2020 年 春 季学期研究生课程考核

阅读报告

考 核 科 目：组合优化与凸优化

学生所在院（系）：计算机科学与技术学院

学 生 所 在 学 科：计算机科学与技术

学 生 姓 名：于晟健

学 号：19S003037

学 生 类 别：全日制学术型硕士研究生

考 核 结 果　　阅 卷 人

Active Learning for Cost-Sensitive Classification*

1 代价敏感分类问题

分类问题一直是机器学习和数据挖掘的重要领域，传统的分类算法都旨在提高分类的准确性，最大限度的减少分类误差，这都是基于所有类的误分类代价相等。然而，当不同类别之间代价不同时，代价敏感分类就显得非常重要。

例如，在医疗领域重要疾病确诊时，将一个有病的人判定成健康比将一个健康的人判定成有病，要严重很多，有病误诊成没兵可能会耽误就医时机，严重时可能危害生命健康，没病误诊为有病仅仅就是白担心一场，并不会有太大危害；同样，在垃圾邮件检测系统当中，如果将垃圾邮件误分类为普通邮件，用户可以很容易识别出来进行删除处理，然而将普通邮件误分类为垃圾邮件丢入垃圾箱中，用户可能就看不到一些重要信息，就会耽误事清。因此，许多重要领域我们需要注重代价，运用代价敏感的算法来处理问题。

正因代价敏感分类不再仅仅以较高分类准确率为目标，在分类的过程中往往还需要考虑错误分类代价，测试代价、计算代价等等。代价能从很多方面对目标进行判断，*Turney* 对代价作了详细的介绍[1]，大致划分为以下几种类型：

（1）错误分类代价：错误分类产生的代价。

（2）测试代价：为了检测某些属性而产生的代价。譬如决策树选取分类点进行属性检测产生的代价。

（3）获取样本的代价：有时候为了获取训练样本需要付出一定的代价，训练样本越多获得的分类模型效果越好，但会产生获取样本的代价。

（4）指导代价：为了获得实例的类别付出的代价。一般为了得到实例的类标号需要付出非常大的代价。

（5）副作用带来的代价：对具体流程的干涉会对生成的结果产生一定的代价。

（6）人机交互代价：个人使用算法时付出的代价。主要包含探求最佳参数和数据预处理等过程产生的代价。

（7）干预代价：为了满足生产目标，需要对生产过程进行人干涉，譬如为了改变属性信息产生的代价。

（8）不稳定性的代价：如果利用同样方法得到的数据集进行训练获得的分类模型之间差异比较大，这样的分类模型就是不稳定的，会产生一定的代价。

代价敏感学习是很大的研究方向，它包含很多子研究范围，并且因为代价的多样性，使代价敏感的学习研究变得相当复杂困难。此外，代价可能随时间不同而有所不同，这使得代价敏感学习问题的研究变得更加复杂。

Krishnamurthy A, Agarwal A, Huang T K, et al. Active Learning for Cost-Sensitive Classification. JMLR 2019.

2 文献相关工作

作者工作基于分歧的主动学习框架，该框架通常在未知背景下研究一般性假设空间[2]。已有研究完成了对于二分类的工作，而作者将其推广至代价敏感多分类（*CSMC*），即假定可以使用分类回归函数准确预测代价。

对于 *CSMC* 的一个区别是作者使用查询规则判断标签的预计代价范围，另一个主要的区别是作者使用平方损失 *oracle* 来搜索版本空间。而已有的研究工作要么明确枚举版本空间[3,4]，要么使用 0/1 损失分类法进行搜索[5,6,7,8]。尽管使用启发式方法的实际实现仍然非常有效，但是在大多数实例中，*oracle* 可以解决 *NP* 难题，因此不会直接发现有效的算法。而作者的方法使用平方损失回归算法，该算法可以通过凸优化有效地实现，并导致多项式时间算法。

除了基于分歧的方法外，许多研究都集中在二元分类中主动学习的插入式规则上，其中有人估计了类条件回归函数[9,10,11,12]。除文献[11]以外，其他研究工作都使用了平滑度假设并具有非参数的形式。取而代之，文献[11]假设已校准的替代损失和抽象的可实现函数类，这与作者的设置更相似。尽管细节有所不同，但作者的工作和这些先前的结果采用了相同的算法方法，即维护隐式版本空间并在适当定义的分歧区域中进行查询。作者的工作有两个显着差异：（1）算法在 *oracle* 计算模型中运行，仅通过平方损失最小化问题访问函数类；（2）结果适用于通用 *CSMC*，与二进制分类法表现出显着差异。

对于线性表示，文献[13,14]研究了具有分布假设的主动学习，而在线学习社区的选择性抽样框架则考虑了对抗假设[15,16,17,18]，这些方法使用专门用于线性表示的查询策略，而不能自然地推广到其他假设类别。在最坏情况下解决 *NP-hard* 优化问题的监督学习预言已被用于其他问题中，包括情境强盗[19,20]和结构化预测[21]。

