Multi-label Active Learning with Auxiliary Learner

Chen-Wei Hung R98922121@CSIE.NTU.EDU.TW
Hsuan-Tien Lin HTLIN@CSIE.NTU.EDU.TW
Department of Computer Science and Information Engineering, National Taiwan University

Editor: Chun-Nan Hsu and Wee Sun Lee

Abstract

Multi-label active learning is an important problem because of the expensive labeling cost in multi-label classification applications. A state-of-the-art approach for multi-label active learning, maximum loss reduction with maximum confidence (MMC), heavily depends on the binary relevance support vector machine in both learning and querying. Nevertheless, it is not clear whether the heavy dependence is necessary or unrivaled. In this work, we extend MMC to a more general framework that removes the heavy dependence and clarifies the roles of each component in MMC. In particular, the framework is characterized by a major learner for making predictions, an auxiliary learner for helping with query decisions and a query criterion based on the disagreement between the two learners. The framework takes MMC and several baseline multi-label active learning algorithms as special cases. With the flexibility of the general framework, we design two criteria other than the one used by MMC. We also explore the possibility of using learners other than the binary relevance support vector machine for multi-label active learning. Experimental results demonstrate that a new criterion, soft Hamming loss reduction, is usually better than the original MMC criterion across different pairs of major/auxiliary learners, and validate the usefulness of the proposed framework.

Keywords: Active Learning, Multi-label Classification, Support Vector Machine, Query Criteria

1. Introduction

In many practical applications of machine learning, it is expensive to obtain labeled instances for training. For example, in medical applications, labeling usually requires hiring a professional doctor, which is costly in terms of both time and money. Such applications call for active learning (Settles, 2010), which actively queries the labels of only a few instances while maintaining good prediction performance.

Many existing works on active learning focus on tackling the binary classification problem. An earlier work (Seung et al., 1992) proposes the *query by committee* approach to query the label of a single instance on which a committee of learners disagree the most in prediction. In other words, the approach queries the most ambiguous instance from the view of a committee of learners. Many modern approaches consider only one single learner and locate the most ambiguous instance to be queried with different criteria. For example, some approaches compute the distance from an instance to the decision boundary produced by the single learner as the level of ambiguity and queries the most ambiguous (closest-to-boundary) instance. One popular representative of those approaches is *active learning with the support vector machine* (Tong et al., 2000), which takes the support vector machine (Vapnik, 1995) as the single learner.

The binary classification problem assumes that each instance can only be associated with one of the two possible class labels. The problem can be extended to a more general one that allows each instance to be associated with a subset of K possible labels, where $K \geq 2$ is the number of classes. The general problem is called multi-label classification (Tsoumakas and Katakis, 2007) and can be used in many applications in document and multimedia information retrieval (McCallum, 1999; Wang et al., 2008). Arguably the simplest algorithm for solving the multi-label classification problem is *binary relevance* (Godbole and Sarawagi, 2004), which decomposes a K-class multi-label classification problem to K binary classification ones. In the k-th binary classification problem, a binary classifier is trained to tackle the yes/no question on whether the instance is associated with class k. Binary relevance is an important baseline approach for multi-label classification. Nevertheless, the approach cannot directly model the joint information between different labels because of the decomposition. Many extensions (Yang et al., 2009; Read et al., 2009) of binary relevance are thus taken to include the joint information during training.

Compared with binary classification, multi-label classification applications usually pay higher costs in labeling the instances and multi-label classification algorithms usually need more labeled instances to reach satisfactory performance. The needs justify the importance of active learning for multi-label classification (a.k.a. multi-label active learning), which is a challenging research direction that attracts much research attention in recent years (Li et al., 2004; Brinker, 2006; Yang et al., 2009). For multi-label active learning, a baseline approach called *binary version space minimization* (BinMin; Brinker, 2006) works by combining the idea of query-the-closest-instance with binary relevance. In particular, BinMin queries an instance that is closest to the decision boundary for *one of* the binary classifiers within binary relevance. Some other query strategies try to take the predictions of *all* the binary classifiers into account. For example, *maximum loss reduction with maximum confidence* (MMC; Yang et al., 2009) considers a loss function that is jointly defined from the predictions of all the classifiers and queries instances that reduce the loss function the most. Briefly speaking, MMC takes a special internal method to output the most probable labels; then, it decides the instances to be queried from the output of the internal method and some distance-based ambiguity information defined from binary relevance with the support vector machine.

