0} F_L(\boldsymbol{w_2}) &= \sum_{i=1}^{n} L(y_i \cdot \boldsymbol{w_2}^T x_i) \\ &= \sum_{i=1}^{n} L(k \cdot \varepsilon_i) \\ &< \sum_{i=1}^{n} L(\varepsilon_i) \\ &= \lim_{\lambda\to 0} F_L(\boldsymbol{w_1})\end{aligned}$$

这显然与$\boldsymbol{w_1}$是最优解矛盾。

定理 5.1 的证明：用反证法，假设 *L* 为满足条件(13)(14)的损失函数，并假设$\boldsymbol{w_1}$为F_L的最优解，存在一个非最优解$\boldsymbol{w_2}$，令

$$\lim_{\lambda\to 0} F_{L_{0-1}}(\boldsymbol{w_1}) > \lim_{\lambda\to 0} F_{L_{0-1}}(\boldsymbol{w_2})$$

(16)

令 $\varepsilon_{1i} = y_i \cdot \boldsymbol{w_1}^T\boldsymbol{x}_i$和 $\varepsilon_{2i} = y_i \cdot \boldsymbol{w_2}^T\boldsymbol{x}_i$，由条件(14)知存在一个正数 *k*>1,使得

$$L(k \cdot \varepsilon_{2i}) = aL_{0-1}(\varepsilon_{2i}) + b \qquad \text{对所有}i\text{成立}$$

(17)

其中 *a* 为正常数，*b* 为常数。

则可以得到：

$$\lim_{\lambda\to 0} F_L(k\cdot \boldsymbol{w2}) = \sum_{i=1}^{n} L(k\cdot \varepsilon_{2i}) = \sum_{i=1}^{n} (aL_{0-1}(\varepsilon_{2i}) + b) = a\left(\lim_{\lambda\to 0} F_{L_{0-1}}(\boldsymbol{w_2})\right) + nb$$

由于 a>0 和(16)式

$$< a\left(\lim_{\lambda\to 0} F_{L_{0-1}}(\boldsymbol{w_1})\right) + nb \quad (18)$$

又因为由引理 5.2 知道，对于最优解 w1 有：

$$\lim_{\lambda\to 0} F_L(\boldsymbol{w_1}) = \sum_{i=1}^{n} L(\varepsilon_{1i}) = \sum_{i=1}^{n} (aL_{0-1}(\varepsilon_{1i}) + b) = a\left(\lim_{\lambda\to 0} F_{L_{0-1}}(\boldsymbol{w_1})\right) + nb \quad (19)$$

联合(18)，(19)得：

$$\lim_{\lambda\to 0} F_L(\boldsymbol{w_1}) > \lim_{\lambda\to 0} F_L(k\cdot \boldsymbol{w_2})$$

这显然与$\boldsymbol{w_1}$是最优解这个事实矛盾，所以定理 5.1 成立。

5.3 一些“不平衡不敏感”损失函数的例子

从上面两个小节知道，除了“0-1 损失”本身，我们还可以用其他损失函数完全地替换“0-1 损失”函数，并且我们期待这些函数比“0-1 损失”函数有一些更好的性质。根据定理 5.1，我们很容易得到一些“不平衡不敏感”损失函数，它们的函数图像如下图表所示：

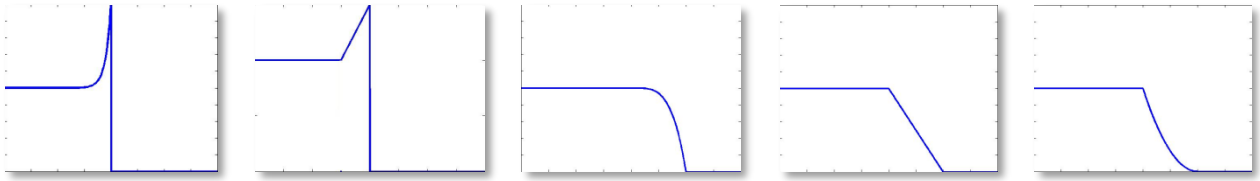


表 3 “不平衡不敏感损失函数”举例

前面两个函数就可能比较意想不到，它们竟然倾向于把错误的间隔扩大化，不过可以验证这类函数的确属于“不平衡不敏感”损失函数。而最后两个函数，形状像滑梯，我把它们命名为“滑梯函数”。

滑梯函数的定义如下：

$$L_{Slide}(x)=\begin{cases}1, & x<0\\ (max(1-x,0))^{k}, & x\geq 0\end{cases}$$

其中$k\geq 1$

从定义知道，滑梯函数相对于零损点为 1 的广义 hinge 损失函数差别只在于 *x*<0 的部分。下面我们证明滑梯函数跟广义 hinge 损失函数一样都是“间隔最大化”的损失函数。

根据文章[50],一个损失函数是“间隔最大化”的损失函数的充分条件如下：如果 $\exists \mathrm{T}>0,(\mathrm{T}$ 可能是无穷$)$，使得

$$\lim_{t\nearrow T}\frac{L(t\cdot[1-\epsilon])}{L(t)}=\infty \qquad \forall\epsilon>0$$

(20)

则L为“间隔最大化”的损失函数。

对于滑梯函数来说，间隔大于 1 的损失值为 0，因此令 T=1，显然满足(20)式。相比之下，“0-1 损失”函数显然不是“间隔最大化”损失函数。通常认为，在“间隔最大化”损失函数下得到的预测模型在泛化能力较好，因为它可以最小化“结构性误差”([56],[50])。此外，滑梯函数相对于“0-1 损失”函数还有连续性，和局部凸性的优点。

5.4 算法的实现

虽然滑梯函数相对于“0-1 损失”函数来说有很多好的性质，但毕竟滑梯函数不是凸损失函数，意味着很可能无法采用以往的凸优化策略以及得不到全局最优解。但下面将介绍目前两种非凸优化的逼近方法，

（一）随机梯度下降法：

随机梯度下降法是解决非凸优化中最常见的方法，著名的算法有“遗传算法”、“模拟退火算法”、“粒子优化算法”等等。正如，分类问题一样，很多机器学习问题的自然表述都是非凸优化问题，Yann[65]建议应该放弃传统凸优化的模式采用随机梯度下降法来进行训练。

以“遗传算法”为例，我们可以定义下面的函数作为适应度函数：

$$f\ （w）= \frac{1}{n}\sum_{i=1}^{n} L_{slide}\,(y_i \boldsymbol{w}^T x_i) + \lambda\|\boldsymbol{w}\|^2$$

初始种群值我们可以用 SVM 或者 RLSC 求解得到。在 Matlab 的 Genetic Algorithm and Direct Search 工具包中包含“遗传算法”、“模拟退火算法”等等随机梯度下降算法，我们可以定义适度函数（目标函数）和初始值后就可以直接应用求解。

（二）半定规划（SemiDefine Programming）

有些文章([59][58])也把 *k*=1 的滑梯函数称为“鲁棒 hinge 损失”，定义$L_{Slide\ ,k=1} = L_{robust}$，下面将会重点研究$L_{robust}$ 损失函数。根据[59]的定理 1，对于“鲁棒 hinge 损失”的最优化问题可以转化为在“η − hinge损失”下对$\boldsymbol{\eta}$和$\boldsymbol{w}$同时进行优化的最优化问题，其中η − hinge损失定义如下:

$$L_{\eta-hinge}\,(x) = \eta \times (max(1-x,0)) + 1 - \eta \qquad 0 \le \eta \le 1$$

且有

$$\begin{aligned} &\min_{\boldsymbol{w}} \min_{0\le\eta\le 1} \frac{1}{n}\sum_{i=1}^{n} L_{\eta_i-hinge}\,(y_i w^T x_i) + \lambda\|\boldsymbol{w}\|^2 \\ &= \min_{\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^{n} L_{robust}\,(y_i w^T x_i) + \lambda\|\boldsymbol{w}\|^2 \end{aligned} \tag{21}$$

上式可以理解为在L_{robust} 的最优解中每个样本对应的损失值都可以通过调节η的值，使其在“η − hinge损失”下得到同样的损失值。下图可以更形象地说明这个原理：