此外，已有研究将平方损失回归用于估计被动 *CSMC* 的代价[22]。

最近，文献[23]介绍了 *CSMC* 的主动学习算法，该算法还使用代价范围来确定要查询的位置。通过使用回归预言机对最大和最小代价执行二进制搜索来计算代价范围，但这种计算会导致标签复杂度处于次优状态。而作者通过一种新颖的代价范围计算方法来解决这种次优问题，该方法是受乘权法用于求解线性程序的启发。这种算法上的改进还需要非常复杂的统计分析，为此可以为有界伪维的类推导新的一致 *Freedman* 型不等式，该结果可能与个人利益有关。

此外，文献[23]还引入了在线近似性以提供额外的可扩展性，并将此算法用于他们的实验。作者的实证结果使用了相同的在线近似值，并且更为全面。最后，作者根据 *Tsybakov* 的低噪声条件[24,25]得出的设置，得出了算法的泛化和标签复杂性边界。

在后续研究中，文献[26]在作者的工作基础上，使用基于回归的方法进行情境匪徒学习，该问题与 *CSMC* 的主动学习有一些相似之处。由于设置的不同，结果无法比拟，但值得讨论其技术。其维护隐式版本空间并计算每个标签的最大和最小代价进行预测，

通过 *epoching* 解决了文献[27]的次优问题。与作者的加权乘积法相比，它可以简化代价范围的计算。但是，抽签会导致标签复杂性增加一个 *log n* 因子，并且在低噪声条件下总边界为 *O*(*poly log n*)，这会产生比作者方法差的保证。

3 问题假设

作者研究了 *K* 个类的代价敏感多类分类（*CSMC*）问题，其中存在实例空间 *X*，标签空间 *Y*={1,…,*K*}和在 *X*×[0,1]^K 上的分布 *D*。如果(*x*, *c*)~*D*，则将 *c* 称为代价向量（*cost-vector*），其中 *c*(*y*)是预测 *y*∈*Y* 的代价。分类器 *h*：*X*→*Y* 的预期代价为 $E(x, c)_{\sim D}[c(h(x))]$，目标是找到具有最小预期代价的分类器。

令 *G*≜{*g*: *X*→[0,1]}表示一组回归基变量，而令 *F*≜*G*^{K} 表示一组矢量回归变量，其中 *f*∈*F* 的 y^{th} 坐标写为 *f*(· ; *y*)。考虑分类器集合 *H*≜{h_f | *f*∈*F*}，其中每个 *f* 定义一个分类器 h_f:

$$h_f(x) \triangleq \arg\min_y f(x; y) \tag{1}$$

将一组回归函数用于分类任务时，很自然地假设 *D* 下的预期代价可以通过集合中的某些函数进行预测。这激发了以下可实现性假设。

假设 1（***Realizability***）定义贝叶斯最优回归因子 *f**，有 *f**(*x*; *y*) ≜E_c[*c*(*y*)| *x*]，∀*x*∈*X* 且 *D*(*x*)> 0)，*y*∈ *Y*。假定 *f**∈*F*。

尽管 *f**总是很明确，但是请注意，代价本身可能很嘈杂。与假设相比，*H* 中零代价分类器的存在（经常在主动学习中假设）更强，而 *H* 中的 h_{f^*}的存在则较弱，但在主动学习中并未得到利用。

此外，还需要对 *G* 类的复杂性进行假设以进行统计分析。为此，假设 *G* 是具有有限伪维的 $L_\infty(X)$的紧凸子集，这是 *VC* 维度对于实值预测变量的自然扩展。

定义 1（***Pseudo-dimension***）函数类 *F*: *X*→*R* 的伪维 *Pdim*(*F*)定义为阈值函数集 H^+≜{(*x*,ξ)→{*f* (*x*)>ξ}: *f*∈*F*}⊂*X*×*R*→{0,1}。

假设 2 假定 *G* 是一个紧凸集，*Pdim*(*G*)=*d*<∞。

例如，在某些基本表示形式中的线性函数，例如 $g(x)=\sum_{i=1}^{d} w_i\phi_i(x)$，其中权重 w_i 以某个范数为界，其伪维数为 *d*。实际上，结果可以完全用覆盖数来表示，并且利用此类具有“参数”覆盖数形式（1/ε）的事实将其转换为伪维。因此，尽管为简化起见而关注参数情况，但作者结果也扩展到了具有“非参数”增长率的类（例如 *Holder* 平滑函数）。请注意，这与文献[23]的观点大相径庭，假设 *G* 是有限的。

假设 *G* 是一个紧凑的凸集，这给管理这个无限大的集带来了计算上的挑战。为了应对这一挑战，作者顺应了主动学习的趋势，即利用现有的关于监督学习的算法研究[5,6,7]，并仅通过回归预言来访问 *G*。给定重要性加权数据集$D = \{x_i, c_i, w_i\}_{i=1}^{n}$，其中 x_i∈*X*, c_i∈*R*, w_i∈R_+，*oracle* 回归计算：

$$ORACLE(D) \in arg \min_{g \in G} \sum_{i=1}^{n} w_i (g(x_i) - c_i)^2 \quad (2)$$