In this work, we extend MMC to a more general framework that clarifies the roles of the internal method and the binary relevance algorithm. In particular, the internal method is called a *major learner*, highlighting its role for outputting the labels; the binary relevance algorithm is called an *auxiliary learner*, emphasizing its role for providing information to help query decisions. The two learners in the framework evolve together during the active learning process, which decides the queries by some criterion based on the disagreement between the two learners. In other words, the framework can be thought as a sibling of query-by-committee using a committee of size two for the multi-label classification problem. The general framework includes MMC, BinMin and the (non-active) random query algorithm as special cases. In addition, it allows us to design better multi-label active learning algorithms with the flexibility of three choices: the major learner, the auxiliary learner and the query criterion.

With the general framework, we explore query criteria other than the one used by MMC. In particular, we propose another criterion based on reducing the Hamming loss, one of the most natural loss functions for multi-label classification (Zhang and Zhou, 2006, 2007; Tai and Lin, 2010). The resulting *Hamming loss reduction* (HLR) criterion simplifies to querying instances that the two learners disagree the most in terms of the Hamming loss. Unlike the MMC criterion, which can be easily affected by a few extreme values in the numerical output of the auxiliary learner, HLR is less affected by the values and hence can result in better queries. Experiments on real-world data sets verify that HLR is a competitive alternative over MMC for multi-label active learning. Furthermore, we improve HLR by proposing the *soft Hamming loss reduction* (SHLR) criterion, which combines the extreme-value tolerance of HLR and the distance-based ambiguity of MMC. Experimental results demonstrate that SHLR usually reaches better performance over both MMC and HLR when fairly coupled with the same pair of major/auxiliary learners.

The paper is organized as follows. First, we introduce the multi-label active learning problem in Section 2. We extend from MMC to our proposed framework in Section 3, derive HLR and SHLR criteria in Section 4, and compare the proposed criteria with state-of-the-art approaches on real-world data sets in Section 5. Finally, we conclude in Section 6.

2. Multi-label active learning

In (non-active) multi-label classification (Tsoumakas and Katakis, 2007), we seek for a classifier that maps each instance $x \in \Re^d$ to a label-set $y \subseteq \{1, 2, ..., K\}$, where K is the number of classes. The label-set y is often conveniently represented as a K-dimensional binary vector in $\{-1, +1\}^K$, where the k-th component is $+1$ if and only if label k is an element of y (Tai and Lin, 2010). Throughout this work, we take the binary vector representation. Given a labeled training set D_l that contains N training examples of the form (x_n, y_n) for $n = 1, 2, \cdots, N$, a multi-label learner $\mathcal{F}$ learns a decision function $f\colon \Re^d \to \Re^K$ from D_l with the hope that its signed output $\hat{f}(x) = \text{sign}(f(x)) \in \{-1, +1\}^K$ predicts y well on any future test example (x, y).

In this work, we consider the setup of pool-based active learning (Lewis and Gale, 1994) for multi-label classification. The setup has also been taken by many existing works (McCallum and Nigam, 1998; Zhang and Chen, 2002; Li et al., 2004; Yang et al., 2009). In the setup, there are R rounds of queries. For the first round, in addition to the training set (labeled pool) D_l, there is an unlabeled pool $D_u = \{(x'_m)\}_{m=1}^{M}$. For the r-th round with $r = 1, 2, \cdots, R$, the active learning algorithm calls a learner $\mathcal{F}$ on D_l to learn a decision function f. Then, based on f, D_l, D_u and S (the allowed number of queries), the active learning algorithm selects a size-S subset $D_s \subseteq D_u$ to be queried. The instances in D_s, along with the labels obtained from a labeling oracle, are then added to D_l and removed from D_u. That is, N is increased by S and M is decreased by S. The setup aims at getting f that predicts unseen instances $(x, y) \in D_u$ well while using a small number of rounds. The steps of whole setup are listed in Algorithm 1.