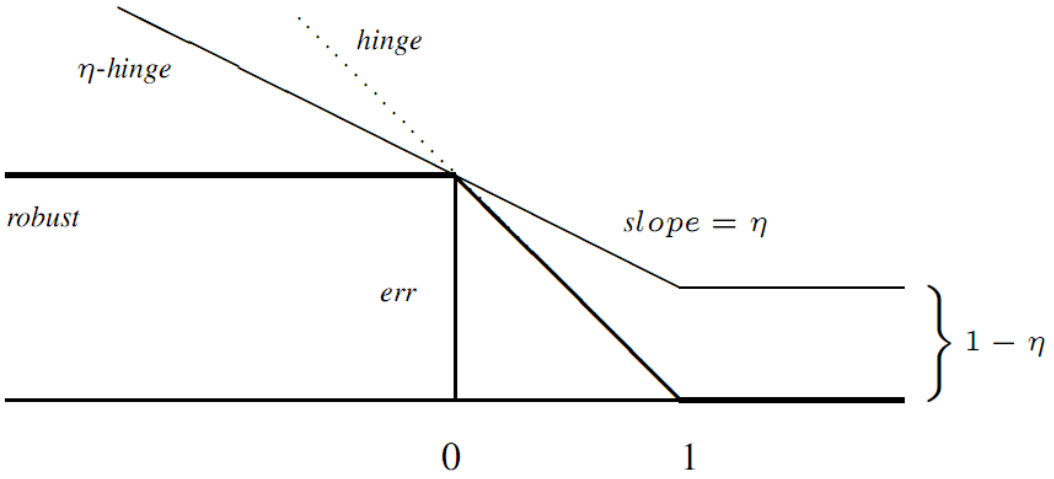


图 15 “η − hinge 损失”示意图

引入松弛变量后，可以把（21）式表述为下面的规划形式：

$$\min_{\boldsymbol{w}} \min_{0 \le \boldsymbol{\eta} \le 1} \frac{1}{n}\sum_{i=1}^{n} (\varepsilon_i + 1 - \eta_i) + \lambda\|\boldsymbol{w}\|^2$$

且满足：

$$\varepsilon_i \ge \eta_i(1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i) \qquad i = 1, \cdots, n$$

$$\varepsilon_i \ge 0 \qquad i = 1, \cdots, n$$

(22)

又根据[59]的定理 2，解决(22)就可以用下面的 SDP 问题来近似：

$$\min_{\boldsymbol{\eta}, M, \boldsymbol{\mu}, \boldsymbol{\nu}, \zeta} \quad \zeta \qquad subject\ to\ \ \boldsymbol{\mu} \ge 0,\ \boldsymbol{\nu} \ge 0,\ 0 \le \boldsymbol{\eta} \le 1,\ M \succeq \boldsymbol{\eta}\boldsymbol{\eta}^{\top},\ diag(M) = \boldsymbol{\eta},$$

$$\begin{bmatrix} G \circ M & \boldsymbol{\eta} + \boldsymbol{\mu} - \boldsymbol{\nu} \\ (\boldsymbol{\eta} + \boldsymbol{\mu} - \boldsymbol{\nu})^{\top} & \frac{2}{\beta}(\zeta - \boldsymbol{\nu}^{\top}\mathbf{e} + \boldsymbol{\eta}^{\top}\mathbf{e}) \end{bmatrix} \succeq 0$$

(23)

其中 $G = X^{\top}X \circ \mathbf{y}\mathbf{y}^{\top}$，$\beta = \lambda$，$\mathbf{e}$ 为元素全为 1 的向量。

又根据 Schur Complement 定理，(23)式中的二次约束 $M \succeq \boldsymbol{\eta}\boldsymbol{\eta}^{\top}$ 可以转化为：

$$\begin{bmatrix} \mathrm{M} & \boldsymbol{\eta} \\ \boldsymbol{\eta}^T & 1 \end{bmatrix} \succeq 0$$

这样就可以作为一个标准的半定规划问题来求解，现有很多免费的 SDP 工具箱都支持这个问题的形式，常用的有 SDPT3，Sedumi 等等。

5.5 代价敏感版本

在不平衡数据集中，通常小类错判的代价要大于大类错判的代价，例如一些潜在客户的发掘，非潜在客户的数量肯定远远多于潜在客户，但潜在客户远比非潜在客户远远重要。

为了应付代价敏感问题的需要，必须把上面的算法推广，使其可以包含类别代价信息。通过修改(21)式，把代价系数直接与损失值相乘就可以得到 SDP 方法的代价敏感版本的最优化目标：

$$\min_{\boldsymbol{w}} \min_{0 \le \boldsymbol{\eta} \le 1} \frac{1}{n}\sum_{i=1}^{n} c_i \times L_{\eta_i - hinge}(y_i \boldsymbol{w}^T \boldsymbol{x}_i) + \lambda\|\boldsymbol{w}\|^2$$

(24)

其中c_i为类y_i对应的相对代价值，例如正类的错判代价与负类的错判代价之比为 4：1，若$y_i = 1$，对应的c_i=0.8，否则，c_i=0.2。类似[59]的推导，我们可以得到一个包含代价信息的 SDP 问题的表述如下：

$$\min_{\eta,M,\mu,\nu,\zeta} \quad \zeta \qquad subject\ to\ \ \mu \geq 0,\ \nu \geq 0,\ 0 \leq \eta \leq 1,\ M \succeq \eta\eta^{\top},\ diag(M)=\eta,$$

$$\begin{bmatrix} G \circ M & \eta + \mu + \nu \\ (\eta + \mu + \nu)^T & \frac{2}{\lambda}(\zeta - \nu^T c + \eta^T e) \end{bmatrix} \succeq 0$$

(25)

唯一区别在于矩阵右下角一项中$\nu^T e$改为$\nu^T c$, 其中 c 为相对代价值组成的向量。显然这个形式的 SDP 问题同样可以用上面提及到的工具包求解。

另外，代价敏感的随机梯度下降版本只要直接修改目标函数就可以，所以不做介绍。

5.6 实验

实验环境：

CPU	Intel P4 530　3.4GZ
Memory	1GB
OS	Windows XP SP2
Matlab 版本	Matlab 2007b

虽然半定规划(SDP)问题被证明可以在多项式时间内求解[19]，但半定规划算法目前在计算代价（无论是空间还是时间）上远远高于已经很成熟的线性规划和二次规划。在 1GB 内存下，三个最流行的半定规划 Matlab 工具包（SDPT3[54]，SEDUMI[66]，SDPLR[67]）能处理的最大样本数都不超过 70 个，否则就出现内存不足的错误。另外，因为实验中的涉及的损失函数比较多，所以先采用简易记法“类型-参数”， 例如“GE-1”就是 *k*=1 时候的广义指数损失函数，“GH-10”就是 *k*=10 的广义 hinge 损失函数。另外，SDP 代表采用半定规划逼近的 k=1 的滑梯函数，GA 代表采用遗传算法逼近的 *k*=1 的滑梯函数(即鲁棒 hinge 损失)，RLSC 代表 L2 损失函数，SVM 代表 hinge 损失函数，LapSVM[14]代表加上拉普拉斯平滑项的 SVM，也是基于 hinge 损失。

另外，RLSC、广义指数损失和广义 hinge 损失对应的算法实现在附录中给出。

5.6.1 人工数据实验

实验参数：λ =0.0142，采用 RBF 核（高斯核），RBF 宽度固定为 0.35。

实验数据是二维的“两月亮”数据，这个数据可以很形象的反映不同分类器在非线性数据下的情况。在 400 个数据平衡的样本中随机采样正例（红色）54 个，反例（蓝色）6