由于假定 G 是一个紧凑的凸集，因此它适合于标准凸优化技术，因此这没有施加其他限制。但是，在线性函数的特殊情况下，此优化只是最小二乘，可以以闭合形式进行计算。请注意，这与使用 0/1 损失最小化预言的先前工作[5,6,7]从根本上不同，后者在大多数情况下都涉及 *NP* 硬优化。

重点 2 假设 G 是凸的只是为了计算可处理性，因为它在有效实施查询策略中至关重要，而对于概括性和标签复杂性边界则不是必需的。不幸的是，最近对非凸类学习的保证[27,28]不能立即产生有效的主动学习策略。还请注意，文献[23]获得了没有凸度的有效算法，但这会产生次优的标签复杂度保证。

给定一组示例和查询的代价，通常将注意力集中在回归函数上，这些函数可以很好地预测这些代价并在给出新的示例 x 的情况下评估其预测中的不确定性。对于回归变量 $G \subset G$ 的子集，测量 x 的可能代价值的不确定性，其中：

$$\gamma(x, G) \triangleq c_+(x, G) - c_-(x, G), c_+(x, G) \triangleq \max_{g \in G} g(x), c_-(x, G) \triangleq \min_{g \in G} g(x). \quad (3)$$

对于向量回归因子 $F \subset F$，将给定 x 的标签 y 的代价范围定义为 $\gamma(x, y, F) \triangleq \gamma(x, G_F(y))$，其中 $G_F(y) \triangleq \{f(\cdot; y) \mid f \in F\}$ 是 F 对 y 的基本回归。请注意，由于假设可实现性，因此，每当 $f^* \in F$ 时，$c_+(x, G_F(y))$ 和 $c_-(x, G_F(y))$ 会提供 $E[c(y)|x]$ 的有效上下限。

为了衡量标注工作量，跟踪了即使要查询单个代价的示例数以及查询的总数中检查示例的编辑工作量很大，但每种代价所需的进一步工作最少，而每种代价需要大量的工作。形式上，将 $Q_i(y) \in \{0,1\}$ 定义为算法在第 i 个示例上查询标签 y 并进行测量的指标：

$$L_1 \triangleq \sum_{i=1}^{n} \bigvee_{y} Q_i(y), L_2 \triangleq \sum_{i=1}^{n} \sum_{y} Q_i(y) \quad (4)$$

4 代价重叠的主动学习

算法 1 中给出了作者算法的伪代码，即代价重叠主动学习（*COAL*）。给定 x 的示例，*COAL* 向 x 询问某些标签 y 的代价。通过公式(1)根据过去的数据（即版本空间）计算一组良好的回归函数，公式(2)计算这些函数可为每个 y 实现的预测范围，以及公式(3)依次查询，来选择这些代价 y 可能是最好的标签，并且具有很大的不确定性。现在，我们详细介绍每个步骤。

为了计算近似版本空间，首先找到使每个标签 y 的经验风险最小的回归函数，在第 i 轮为：

$$R_i(g; y) \triangleq \frac{1}{i-1} \sum_{j=1}^{i-1} \left(g(x_j) - c_j(y)\right)^2 Q_j(y) \quad (5)$$

Algorithm 1 Cost Overlapped Active Learning (COAL)

Input: Regressors $\mathcal{G}$, failure probability $\delta \leq 1/e$.
Set $\psi_i = 1/\sqrt{i}$, $\kappa = 3$, $\nu_n = 324(d\log(n) + \log(8Ke(d+1)n^2/\delta))$.
Set $\Delta_i = \kappa \min\{\frac{\nu_n}{i-1}, 1\}$.
for $i = 1, 2, \ldots, n$ **do**
    $g_{i,y} \leftarrow \arg\min_{g\in\mathcal{G}} \widehat{R}_i(g; y)$. (See (5)).
    Define $f_i \leftarrow \{g_{i,y}\}_{y=1}^{K}$.
    (Implicitly define) $\mathcal{G}_i(y) \leftarrow \{g \in \mathcal{G}_{i-1}(y) \mid \widehat{R}_i(g; y) \leq \widehat{R}_i(g_{i,y}; y) + \Delta_i\}$.
    Receive new example x. $Q_i(y) \leftarrow 0, \forall y \in \mathcal{Y}$.
    for every $y \in \mathcal{Y}$ **do**
        $\widehat{c_+}(y) \leftarrow \textsc{MaxCost}((x, y), \psi_i/4)$ and $\widehat{c_-}(y) \leftarrow \textsc{MinCost}((x, y), \psi_i/4)$.
    end for
    $Y' \leftarrow \{y \in \mathcal{Y} \mid \widehat{c_-}(y) \leq \min_{y'} \widehat{c_+}(y')\}$.
    if $|Y'| > 1$ **then**
        $Q_i(y) \leftarrow 1$ if $y \in Y'$ and $\widehat{c_+}(y) - \widehat{c_-}(y) > \psi_i$.
    end if
    Query costs of each y with $Q_i(y) = 1$.
end for