个构成训练集，剩下 340 个样本作为无标签数据（无色）。代入不同的损失函数进行训练得到分类器如下图表：

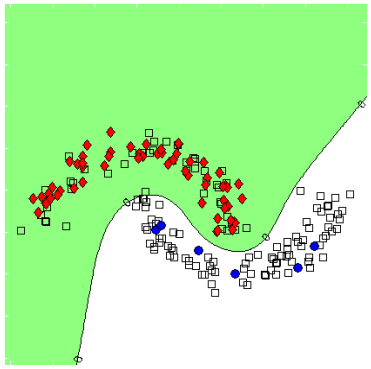
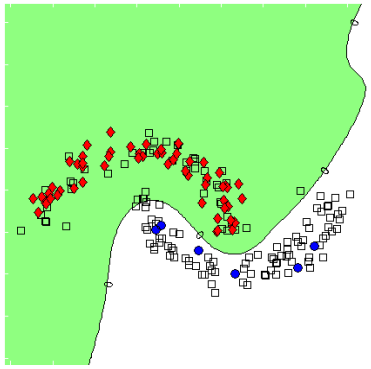
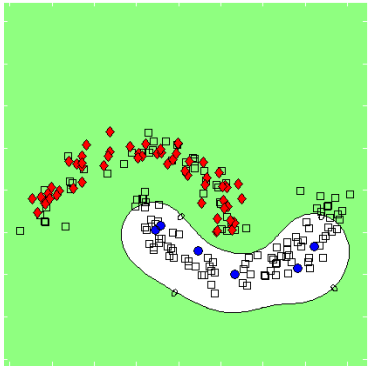
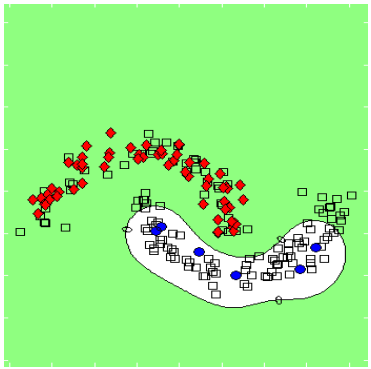
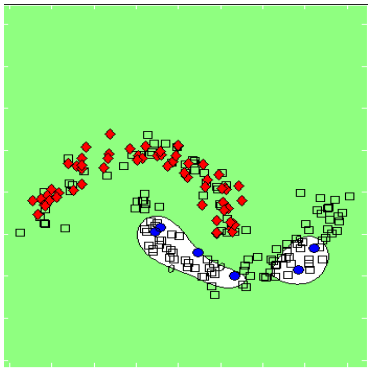
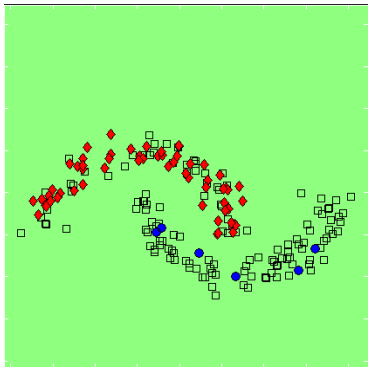
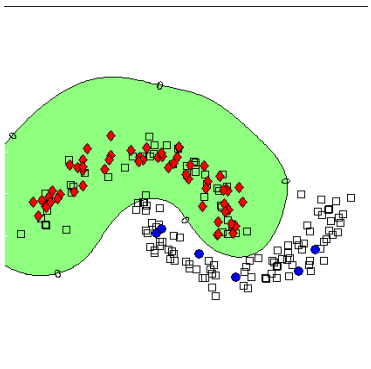
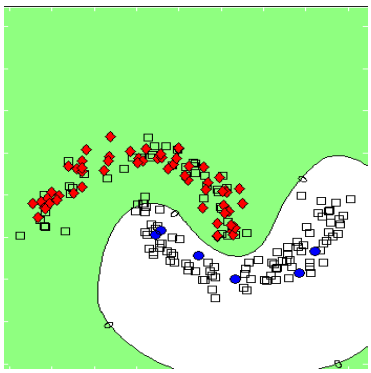
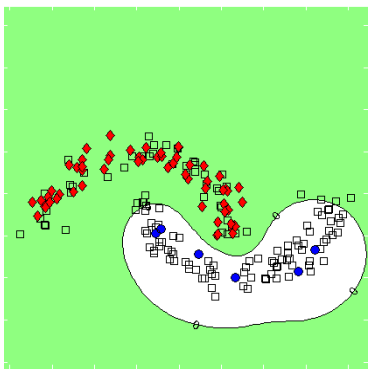
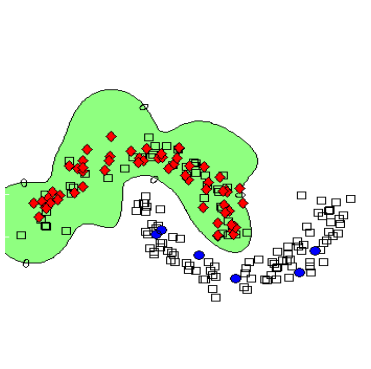
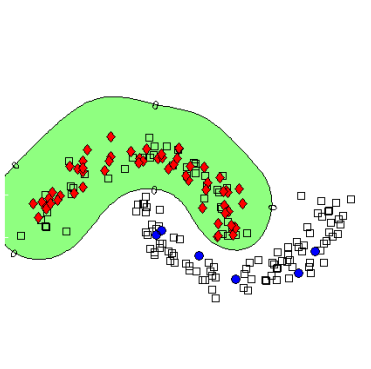
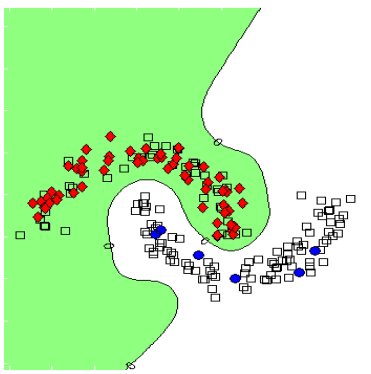
		
SDP	RLSC	GE-1
		
GE-0.5	GE-0.2	GE-0.1
		
GH-2	GH-5	GH-10
		
GH-1000	SVM/GA	LapSVM

表 4　各种损失函数下的 RBF 分类边界

上面的分类结果可以大致反映一些损失函数的特性，例如 SDP 和 RLSC，LapSVM 的分类边界是比较理想的。另外我们看到广义指数损失函数随着系数的下降分类性能下降，与第四章的分析相符。广义 hinge 损失函数类随在指数的增大表现出有趣的震荡特性，先扩张后收缩，这应该是不同指数下间隔最大化的表现。

5.6.2 实际数据实验

这里 UCI 数据库[12]中选取了 6 个不同比例的二分类数据，它们是“haberman”,“hepatitis”、“ionosphere”、“pima-indians-diabetes”、“sick-euthyroid”、“wdbc”，详情如下：

数据集	总样本数	类别比例	属性维数
haberman	306	225:81	4
hepatitis	155	85:70	20
ionosphere	351	225:126	35
pima-indians-diabetes	768	268:500	9
sick-euthyroid	3263	293:2870	26
Wdbc(Breast Cancer)	569	212:357	31

表 5　数据集描述

实验一（SDP 正确性实验）：

实验参数：$\lambda = 0.00001$，采用 RBF 核（高斯核），RBF 宽度为 $\sqrt{\frac{\text{属性维数}}{2}}$

样本构成：这个实验主要针对 SDP 算法，考虑到它的空间时间复杂度大，因此规定每组训练样本的大小为 60 个，把一个数据集按 60 个分组，每次选一组作为训练集其余作为测试集，最后结果为几次测试结果的均值。

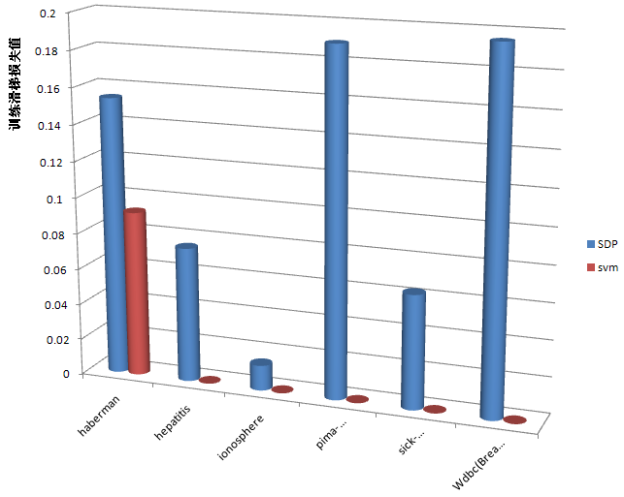


图 16　SPD 与 SVM 在训练集上一次滑梯误差的比较

实验结果分析：把λ设置这样小的原因是去掉平滑项的影响，直接测试不同损失函数的性质。

- 首先，发现 SDP 算法的训练误差在很多数据集中较大（图 16），但根据 5.4 节，SDP 算法应该最小化“鲁棒 hinge 损失函数”（即一次滑梯损失），显然实验结果与理论矛盾，意味着[59]提出的 SDP 松弛算法并不是优化“鲁棒 hinge 损失”。而 SDP 算法只在一处地方松弛了原来约束：$M = \eta\eta^T$ 松弛为$M \succeq \eta\eta^\top$，应该是这个松弛改变了优化目标。
- 因为训练样本少，在属性维数大的数据集上容易发生过拟合现象，这次实验结果也表明广义指数损失函数类的抗过拟合能力较强，而 RLSC(L2 损失)的抗过拟合能力较弱。

实验二（GA 正确性实验）：

类似 SDP 的实验设置，得到结果是 GA 算法确实等价于最小化滑梯损失，说明 GA 算法优化目标与理论相符。因此后面的实验采用 GA 算法与其他损失函数的算法比较。

实验三（不平衡程度非正则化实验）

实验参数：λ =0.00001，采用 RBF 核（高斯核），RBF 宽度为$\sqrt{\frac{\text{属性维数}}{2}}$

样本构成:选取“haberman”，“pima-indians-diabetes”进行本次实验，因为它们属性维数相对较少，不容易导致过拟合。对每个数据集进行重新采样（下采样，没有重复数据），生成 10 个大类与小类比例为[0.1,0.2,...,1]的不同不平衡程度的数据集。然后对每个新的数据集进行 5 重交叉检验的实验。

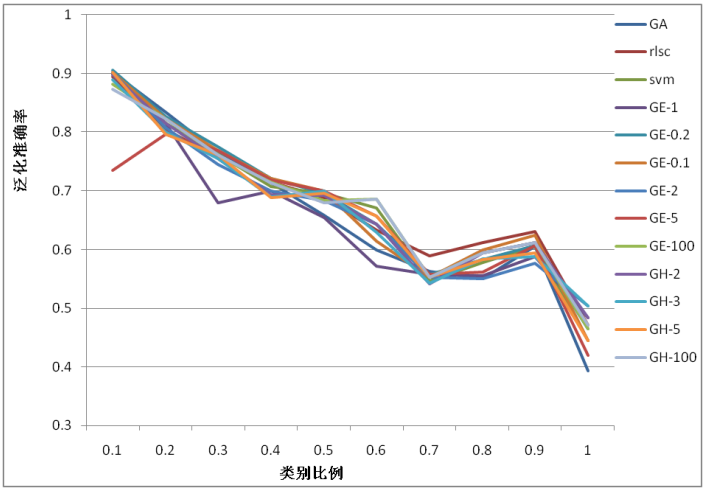


图 17 haberman 数据：不同比例 vs 准确率

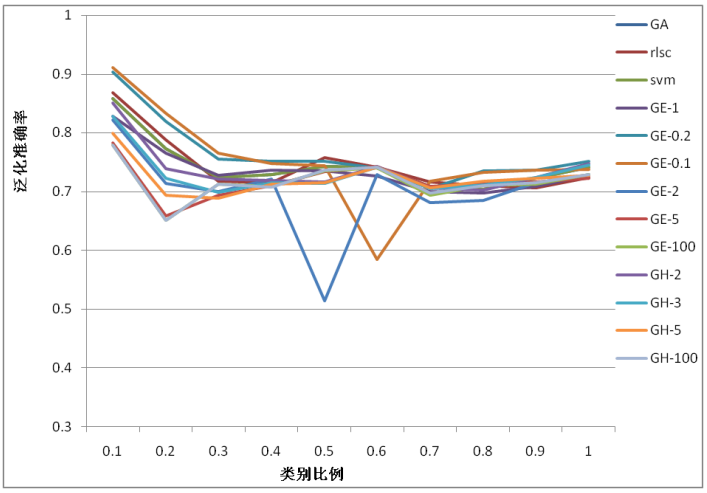


图 18　pima 数据：不同比例 vs 准确率

实验结果分析：同样，这次正则化系数λ仍然设置很小，以便忽略平滑项的影响。

- 对“pima-indians-diabetes”结果分析，看到无论任何比例下 GA 与 SVM 的结果都相同，意味着 SVM 得到的初始解就是“滑梯损失”的局部最小值。随着样本数增多，GA 要跳出局部最小值就越来越难。
- 同样对“pima-indians-diabetes”结果分析,SVM 算法在不同不平衡情况表现都较好。另外在不平衡程度较严重的时候，泛化误差随着广义 hinge 损失的指数增大而增大，即分类性能下降。
- 对“haberman”的结果分析，看到在不平衡比较严重时(比例为 0.1-0.4 时)GA 算法无论训练误差还是泛化误差都比 SVM 算法要小，之后的比例 GA 算法出现过拟合现象，表现反复。
- 同样对“haberman”的结果分析，从总体来说，无论什么损失函数，随着不平衡程度减少（比例升高），训练误差和泛化误差的都是一路增大，这个恐怕是因为数据本身对于 RBF 核来说的可分较差的原因。

实验四（不平衡程度正则化实验）

实验参数：$\lambda=0.5$，采用 RBF 核（高斯核），RBF 宽度为$\sqrt{\frac{\text{属性维数}}{2}}$

样本构成:选取“haberman”，“pima-indians-diabetes”进行本次实验，因为它们属性维数相对较少，不容易导致过拟合。对每个数据集进行重新采样（下采样，没有重复数据），生成 10 个大类与小类比例为[0.1,0.2,…,1]的不同不平衡程度的数据集。然后对每个新的数据集进行 5 重交叉检验的实验。

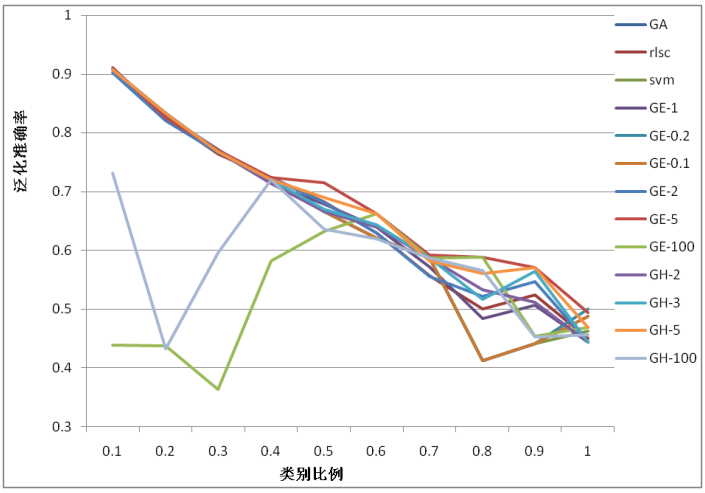


图 19　haberman 数据：不同比例 vs 准确率（正则化）

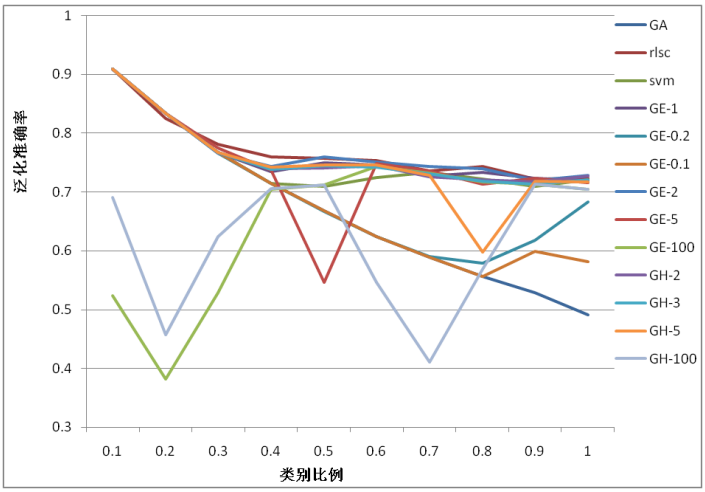


图 20　pima 数据：不同比例 vs 准确率（正则化）

实验结果分析：这次正则化系数λ设置为正常水平，考虑平滑项的影响。

- 因为平滑项会大大增加小类被掩盖的倾向，所以看到在较不平衡的时候多个损失函数都得到相同的结果。
- 虽然 RLSC 在第四章被认为不平衡敏感，但在“pima-indians-diabetes”数据集较不平衡的情况表现优秀，而在“haberman”中表现较差，说明损失函数的凸性在不平衡条件下产生正面还是负面影响是取决于数据的分布。
- GA 算法收敛于局部最优点，体现不出滑梯函数类的“间隔最大化”特性，导致训练得到分类器过拟合情况严重。

5.6.3 实验总结

“两月亮”数据集实验表明广义指数损失函数的泛化性能随指数下降而下降，而“pima-indians-diabetes”数据集实验表明广义 hinge 损失函数的泛化性随指数增大而下降，这都与第四章的分析吻合。

实验一证明[59]提出的 SDP 方法的优化目标已经不是原问题的一次滑梯损失;而实验二、实验三证明 GA 方法虽然符合优化目标，但很容易收敛于局部最优解并导致过拟合现象比较严重，无法体现间隔最大化的特性。因此，目前两种方法可以说都不能够实际解决非凸优化问题。

另外，在实验三，实验四中发现，在不平衡数据集分类性能差的原因可能与数据是否平衡无关，是因为假设空间选择问题。还有就是，虽然第四章指出了凸损失函数可能因为数据不平衡导致的问题，但至于是否真的出现是取决于数据分布，某些情况下一些凸函数的性质甚至在数据不平衡中有积极作用。

第六章　多分类的推广

实际应用中，多分类情况最为常见，例如文本分类，图像识别，基因分类等等。多分类问题中数据类别不平衡出现的可能性要远远高于二分类，还有就是，常见的多分类算法，例如“全对全”，“一对全”，“纠错编码”等等的方法，是基于把多分类分解为多个二分类问题来解决，这时候负类数目为其他所有类别的数目之和，因此数据不平衡更加明显。

6.1 多分类的损失函数

假设现在的反馈是 *K* 个分类，即$y \in \{1,2,\dots,K\}$，且$K \geq 2$ 。 因为要求预测模型对于类别的预测一致： 有

$$F_k(\mathbf{x}) = \sum_{h_j \in \mathcal{H}} \beta_j^{(k)} h_j(\mathbf{x})$$

其中显然的预测规则是在 *x* 处有 $\arg\max_k F_k(\mathbf{x})$。

这样对于每个样本就会对应一个 *K* 维的 ***F(x)***输出向量。若真实值 $y = c_k$，我们定义间隔向量（*K-1* 维）如下：

$$\boldsymbol{m}(y, f_1, \dots, f_K) = (f_y - f_1, \dots, f_y - f_{y-1}, f_y - f_{y+1}, \dots, f_y - f_K) \quad (26)$$