回想一下 $Q_j(y)$是在第 j 个示例中查询标签 y 的指标。计算最小化程序需要一个 *oracle* 调用。作者隐式地构建了第 7 行中的版本空间 $G_i(y)$，因为幸存的具有低平方损失的回归函数对经验最小化风险感到遗憾。对此后悔的容忍度是第 i 轮的Δ_i，其缩放比例类似于 $O(d/i)$，其中 d 是类别 G 的伪维。

然后，*COAL* 计算新示例 x 上版本空间 $G_i(y)$预测的最大和最小代价。因为真正的期望代价是 $f^*(x; y)$，并且正如将看到的 $f^*(\cdot; y) \in G_i(y)$，所以这些数量充当该值的置信区间。该计算由 *MaxCost* 和 *MinCost* 子程序完成，它们分别生成 $c_+(x, G_i(y))$和 $c_-(x, G_i(y))$的近似值，见公式(3)。

最后，使用预测代价，*COAL* 发出（可能为零）查询。该算法会查询具有较大代价范围的任何非主导标签，如果标签的估计最低代价小于最小最大代价（在所有其他标签中），并且代价范围是该标签之间的差异，则该标签为非主导估计的最高和最低代价。

直观地，*COAL* 会查询每个标签的代价，但不能排除其在 x 上具有最小代价的情况，只有在代价的实际价值存在足够歧义的情况下才能查询。想法是，几乎没有分歧的标签不会为进一步减少版本空间提供太多信息，因为通过构造，所有回归器都会遭受相似的平方损失。此外，由于假设 h_f的代价敏感性能仅取决于其预测的标签，因此仅需要查询可能是最好的标签。因此，不需要查询占主导地位或代价范围小的标签。

类似的查询策略已用于二进制和多类分类的先前研究中[16,17,18]，但专门用于线性表示。线性情况的主要优势在于，集合 $G_i(y)$（通常是具有相似属性的另一个集）以及最大和最小代价具有闭合形式的表达式，因此算法易于实现。然而，使用通用集合 G 和回归预言，计算这些置信区间就不那么直接了。使用 *MaxCost* 和 *MinCost* 子程序，然后在下面讨论算法的这一方面。

4.1 代价范围的有效计算

在本节中，我们描述作者的 *MaxCost* 子程序，该程序使用回归预言来近似估计由 $G_i(y)$实现的标签 y 上的最大代价，如公式(3)中所定义。最低代价的计算只需要在本节末尾讨论的少量修改。

描述算法需要一些附加的符号。令$\widetilde{\Delta}_j \triangleq \Delta_j + \hat{R}_j(g_{j,y}; y)$是定义在第 j 轮版本空间的约束右侧，其中 $g_{j,y}$ 是第 j 轮的标签 y 的 *ERM*，$\hat{R}_j(\cdot; y)$是风险函数，而Δ_j 是 *COAL* 中使用的半径。注意，可以通过单个 *oracle* 调用找到 $g_{j,y}$，因此可以有效地计算该数量。由于在 $G_i(y)$的定义中要求 $g \in G_{i-1}(y)$，所以等效表示为$G_i(y) = \bigcap_{j=1}^{i}\{g: \hat{R}_j(g; y) \leq \Delta_j\}$。作者的方法基于以下观察：给定示例 x 和第 i 轮的标签 y，找到一个预测 g 上的标签 y 的最大代价的函数 $g \in G_i(y)$等于解决最小化问题：

$$minimize_{g \in G}(g(x) - 1)^2 \ \ such\ that\ \forall 1 \leq j \leq i, \hat{R}_j(g; y) \leq \widetilde{\Delta}_j \tag{7}$$

Algorithm 2 MaxCost

Input: (x, y), tolerance TOL, (implicitly) risk functionals $\{\widehat{R}_j(\cdot; y)\}_{j=1}^{i}$.
Compute $g_{j,y} = \operatorname{argmin}_{g \in \mathcal{G}} \widehat{R}_j(g; y)$ for each j.
Let $\Delta_j = \kappa \min\{\frac{\nu_n}{j-1}, 1\}$, $\tilde{\Delta}_j = \widehat{R}_j(g_{j,y}; y) + \Delta_j$ for each j.
Initialize parameters: $c_\ell \leftarrow 0, c_h \leftarrow 1, T \leftarrow \frac{\log(i+1)(12/\Delta_i)^2}{\mathrm{TOL}^4}, \eta \leftarrow \sqrt{\log(i+1)/T}$.
while $|c_\ell - c_h| > \mathrm{TOL}^2/2$ **do**
    $c \leftarrow \frac{c_h - c_\ell}{2}$
    $\mu^{(1)} \leftarrow \mathbf{1} \in \mathbb{R}^{i+1}$. ▷ Use MW to check feasibility of Program (9).
    for $t = 1, \ldots, T$ **do**
        Use the regression oracle to find

$$g_t \leftarrow \operatorname*{argmin}_{g \in \mathcal{G}} \mu_0^{(t)}(g(x) - 1)^2 + \sum_{j=1}^{i} \mu_j^{(t)} \widehat{R}_j(g; y) \tag{6}$$