而我们的损失函数的输入也是对应 *K*-1 维的间隔向量：

$$L\big(\boldsymbol{m}(y, f_1, \dots, f_K)\big) = L(f_y - f_1, \dots, f_y - f_{y-1}, f_y - f_{y+1}, \dots, f_y - f_K) \quad (27)$$

下面是几个常见的多分类损失函数：

Logistic 回归多分类损失函数[50]：

$$L_{logistic}\left(\boldsymbol{m}(y, f_1, \dots, f_K)\right) = log\left(e^{f_1 - f_y} + \dots + e^{f_{y-1} - f_y} + 1 + e^{f_{y+1} - f_y} + \dots + e^{f_K - f_y}\right)$$

SVM 多分类损失函数[57]：

$$L_{SVM}\left(\boldsymbol{m}(y, f_1, \dots, f_K)\right) = \sum_{j=1}^{K-1} max(1 - m_j, 0)$$

其中m_j是间隔向量 ***m*** 的第 *j* 个分量。其函数图像如图 21

多分类下的 0-1 损失函数：

$$L_{0-1}(\boldsymbol{m}(y,f_1,\dots,f_K))=\begin{cases}1, & if\ \ exit\, m_j\le 0\\ 0, & else\end{cases}$$

也就是说只要存在错分，损失值就为 1。

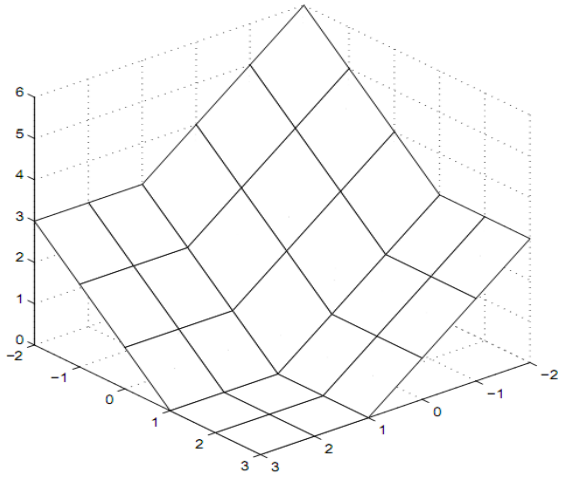


图 21　多分类 SVM 损失函数

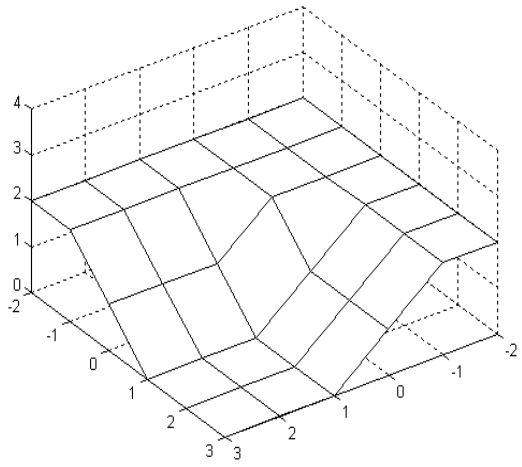


图 22　多分类一次滑梯损失函数

6.2 多分类下的“不平衡不敏感”

多分类下，“不平衡不敏感”损失函数的定义同样适用，无需修正，即其最优解集应该是“0-1 损失”下的最优解集的子集，但是最优解 $\boldsymbol{w}$ 的向量形式已经变为 K 个最优解向量$\{\mathbf{w}_1,\mathbf{w}_2,\dots,\mathbf{w}_K\}$，其中 K 为分类数目。同样，为了方便地设计出这类“不平衡不敏感”损失函数，在多分类情况，我也相应提出其充分条件：

3. 条件一：

$$\forall k\ge 1,\qquad L(\boldsymbol{k}\circ\boldsymbol{m})\le L(\boldsymbol{m})$$

(28)

4. 条件二:

$$\exists \boldsymbol{k_0}\ge 1,\forall \boldsymbol{k}\ge \boldsymbol{k_0},\exists a>0,b\in\mathbb{R}$$

$$L(\boldsymbol{k}\circ\boldsymbol{m})=aL_{0-1}(\boldsymbol{m})+b$$

(29)

注意到多分类下，损失函数的变量是间隔向量 $\boldsymbol{m}$，而$\boldsymbol{k}>1$，表示向量 $\boldsymbol{k}$ 的每个元素都大于或等于 1，另外$\boldsymbol{k}\circ\boldsymbol{m}$表示是元素与元素相乘，结果是一个向量。

定理 6.1：如果一个损失函数 L 满足上面两个条件，则它是“不平衡不敏感损失函数”。

定理 6.1 的证明与定理 5.1 非常相似，主要思路也是先证明一个类似引理 5.2 的引理：

引理 6.2：若 L 为满足条件(28)(29)的损失函数，则F_L的最优解$\{\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, \ldots, \boldsymbol{w}_K^*\}$必须满足，对于任意样本下标 i，假设$y_i = k$，有：

$$L\begin{pmatrix}(\boldsymbol{w}_k^* - \boldsymbol{w}_1^*)^T\boldsymbol{x}_i, \ldots, (\boldsymbol{w}_k^* - \boldsymbol{w}_{k-1}^*)^T\boldsymbol{x}_i, (\boldsymbol{w}_k^* - \boldsymbol{w}_{k+1}^*)^T\boldsymbol{x}_i, \\ \ldots, (\boldsymbol{w}_k^* - \boldsymbol{w}_K^*)^T\boldsymbol{x}_i\end{pmatrix}$$

$$= aL_{0-1}\begin{pmatrix}(\boldsymbol{w}_k^* - \boldsymbol{w}_1^*)^T\boldsymbol{x}_i, \ldots, (\boldsymbol{w}_k^* - \boldsymbol{w}_{k-1}^*)^T\boldsymbol{x}_i, \\ (\boldsymbol{w}_k^* - \boldsymbol{w}_{k+1}^*)^T\boldsymbol{x}_i, \ldots, (\boldsymbol{w}_k^* - \boldsymbol{w}_K^*)^T\boldsymbol{x}_i\end{pmatrix} \quad + b$$

证明思路先固定 *K-2* 个间隔分量，在满足条件(29)下，若一个间隔分量增大导致损失函数值下降的话就会与最优性矛盾，同理，所有分量的增大都不能导致损失值的变化，这样的损失函数肯定为 0-1 损失函数正仿射。类似地，我们很容易仿照定理 5.1 的证明来证明定理 6.2。

根据上面的充分条件，我们设计出“滑梯损失函数”的多分类版本：

$$L_{Slide}(\boldsymbol{m}) = \begin{cases}\sum_{j\in[1,K-1]} (max(1-m_j, 0))^{\,p}, & \text{若对所有} j \text{ 都有 } m_j \geq 0 \\ K-1, & \text{否则}\end{cases}$$

其中 p 为正数。当 p=1 时，其函数图像如图 22.

类似二分类时候的(22)式，参考[57]的多分类 SVM 的规划表述，可以得到多分类下一次滑梯损失函数的规划表述如下：

$$\min_{\{\boldsymbol{w}_i\}} \min_{0\leq\eta_{ik}\leq 1} \frac{1}{n}\sum_{i=1}^{n}\sum_{k\neq y_i}(\varepsilon_{ik} + 1 - \eta_{ik}) + \lambda\sum_{i}^{K}\|\boldsymbol{w}_i\|^2$$

且满足: $\varepsilon_{ik} \geq \eta_{ik}(1 - \boldsymbol{w}_{y_i}{}^T\boldsymbol{x}_i + \boldsymbol{w}_k{}^T\boldsymbol{x}_i)$，$\varepsilon_{ik} \geq 0$

$$i = 1, \cdots, n \qquad k \neq y_i$$

(30)

在给出多分类代价矩阵 C 下，C_{ij} 表示把第 i 类错分为第 j 类的代价，则我们可以得到包含代价敏感的规划表述：

$$\min_{\{\boldsymbol{w}_i\}} \min_{0\leq\eta_{ik}\leq 1} \frac{1}{n}\sum_{i=1}^{n}\sum_{k\neq y_i} C_{y_i k}(\varepsilon_{ik} + 1 - \eta_{ik}) + \lambda\sum_{i}^{K}\|\boldsymbol{w}_i\|^2$$

且满足: $\varepsilon_{ik} \geq \eta_{ik}(1 - \boldsymbol{w}_{y_i}{}^T\boldsymbol{x}_i + \boldsymbol{w}_k{}^T\boldsymbol{x}_i)$，$\varepsilon_{ik} \geq 0$

$$i = 1, \cdots, n \qquad k \neq y_i$$

(31)

跟二分类一样，上面的优化是非凸优化，传统的凸优化技术也不适用，鉴于本人水平有限，暂时无法提供较好解决算法。

第七章　采样不平衡

7.1 采样不平衡的定义

传统的数据不平衡可以看成是不同类别的先验分布大小相差很大，但在实际应用中，有时候即使类别先验分布相差不大，而由于采样的代价不同或者标签的代价不同，使得标签样本中类别数目相差较大，这样的情况被称为“采样不平衡”。采样不平衡与传统数据不平衡相同之处是已经标签样本中都存在类别不平衡现象，而与传统数据不平衡不同的是已标签样本的类别分布与待标签样本中的类别分布严重不同。

采样不平衡的实际例子，如病毒样本检测，若检测类别 A 代价较高，类别 B 的代价较低，那么采样中 A 的比例可能远少于 B，但实际分布中 A 的比例可能大于 B；又如市场调查，如果调查地点在商场，那么对于针对一般社会群体的预测就会产生训练样本分布与实际分布不同的情况，也可能产生采样不平衡的情况。

7.2 重采样方法

如果我们知道实际的类别比例是多少，我们可以实施重新采样的方法，令训练集的类别分布近似实际分布。这个过程可以完全等价与把其看做代价敏感情况来处理，因为对于小类来说，重新采样相当于把小类样本以一定概率复制，这样分类器原先错分一个小类样本，现在变成错分多个小类样本，代价升高，类似，重新采样把大类的样本以一定概率忽略，使其错分代价降低。事实上，现在代价敏感问题大多数算法都是基于重采样的[62][63][46]。

7.3 利用无标签数据

值得一提的是，半监督学习是提升采样不平衡下分类性能的一个不错方法。可能会产生这样疑问：在传统数据不平衡的时候利用无标签数据不也可以改善分类性能吗。实际上，不是任何时候无标签样本都可以起到作用的，甚至起反作用。正则化框架说明，最优化目标是主观误差与平滑项的折中。 而因为无标签数据不能判断其损失值，因此它起作用的地方只能是平滑项。而所谓的平滑可以这样理解，就是在某个度量下，距离近的预测值应该接近[16][15] 。换句话说，只有无标签样本与某个已标签样本足够近，平滑才起作用。在

传统的数据不平衡中，因为小类在无标签样本仍然是小类，所以很可能已标签的小类样本的周围的无标签近邻不是同类，即近邻的估计很难正确，这样的平滑结果是反效果。因为基于“不平衡不敏感”损失函数的半监督算法还没有设计出来，而基于 Tikhonov 正则化框架下的半监督算法可以参考采用 Belkin 的 LapSVM[14]。

总结与讨论

数据不平衡一直被认为是影响分类器性能的一个重要原因，很多学者试图通过重采样，组合方法，改变评价指标等等方法来研究数据不平衡问题。而虽然这些方法可能在某种程度上说明了那些算法可以较好地处理不平衡数据，但它们都不能告诉人们为何有些算法会在数据不平衡情况下会导致分类性能下降，或者说为何有些时候性能不会下降。另外，传统的研究方法很容易把我们目标模糊化，例如应用不同的评价指标去做模型选择，但正如第三章介绍的代理损失函数，现有大部分算法的优化目标是来源于 0-1 损失的，用别的评价指标去选择用不同途径优化 0-1 损失的算法这个做法看起来就比较曲折。

而本文从 Tikhonov 正则化框架下机器学习问题转化为最优化问题的角度入手，目标明确，即分析在数据不平衡下采用不同损失函数对优化问题的最优解的影响。为此，我通过对凸损失函数分三种情况讨论，引进广义指数损失函数和广义 hinge 损失函数在人工数据上举例说明，进一步指出相当于 0-1 损失下的分类器性能下降的根本原因是实施凸优化的结果，换句话说就是我们优化的目标跟我们评价性能的目标不相符程度因为数据不平衡而更加显著。那么除了 0-1 损失自身外有无其他损失函数可以与其优化结果相符呢？我提出了“不平衡不敏感损失函数”的概念，旨在定义出一类损失函数其优化目标无论数据如何变化都符合 0-1 损失的优化目标，并且比 0-1 损失具有更加适合优化的性质，如连续性、局部凸、间隔最大化特性等。同时，本文也给出并证明了“不平衡不敏感损失函数”的充分条件。“滑梯函数”就是这样一类“不平衡不敏感损失函数”，当然这类函数对应的最优化问题成为非凸问题，一般方法无法确保解的最优性。根据已有的研究，我尝试了随机梯度下降，即 GA 方法，和半定规划近似，即 SDP 方法来解决这个最优化问题。可惜的是，实验证明[59]提出的 SDP 方法的优化目标已经不是原问题的一次滑梯损失（鲁棒 hinge 损失）；而 GA 方法虽然符合优化目标，但很容易收敛于局部最优解并导致过拟合现象比较严重，无法体现间隔最大化的特性。因此，目前两种方法可以说都不能够实际解决非凸优化问题。

另外，“两月亮”数据集实验表明广义指数损失函数的泛化性能随指数下降而下降，而“pima-indians-diabetes”数据集实验表明广义 hinge 损失函数的泛化性随指数增大而下降，这都与第四章的分析吻合。然而，在实验中发现，在不平衡数据集分类性能差的原因可能与数据是否平衡无关，是因为假设空间选择问题。还有就是，虽然第四章指出了凸损失函数可能因为数据不平衡导致的问题，但至于是否真的出现是取决于数据分布，某些情况下一些凸函数的性质在特定的数据分布上有积极作用。

在第七章，我给出了多分类问题的形式化描述和满足“不平衡不敏感”在多分类下充分条件，以及“滑梯函数”的多分类形式和规划表述。第八章，我提出了“采样不平衡”的概念，并指出其本质与代价敏感问题一致，以及这个情况下应用半监督学习的好处。

除了实际代价指标为 0-1 损失外，文章中的讨论完全可以移植到对类别错分代价敏感的指标上。而更加应该指出的是，即使实际的评价指标非前面提及到两者，例如用召回率，

AUC 等指标，这样放到最优化问题上讨论还是十分有益的，并要注意目标算法使用的代理损失函数与实际评价指标的差距。

下面是一些我觉得需要进一步讨论和研究工作的话题：

- 比较损失函数：在实验中，发现广义指数损失函数类在不同情况下都有较好的性能，当然对应的参数值可能每次有点不同。我们能否每次训练多个参数的广义指数损失函数的分类器来通过比较选择一个较适合该问题的分类器呢？
- 凸与非凸的组合：我们知道凸函数之和肯定是凸函数，非凸函数之和肯定为非凸函数，而凸函数和非凸函数之和有可能是凸函数，能否设计这样的损失函数组合使得原来的非凸优化问题可以用凸优化问题来近似呢？
- 其他正则化框架：除了 Tikhonov 正则化框架外，其他正则化框架例如 Lp 正则化也可以直接得到与第五章一样的结果，未来可以考虑不同正则化框架是否会对“不平衡不敏感”损失函数的性质产生影响。

附录

下面介绍正文第四章中提及到的三种算法的实现。无论哪一种算法，最终分类函数的形式都为：

$$f(x)=\sum_{i=1}^{n}\alpha_i\times k(x_i,x)-bias$$

因此下面主要解析如何求得系数向量$\boldsymbol{\alpha}$，和偏置 *bias*。

带 bias 的 RLSC 算法

$$\boldsymbol{\alpha}=(\mathbf{K^2}-\mathbf{K}\cdot\mathbf{1}\mathbf{1}^{\mathrm{T}}\cdot\mathrm{K}+\lambda\mathrm{K})^{-1}(\mathrm{K}\cdot\mathbf{1}-\mathrm{K}\cdot\mathbf{1}\mathbf{1}^{\mathrm{T}}\cdot\mathrm{Y})$$

$$\mathrm{Bias}=\mathbf{1}^{\mathrm{T}}\cdot(\mathrm{K}\boldsymbol{\alpha}-\mathrm{Y})/n;$$

其中 n 为训练样本数目，K 为$n\times n$的核矩阵，$\mathbf{1}$ 为全 1 的$n\times 1$向量，Y 为对角阵，对角线上的元素Y_{ii}对应第 i 个样本的标签值，λ为正则化系数。

基于广义指数损失函数的 SVM 算法

因为基于广义指数损失的优化问题是一个凸优化问题，因此根据第二章，只需要给出对应的目标函数，一阶偏导向量函数和二阶偏导矩阵函数就可以通过 matlab 的“fminunc”接口求解。因此下面给出它们的表述：

目标函数：

$$F=\sum_{i=1}^{n}\mathrm{e}^{-p(y_i(K_i\boldsymbol{\alpha}-bias))}+\boldsymbol{\lambda\alpha}^{\mathrm{T}}K\boldsymbol{\alpha}$$