        If the objective in (6) for g_t is at least $\mu_0^{(t)} c + \sum_{j=1}^{i} \mu_j^{(t)} \tilde{\Delta}_j$, $c_\ell \leftarrow c$, go to 5.
        Update

$$\mu_j^{(t+1)} \leftarrow \mu_j^{(t)} \left(1 - \eta \frac{\tilde{\Delta}_j - \widehat{R}_j(g_t; y)}{\Delta_j + 1}\right), \quad \mu_0^{(t+1)} \leftarrow \mu_0^{(t)} \left(1 - \eta \frac{c - (g_t(x) - 1)^2}{2}\right).$$

    end for
    $c_h \leftarrow c$.
end while
Return $\widehat{c_+}(y) = 1 - \sqrt{c_\ell}$.

根据这一观察结果，作者的策略将是找到问题(7)的近似解决方案，不难看出，这也为标签 y 产生了 x 上最大预测代价的近似值。

在算法 2 中，作者展示了如何使用回归预言来有效地解决该程序。作者首先利用集合 G 的凸性，这意味着可以进一步将优化问题(7)重写为：

$$minimize_{p\in\Delta G}E_{g\sim P}[(g(x)-1)^2]\ such\ that\ \forall 1\le j\le i, E_{g\sim P}[\hat{R}_j(g;y)]\le\widetilde{\Delta}_j \quad (8)$$

根据凸度的定义，上述重写实际上是修饰为 $G=\Delta(G)$，但是结果是作者的重写导致优化变量 P 中的目标和约束均为线性。因此，有效地希望求解线性 P 中的程序，作者的计算工具是对集合 G 的回归预言。为此，作者创建了一系列可行性问题，在这些问题中，作者反复猜测问题的最佳目标值(8，然后检查是否确实存在分布 P 满足所有约束并给出假定的目标值。也就是说，作者检查

$$?\exists P\in\Delta(G) such\ that\ E_{g\sim P}[(g(x)-1)^2]\le c\ and\ \forall 1\le j\le i, E_{g\sim P}[\hat{R}_j(g;y)]\le\widetilde{\Delta}_j \quad (9)$$

如果找到这样的解决方案，则会增加猜测，否则将减少猜测并继续进行，直到将最佳值定位到足够小的间隔为止。

还有待说明如何解决可行性问题(9)。注意到这是一个线性可行性问题，作者共同调用乘法权重（*MW*）算法和回归预言，以便找到一个近似可行的解决方案或将该问题证明为不可行。*MW* 是一种迭代算法，可在约束范围内保持权重 μ。在每次迭代中，公式(1)通过采用由 μ 加权的线性组合将约束分解为一个，公式(2)通过单个约束检查较简单问题的可行性，并且公式(3)如果较简单问题可行，则更新约束权重使用建议的解决方案的余量。算法 2 中介绍了步骤(1)和(3)的详细信息。

对于步骤(2)，必须解决的较简单问题采用以下形式：

$$?\exists P\in\Delta(G) such\ that\ \mu_0E_{g\sim P}[(g(x)-1)^2]+\sum_{j=1}^{i}\mu_jE_{g\sim P}\left[\hat{R}_j(g;y)\right]\le\ \mu_0c+\sum_{j=1}^{i}\mu_j\ \widetilde{\Delta}_j$$

该程序可以通过一次调用 *oracle* 回归来解决，因为左侧的所有项都涉及平方损失，而右侧的是常数。因此，可以使用回归预言高效地实现 *MW* 算法。最后，回想起上面的描述是针对目标 c 的固定值的，回想起最大值可以通过对 c 的二进制搜索来近似，这导致了基于 *oracle* 的算法来计算最大代价。对于此过程，具有以下计算保证。

定理 3 算法 2 返回估计值$\widehat{c_+}(x;y)$，使得$c_+(x;y)\le\widehat{c_+}(x;y)\le c_+(x;y)+TOL$并以 $O(max\{1, i^2/v_n^2\}\ log(i)\ log(1/TOL)/TOL^4)$调用 *oracle* 回归。

同时，可以用完全相同的方式估算最小代价，用程序(7)中的$(g(x)-0)^2$ 代替目标$(g(x)-1)^2$。在 *COAL* 中，作者在第 i 次迭代将 *TOL* 设置为 $1/\sqrt{i}$，并有 $v_n=O(d)$。结果，可以在处理 n 个示例后约束总的 *Oracle* 复杂度。