一阶偏导：

$$\frac{\partial F}{\partial\alpha_i}=2\lambda K_i\boldsymbol{\alpha}-p\sum_{r=1}^{n}y_rK_{ri}\,e^{-py_r(K_r\boldsymbol{\alpha}-bias)}$$

$$\frac{\partial F}{\partial bias}=p\sum_{r=1}^{n}y_r\,e^{-py_r(K_r\boldsymbol{\alpha}-bias)}$$

二阶偏导：

$$\frac{\partial^2 F}{\partial \alpha_i \, \partial \alpha_j} = 2\lambda K_{ij} \; + p^2 \sum_{r=1}^{n} K_{ri} K_{rj} \, e^{-p y_r (K_r \boldsymbol{\alpha} - bias)}$$

$$\frac{\partial^2 F}{\partial \alpha_i \, \partial bias} = -p^2 \sum_{r=1}^{n} K_{ri} \, e^{-p y_r (K_r \boldsymbol{\alpha} - bias)}$$

$$\frac{\partial^2 F}{\partial^2 bias} = p^2 \sum_{r=1}^{n} e^{-p y_r (K_r \boldsymbol{\alpha} - bias)}$$

其中 n 为训练样本数目，K 为$n \times n$的核矩阵, K_{ij} 为矩阵 K 的第 i 行第 j 列的元素，而 K_i表述第i个行向量, y_i对应第 i 个样本的标签值，λ为正则化系数。

基于广义 Hinge 损失函数的 SVM 算法

因为基于广义 hinge 损失的优化问题是一个凸优化问题，因此根据第二章，只需要给出对应的目标函数，一阶偏向量函数，和二阶偏导矩阵函数就可以通过 matlab 的“fminunc”接口求解。因此下面给出它们的表述：

目标函数：

$$F = \sum_{i=1}^{n} ((1 - y_i (K_i \boldsymbol{\alpha} - bias))_+)^p + \lambda \boldsymbol{\alpha}^{\mathrm{T}} K \boldsymbol{\alpha}$$

一阶偏导：

$$\frac{\partial F}{\partial \alpha_i} = 2\lambda K_i \boldsymbol{\alpha} - p \sum_{r=1}^{n} y_r K_{ri} ((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+)^p \cdot ((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+$$

$$\frac{\partial F}{\partial bias} = \; p \sum_{r=1}^{n} y_r ((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+)^p \cdot ((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+$$

二阶偏导：

$$\frac{\partial^2 F}{\partial \alpha_i \, \partial \alpha_j} = 2\lambda K_{ij} + p^2 \sum_{r=1}^{n} K_{ri} K_{rj} ((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+)^p \cdot ((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+$$

$$\frac{\partial^2 F}{\partial \alpha_i \, \partial bias} = -p^2 \sum_{r=1}^{n} K_{ri} \left((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+\right)^p \cdot \left((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+\right.$$

$$\frac{\partial^2 F}{\partial^2 bias} = p^2 \sum_{r=1}^{n} \left((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+\right)^p \cdot \left((1 - y_r (K_r \boldsymbol{\alpha} - bias))_+\right.$$

其中 n 为训练样本数目，K 为$n \times n$的核矩阵, K_{ij} 为矩阵 K 的第 i 行第 j 列的元素，而 K_i表述第i个行向量, y_i对应第 i 个样本的标签值，λ为正则化系数。另外，函数$(\mathrm{x})_+$表示为 x 的正部与$max(x, 0)$等价。

结束语

本次毕业论文从选题到结题的时间不足三个月，但我对数据不平衡问题的思考断断续续已经超过三年时间了。一开始的时候，我的知识是散开一片一片的，通过研究别人的论文来了解这个领域的研究状况，但由于自身知识积累不足，一直都无法把别人的研究成果与机器学习的本质联系起来。幸好，大四的下学期是一段比自由的时间，在毕业论文最终定题之前，我抛开一起杂念自学了实分析和泛函分析等数学基础课程，这数学基础为我后来自学 MIT OCW 的“统计学习”网上课程铺路。在这些学习后，我领会了机器学习中的各种问题在泛函分析或概率测度下的更本质的表述，这让我觉得是以前一直比较模糊的问题弄清楚的时机来了，因而有了这个论文的选题。

这篇论文的研究思路是先给所谓“不平衡问题”下一个更准确的定义，即在给定代价指标（这里是 0-1 损失）下分类器性能随不平衡程度加重而下降；而之前普遍认为的所谓“不平衡问题”就是在数据不平衡情况下传统分类器的分类性能下降的问题，明显，先前的这种定义实在太过模糊了。接着，我希望向大家解释我们的传统分类通常使用的是“代理损失函数”，需要代理的就是“0-1”损失，而“不平衡问题”的出现恰恰反映出这些“代理”的失职。这里也回应了很多学者热衷于用其他的指标（例如召回率，AUC 值）来在不平衡情况下做模型选择，而他们似乎没发现他们选择得到的“代理”的原本意义。“代理”的原本意义就是为了实施凸优化而存在的，因为传统算法设计者绝大部分都喜欢凸优化而讨厌非凸优化（神经网络除外）。当然，这篇论文最重要的成果就是指出导致“不平衡问题”的本质原因是实施凸优化的结果。既然凸优化存在问题，我继续思考怎样“非凸优化”才不会有问题，因此有了“不平衡不敏感损失函数”的出现。最后我对多分类下的“不平衡不敏感”作了些推广并讨论了“不平衡问题”的小变种“采样不平衡”。

整个研究的过程远没有设想的一帆风顺，先是非凸优化技术不完善在考虑如何降低计算代价上花了很多时间，而后来在实验中又遇到一些不可控制干扰因素使现在的实验无法完全支持提出的理论。在这里，我要感谢从大二开始一直指导我机器学习方面陈琼老师，她给我很自由空间去学习和研究，并一直鼓励我。同时，我也特别感谢 04 应数的齐飞同学，他在我数学的学习上给我很多帮助和鼓励。

最后向所有给予我关心和帮助的师长、亲人、同学表示衷心的感谢。

参考文献

[1]. 峰, 凌晓. (2007), '代价敏感分类器的比较研究', *计算机学报* **30**(008), 1203--1212.

[2]. 姚程宽 (2007), 'SVM 在不平衡样本集中的应用研究', *计算机与数字工程*, 10.

[3]. 兴, 吴洪. (2006), '适用于不平衡样本数据处理的支持向量机方法', *电子学报* **34**(B12), 2395--2398.

[4]. 罗兵 & 余光柱 (2007), '不平衡类分布下多分类问题的提升算法', *長江大學學報 (自科版)* **4**(2), 50--54.

[5]. 杨明 & 杨萍 (2007), '一种面向不平衡分类数据的核求解算法', *控制与决策*, 06.

[6]. 林智勇; 郝志峰 & 杨晓伟 (2008), '不平衡数据分类的研究现状', *计算机应用研究*, 02.

[7]. 陶晓燕; 姬红兵 & 马志强 (2007), '基于样本分布不平衡的近似支持向量机', *计算机科学*, 05.

[8]. 彬, 李建.; 郑辉 & 霞, 牛忠. (2007), 'AdaBoost 算法中的数据类别不平衡现象', *电信 技术 研究* (011), 11--17.

[9]. 郑恩辉; 李平 & 宋执环 (2005), '不平衡数据知识挖掘: 类分布对支持向量机分类的影响', *信息与控制* **34**(6), 703--708.

[10]. 辉, 郑恩.; 许宏; 李平 & 环, 宋执. (2006), '基于 ε-SVM 的不平衡数据挖掘研究', *浙江大学学报: 工学版* **40**(010), 1682--1687.

[11]. 缪志敏; 胡谷雨; 丁力; 赵陆文 & 潘志松 (2008), 'SVDD 在类别不平衡学习中的应用', *应用科学学报*, 01.

[12]. A. Asuncion, D. N. (2007), 'UCI Machine Learning Repository', Technical report, University of California, Irvine, School of Information and Computer Sciences.

[13]. Akbani, R.; Kwek, S. & Japkowicz, N. (2004), 'Applying support vector machines to imbalanced datasets', *Proceedings of the 15th European Conference on Machine Learning (ECML)*, 39--50.

[14]. Belkin, M. & Niyogi, P. (2004), 'Semi-Supervised Learning on Riemannian Manifolds', *Machine Learning* **56**(1), 209--239.

[15]. Belkin, M. & Niyogi, P. (2002), 'Laplacian eigenmaps and spectral techniques for embedding and clustering', *Advances in Neural Information Processing Systems* **14**, 585--591.

[16]. Belkin, M.; Niyogi, P. & Sindhwani, V. (2005), 'On manifold 正则化', *Proceedings of the Tenth*

International Workshop on Artificial Intelligence and Statistics (AISTAT 2005).