推论 4 处理完 n 个示例后，*COAL* 以 $O(K(d^3+n^5/d^2))$复杂度调用平方损失 *oracle*。

因此，对于允许有效平方损失优化的任何集合 G，可以在多项式时间内实现 *COAL*。与文献[23]要求 $O(n^2)$调用 *oracle* 相比，从表面上看，此算法的确较差，因为算法较慢。但是，算法将对版本空间施加更强的约束，这将导致更好的统计分析，这将在下面讨论。但是，这些以迭代或顺序方式使用批处理平方损失优化的算法对计算的要求太高，无法扩展到更大的问题。

5 论文实验结果和分析

与多类分类相比，*CSMC* 为成功和失败提供了一种更具表达力的语言，它使学习算法可以为获得良好性能而进行必要的取舍，并扩大了潜在的应用范围。例如，*CSMC* 可以自然地表达等级分类中的部分失败[29]。

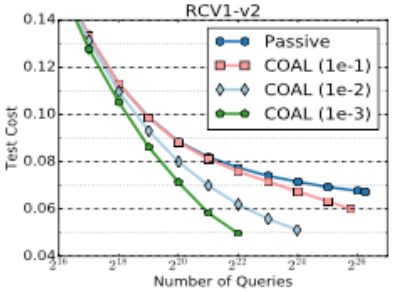

Figure 1: Empirical evaluation of COAL on Reuters text categorization dataset. Active learning achieves *better* test cost than passive, with a factor of 16 fewer queries.

通过实验，作者发现，*COAL* 在许多分层分类数据集的标注工作上节省了数量级（在被动式学习和 *COAL* 在 *Reuters* 文本分类上的比较，参见图 1），从而大大优于被动式学习基准。

5.1 模拟主动学习

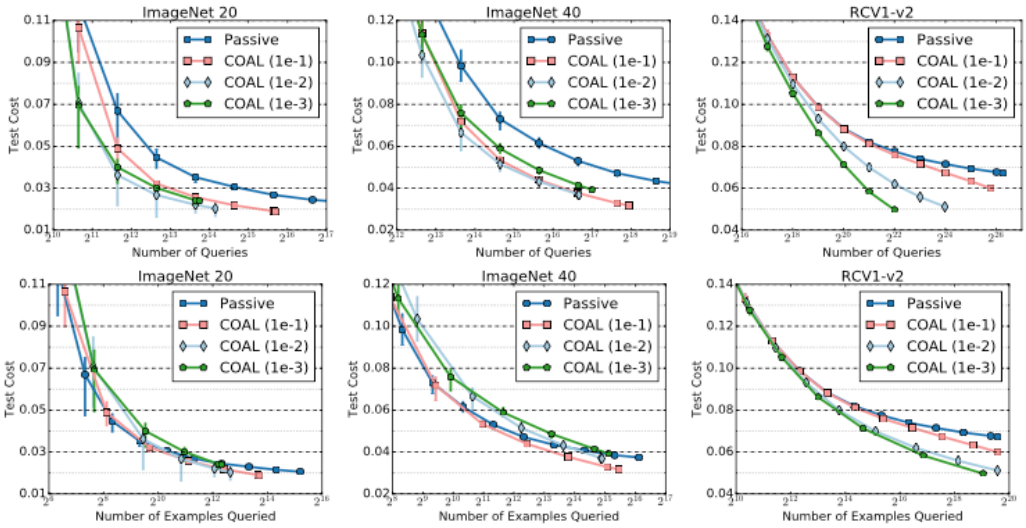

Figure 2: Experiments with COAL. Top row shows test cost vs. number of queries for simulated active learning experiments. Bottom row shows test cost vs. number of examples with even a single label query for simulated active learning experiments.

在图 2 的第一行中，作者针对每个数据集和成熟度绘制了针对中位数测试代价的查询数量以及从第 15 个分位数扩展到第 85 个分位数的条形图。总体而言，*COAL* 在性能和查询之间实现了更好的权衡。有了适当的柔和度参数，主动学习可以达到与被动学习类似的测试代价，查询量减少 8 到 32 倍。在 *ImageNet* 40 和 *RCV*1-*v*2（如图 1 所示）上，主动学习可实现更好的测试代价，查询量减少 16 倍。在 *RCV*1-*v*2 上，*COAL* 查询像被动查询一样多达 256*k* 查询，因为数据非常稀疏，并且线性回归具有以下特性：当

示例具有新的看不见的功能时，代价范围最大。一旦 *COAL* 几次看到所有功能，它就会查询更多。

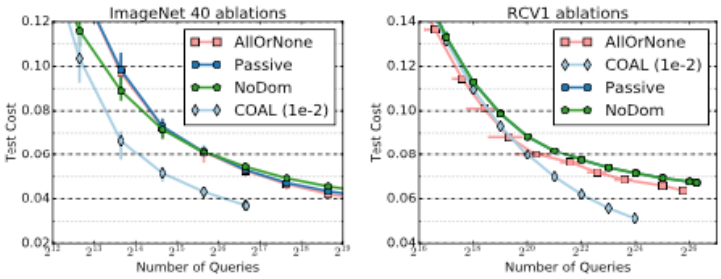

Figure 3: Test cost versus number of queries for COAL, in comparison with active and passive baselines on the ImageNet40 and RCV1-v2 dataset. On RCV1-v2, passive learning and NoDom are nearly identical.

作者还将 *COAL* 与两个活跃的学习基准进行比较。两种算法仅在查询规则方面与 *COAL* 不同。*AllOrNone* 可以同时使用支配条件和代价范围条件查询所有标签，也可以不查询任何标签，并且是对现有多班主动学习者的改编[18]。*NoDom* 只是根据主动回归的启发使用代价范围条件[30]。*ImageNet* 40 和 *RCV*1-*v*2 的结果显示在图 3 中，其中使用 *AUC* 策略选择学习率。通过目视检查选择醇厚度作为基线，而 *COAL* 使用 0.01。在 *ImageNet* 40 上，消融方式对被动学习的影响很小，而在 *RCV*1-*v*2 上，*AllOrNone* 确实提供了一定程度的改善。但是，在这两个数据集上，*COAL* 的性能都大大优于基线和被动学习。

虽然并不总是最好的，但作者建议将醇厚度设置为 0.01，因为它可以在所有三个数据集上实现合理的性能。

5.2 学习搜索

作者使用了 *Aggravate*[31,32,33]，该方法运行了一种习得的策略来产生标签的骨架序列。然后，对于输入中的每个位置，它将考虑所有可能的偏差动作，并对其余序列执行预言。此完整输出的损失将用作偏离操作的成本。以这种方式运行，当输入句子包含 *len* 个单词并且每个单词可以采用 *K* 个可能的标签之一时，*Aggravate* 需要 *len*×*K* 个卷展栏。

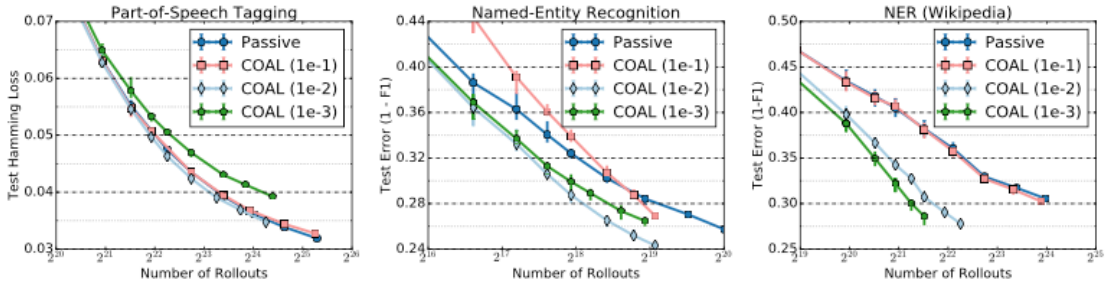

Figure 4: Learning to search experiments with COAL. Accuracy vs. number of rollouts for active and passive learning as the CSMC algorithm in learning-to-search.

由于每次推出都需要 *O*(*len*)时间，因此这在计算上是令人望而却步的，因此作者使用主动学习来减少推出次数。在三个联合预测数据集（统计数据在表 1 中）中使用了 *COAL* 和 *Aggravate* 内部的被动学习基线。如上所述，使用几个柔度值和相同的 *AUC* 标准来选择最佳学习率，结果在图 4 中，作者的推荐醇度再次为 0.01。

总体而言，主动学习减少了所需的推广次数，但在三个数据集上有所改善。在 *Wikipedia* 数据上，*COAL* 的推出次数减少了 4 倍，以实现与被动学习类似的性能，并获得更好的测试性能。*NER* 任务上出现了类似但不太生动的行为。另一方面，与被动学习有关的 *POS* 标记任务相比，*COAL* 的改进很小。这与作者的理论和先前的经验结果[34]相吻合，后者表明，主动学习不一定会在被动学习后得到改善。

6 总结

作者提出了一种新的主动学习算法，用于代价敏感的多类分类。该算法在运行时间，泛化误差和标签复杂度方面都具有很强的理论保证。主要的算法创新是一种计算版本空间中回归函数预测的最大和最小代价的新方法。作者还设计了一种在线算法，该算法受理论分析的启发，在 *CSMC* 和结构化预测中均优于被动基线。

从技术上讲，作者的算法使用平方损失预兆来搜索版本空间并驱动查询策略。这与最近使用 *argmax* 或 0/1 损失最小化 *oracle* 解决信息获取问题（如背景强盗）的最新结果形成对比[19]。由于这些通常涉及 *NP* 硬优化，因此一个有趣的问题是是否可以对其他信息获取问题使用平方损失预言。

7 参考文献

[1] *Turney P D. Types of cost in inductive concept learning. The Computing Research Repository*, 2002, *cs. LG*/0212034: 15-21.

[2] *Steve Hanneke. Theory of disagreement-based active learning. Foundations and Trends in Machine Learning*, 2014.