[17]. Bennett, K. P. & Parrado-Hernández, E. (2006), 'The Interplay of Optimization and Machine Learning Research', *J. Mach. Learn. Res.* **7**, 1265--1281.

[18]. Bousquet, O.; Elisseeff, A. & Ron, D. (2002), 'Stability and Generalization', *Journal of Machine Learning Research* **2**(3), 499--526.

[19]. Boyd, S. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

[20]. Buhlmann, P. & Hothorn, T. (2008), 'Boosting algorithms: 正则化, prediction and model fitting', *Statistical Science*.

[21]. Chawla, N.; Japkowicz, N. & Kotcz, A. (2004), 'Editorial: special issue on learning from imbalanced data sets', *ACM SIGKDD Explorations Newsletter* **6**(1), 1--6.

[22]. Cohen, G.; Hilario, M. & Pellegrini, C. (2004), 'One-class support vector machines with a conformal kernel. a case study in handling class imbalance', *Structural, Syntactic, and Statistical Pattern Recognition*, 850--858.

[23]. Collobert, R.; Sinz, F.; Weston, J. & Bottou, L. (2006), 'Trading convexity for scalability', *Proceedings of the 23rd international conference on Machine learning*, 201--208.

[24]. Cortes, C. & Vapnik, V. (1995), 'Support-Vector Networks', *Machine Learning* **20**(3), 273-297.

[25]. CUCKER, F. & SMALE, S. (), 'ON THE MATHEMATICAL FOUNDATIONS OF LEARNING', *AMERICAN MATHEMATICAL SOCIETY* **39**(1), 1--49.

[26]. Dai, S. & Zhang, Y. (2003), 'Color image segmentation with watershed on color histogram and Markov random fields', *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on* **1**.

[27]. De Bie, T. (2007), 'Deploying sdp for machine learning', *Proceedings of the fifteenth European Symposium on Artificial Neural Networks*.

[28]. Donoho, D. & Grimes, C. (2003), 'Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data', *Proceedings of the National Academy of Sciences of the United States of America* **100**(10), 5591.

[29]. Drummond, C. & Holte, R. (2003), 'C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling', *Workshop on Learning from Imbalanced Data Sets II*.

[30]. Elazmeh, W.; Japkowicz, N. & Matwin, S. (2006), 'Evaluating Misclassifications in Imbalanced Data', *LECTURE NOTES IN COMPUTER SCIENCE* **4212**, 126.

[31]. Estabrooks, A. & Japkowicz, N. (2001), 'A Mixture-of-Experts Framework for Learning from Imbalanced Data Sets', *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, 34--43.

[32]. Estabrooks, A.; Jo, T. & Japkowicz, N. (2004), 'A Multiple Resampling Method for Learning from Imbalanced Data Sets', *Computational Intelligence* **20**(1), 18--36.

[33]. Evgeniou, T.; Pontil, M. & Poggio, T. (2000), '正则化 Networks and Support Vector Machines', *Advances in Computational Mathematics* **13**(1), 1--50.

[34]. Garcia, V.; Alejo, R.; Sánchez, J.; Sotoca, J. & Mollineda, R. (2006), 'Combined Effects of Class Imbalance and Class Overlap on Instance-Based Classification', *LECTURE NOTES IN COMPUTER SCIENCE* **4224**, 371.

[35]. Gдrtner, T. (2003), 'A survey of kernels for structured data', *SIGKDD Explorations* **5**(1), 49--58.

[36]. Har-Peled, S.; Roth, D. & Zimak, D. (), 'Constraint Classification: A New Approach to Multiclass Classification', *Urbana* **51**, 61801.

[37]. Japkowicz, N. (2003), 'Class imbalances: are we focusing on the right issue', *Workshop on Learning from Imbalanced Data Sets II*.

[38]. Japkowicz, N. (2002), 'The class imbalance problem: A systematic study', *Intelligent Data Analysis* **6**(5), 429--449.

[39]. Japkowicz, N. (2001), 'Concept-Learning in the Presence of Between-Class and Within-Class Imbalances', *Advances in Artificial Intelligence: 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, Ai 2001, Ottawa, Canada, June 7-9, 2001: Proceedings*.

[40]. Japkowicz, N. (2000), 'The class imbalance problem: Significance and strategies', *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI'2000)* **1**, 111--117.

[41]. Japkowicz, N. (2000), 'Learning from imbalanced data sets: a comparison of various strategies', *AAAI Workshop on Learning from Imbalanced Data Sets. Tech Rep. WS-00-05, Menlo Park, CA: AAAI Press*.

[42]. Jordan, M. (1998), 'Learning in Graphical Models', .

[43]. LeCun, Y.; Chopra, S.; Hadsell, R.; Marc'Aurelio, R. & Huang, F. (2006), A Tutorial on Energy-Based Learning, *in* G. Bakir; T. Hofman; B. Schцlkopf; A. Smola & B. Taskar, ed.,'Predicting Structured Data',

MIT Press, .

[44]. Lee, H. & Cho, S. (2006), 'The Novelty Detection Approach for Different Degrees of Class Imbalance', *LECTURE NOTES IN COMPUTER SCIENCE* **4233**, 21.

[45]. Li, W.; Leung, K. & Lee, K. (2007), 'Generalizing the Bias Term of Support Vector Machines', *Proceedings of the International Conference on Artificial Intelligence*.

[46]. Liu, X.; Wu, J. & Zhou, Z. (2006), 'Exploratory Under-Sampling for Class-Imbalance Learning', *Proceedings of the Sixth International Conference on Data Mining*, 965--969.

[47]. Liu, X. & Zhou, Z. (2006), 'The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study', *Proceedings of the Sixth International Conference on Data Mining*, 970--974.

[48]. Nickerson, A.; Japkowicz, N. & Milios, E. (2001), 'Using unsupervised learning to guide re-sampling in imbalanced data sets', *Proceedings of the Eighth International Workshop on AI and Statitsics*, 261--265.

[49]. Orriols, A. & Bernadó-Mansilla, E. (2005), 'The class imbalance problem in learning classifier systems: a preliminary study', *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, 74--78.

[50]. Rosset, S.; Zhu, J. & Hastie, T. (), 'Margin Maximizing Loss Functions', *criterion* **3**(4), 0.

[51]. Sarwar, B.; Karypis, G.; Konstan, J. & Reidl, J. (2001), 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th international conference on World Wide Web*, 285--295.

[52]. Smale, S. & Yao, Y. (2006), 'Online Learning Algorithms', *Foundations of Computational Mathematics* **6**(2), 145--170.

[53]. Tenenbaum, J.; Silva, V. & Langford, J. (2000), 'A Global Geometric Framework for Nonlinear Dimensionality Reduction', *Science* **290**(5500), 2319--2323.

[54]. Toh, K.; Tütüncü, R. & Todd, M. (2006), 'On the implementation and usage of SDPT3--a MATLAB software package for semidefinitequadratic-linear programming, version 4.0', *Department of Mathematics, National University of Singapore, Tech. Rep., Jul*.

[55]. Trogkanis, N. & Paliouras, G. (), 'TPN 2: Using positive-only learning to deal with the heterogeneity of labeled and unlabeled data', *The Discovery Challenge Workshop*.

[56]. Vapnik, V. (2000), *The Nature of Statistical Learning Theory*, Springer.

[57]. Weston, J. & Watkins, C. (1998), 'Multi-class support vector machines', .

[58]. Wu, Y. & Liu, Y. (2007), 'Robust truncated-hinge-loss support vector machines', *Journal of the American Statistical Association* **102**, 974--983.

[59]. Xu, L.; Crammer, K. & Schuurmans, D. (2006), 'Robust support vector machine training via convex outlier ablation', *Proceedings National Conference on Artificial Intelligence (AAAI-06)*.

[60]. Yu, Y.; Zhan, D.; Liu, X.; Li, M. & Zhou, Z. (2007), '. Predicting future customers via ensembling gradually expanded trees. Special Issue on the PAKDD 2006 Data Mining Competition. International Journal of Data Warehousing and Mining', *3 (2)*, 12--21.

[61]. Zhou, D. (2003), 'Capacity of reproducing kernel spaces in learning theory', *Information Theory, IEEE Transactions on* **49**(7), 1743--1752.

[62]. Zhou, Z. & Liu, X. (), 'On Multi-Class Cost-Sensitive Learning', *Proceeding of the 21st National Conference on Artificial Intelligence*, 567--572.

[63]. Zhou, Z. & Liu, X. (2006), 'Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem', *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 63--77.

[64]. Zhou, Z.; Zhan, D. & Yang, Q. (), 'Semi-Supervised Learning with Very Few Labeled Training Examples', *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 675--680.

[65]. http://videolectures.net/eml07_lecun_wia/

[66]. http://sedumi.mcmaster.ca/

[67]. http://dollar.biz.uiowa.edu/~burer/software/SDPLR/