[3] *Maria Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In International Conference on Machine Learning*, 2006.

[4] *Chicheng Zhang and Kamalika Chaudhuri. Beyond disagreement-based agnostic active learning. In Advances in Neural Information Processing Systems*, 2014.

[5] *Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In Advances in Neural Information Processing Systems*, 2007.

[6] *Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In International Conference on Machine Learning*, 2009.

[7] *Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In Advances in Neural Information Processing Systems*, 2010.

[8] *Tzu-Kuo Huang*, *Alekh Agarwal*, *Daniel Hsu*, *John Langford*, *and Robert E. Schapire*. *Efficient and parsimonious agnostic active learning*. *In Advances in Neural Information Processing Systems*, 2015.

[9] *R.M. Castro and R.D. Nowak*. *Minimax bounds for active learning*. *Transaction on Information Theory*, 2008.

[10] *Stanislav Minsker*. *Plug-in approach to active learning*. *Journal of Machine Learning Research*, 2012.

[11] *Steve Hanneke and Liu Yang*. *Surrogate losses in passive and active learning*. *arXiv*:1207.3772, 2012.

[12] *Alexandra Carpentier*, *Andrea Locatelli*, *and Samory Kpotufe*. *Adaptivity to noise parameters in nonparametric active learning*. *In Conference on Learning Theory*, 2017.

[13] *Maria Florina Balcan*, *Andrei Broder*, *and Tong Zhang*. *Margin based active learning*. *In Conference on Learning Theory*, 2007.

[14] *Maria Florina Balcan and Philip M. Long*. *Active and passive learning of linear separators under log-concave distributions*. *In Conference on Learning Theory*, 2013.

[15] *Giovanni Cavallanti*, *Nicolo Cesa-Bianchi*, *and Claudio Gentile*. *Learning noisy linear classifiers via adaptive and selective sampling*. *Machine Learning*, 2011.

[16] *Ofer Dekel*, *Claudio Gentile*, *and Karthik Sridharan*. *Robust selective sampling from single and multiple teachers*. *In Conference on Learning Theory*, 2010.

[17] *Francesco Orabona and Nicolo Cesa-Bianchi*. *Better algorithms for selective sampling*. *In International Conference on Machine Learning*, 2011.

[18] *Alekh Agarwal*. *Selective sampling algorithms for cost-sensitive multiclass prediction*. *In International Conference on Machine Learning*, 2013.

[19] *Alekh Agarwal*, *Daniel Hsu*, *Satyen Kale*, *John Langford*, *Lihong Li*, *and Robert E. Schapire*. *Taming the monster*: *A fast and simple algorithm for contextual bandits*. *In International Conference on Machine Learning*, 2014.

[20] *Vasilis Syrgkanis*, *Akshay Krishnamurthy*, *and Robert E. Schapire*. *Efficient algorithms for adversarial contextual learning*. *In International Conference on Machine Learning*, 2016.

[21] *Hal Daume III*, *John Langford*, *and Daniel Marcu*. *Search-based structured prediction*. *Machine Learning*, 2009.

[22] *John Langford and Alina Beygelzimer*. *Sensitive error correcting output codes*. *In Conference on Learning Theory*, 2005.

[23] *Akshay Krishnamurthy*, *Alekh Agarwal*, *Tzu-Kuo Huang*, *Hal Daum´e*, *III*, *and John Langford*. *Active learning for cost-sensitive classification*. *In International Conference on Machine Learning*, 2017.

[24] *Enno Mammen and Alexandre B. Tsybakov. Smooth discrimination analysis. The Annals of Statistics*, 1999.

[25] *Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. Annals of Statistics*, 2004.

[26] *Dylan Foster, Alekh Agarwal, Miroslav Dudik, Haipeng Luo, and Robert Schapire. Practical contextual bandits with regression oracles. In International Conference on Machine Learning*, 2018.

[27] *Tengyuan Liang, Alexander Rakhlin, and Karthik Sridharan. Learning with square loss: Localization through offset rademacher complexity. In Conference on Learning Theory*, 2015.

[28] *Alexander Rakhlin and Karthik Sridharan. On equivalence of martingale tail bounds and deterministic regret inequalities. In Proceedings of the* 2017 *Conference on Learning Theory*, 2017.

[29] *Carlos N. Silla Jr. and Alex A. Freitas. A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery*, 2011.

[30] *Rui Castro, Rebecca Willett, and Robert D. Nowak. Faster rates in regression via active learning. In Advances in Neural Information Processing Systems*, 2005.

[31] *Stephane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. arXiv*:1406.5979, 2014.

[32] *Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daum´e III, and John Langford. Learning to search better than your teacher. In International Conference on Machine Learning*, 2015.

[33] *Wen Sun, Arun Venkatraman, Geoffrey J. Gordon, Byron Boots, and J. Andrew Bagnell. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In International Conference on Machine Learning*, 2017.

[34] *Daniel Hsu. Algorithms for Active Learning. PhD thesis, University of California at San Diego*, 2010.