Algorithm 1 Pool-based multi-label active learning

```
Input: a labeled pool $D_l$; an unlabeled pool $D_u$; the number of rounds $R$; the allowed number of
    queries $S$; a labeling Oracle; a multi-label learner $\mathcal{F}$
for $r = 1, 2, \cdots, R$ do
    $f \leftarrow \mathcal{F}(D_l)$
    $D_s \leftarrow$ query($f$, $D_l$, $D_u$, $S$)
    $D_l \leftarrow D_l \cup (D_s$, Oracle.label$(D_s))$; $D_u \leftarrow D_u \setminus D_s$
end for
```

We highlight several details of the setup that are adopted in this work. The details are same as the ones used by representative existing works (Li et al., 2004; Yang et al., 2009). Firstly, we assume that $\mathcal{F}$ is a supervised routine that only trains with D_l, which allows using mature supervised multi-label learners for active learning; secondly, we assume that S is a fixed value that can be more than 1, which is a more flexible setting that fits the real-world needs better; lastly, we assume to be getting the full label-set y'_m from the labeling oracle for each instance $x'_m \in D_s$, which means we ask the precious oracle to provide all the information for each instance in D_s at once in each round.

3. Multi-label active learning with auxiliary learner

In this section, we start by introducing a state-of-the-art approach, *maximum loss reduction with maximum confidence* (MMC) (Yang et al., 2009). The approach fits the setup established in Section 2. Then, we extend MMC to a more general framework that clarifies the roles of the internal blocks.

3.1. Maximum loss reduction with maximum confidence

MMC is built from the binary relevance support vector machine (SVM) for multi-label classification. In particular, denote the binary relevance SVM as a multi-label learner $\mathcal{G}$, which learns a decision function $g(x)$ from D_l. We will denote the k-th component of $g(x)$ by $g^{(k)}(x)$, which outputs the decision value of the k-th SVM on an instance x. In each round of MMC, after learning $g(x)$ from D_l, the functions $g^{(k)}$ are combined by a *stacking with logistic regression* (SLR) learner $\mathcal{F}_g$ to obtain a decision function f_g, which plays the role of f in Algorithm 1. After obtaining $f_g(x)$, the optimal set D_s^* to be queried is determined with the *maximum margin reduction* (MMR) criterion, which will be discussed further in Section 4.2:

$$D_s^* \quad = \underset{|D_s|=S, D_s \subseteq D_u}{\text{argmax}} \left(\textstyle\sum_{x' \in D_s} \sum_{k=1}^{K} \left(\dfrac{1 - \hat{f}_g^{(k)}(x') \cdot g^{(k)}(x')}{2} \right) \right). \tag{1}$$

Let the instance-wise scoring function of MMR be

$$U_{\text{MMR}}(x' \mid f_g, g) = \sum_{k=1}^{K} \left(\frac{1 - \hat{f}_g^{(k)}(x') \cdot g^{(k)}(x')}{2} \right).$$

Equation (1) is the same as ordering each instance $x'_m \in D_u$ by the scoring function and then include the top-S instances as D_s^*. Several questions immediately arise after listing (1). First, does $\mathcal{G}$ have to be the binary relevance SVM or even the binary relevance algorithm? Second, does $\mathcal{F}_g$ have to depend on g, or can we use other multi-label learners? Third, can we use a better scoring function other than U_{MMR}? The questions motivate us to extend MMC to a more general framework, which is parameterized by more flexible $\mathcal{G}$, $\mathcal{F}$ and U, as discussed next.

3.2. The proposed framework

Our proposed framework is called *active learning with auxiliary learner*, as listed in Algorithm 2. Compared with the basic setup of pool-based active learning in Algorithm 1, we add an auxiliary learner called $\mathcal{G}$ and name the original $\mathcal{F}$ as a major learner. We take a special instance-wise scoring function U based on decision functions f and g, and consider a query criterion that selects S instances with the highest scores from D_u.

MMC can be viewed as a special case of the proposed framework. In original formulation, MMC uses the binary relevance SVM as the auxiliary learner $\mathcal{G}$, stacking with logistic regression (that depends on g) as the major learner $\mathcal{F}$, and U_{MMR} as the instance-wise scoring function. We can

Algorithm 2 Active learning with auxiliary learner

```
Input: a labeled pool $D_l$; an unlabeled pool $D_u$; the number of rounds $R$; the allowed number of
    queries $S$; a labeling Oracle; a major learner $\mathcal{F}$; a auxiliary learner $\mathcal{G}$; an instance-wise scoring
    function $U$.
for $r = 1, 2, \cdots, R$ do
    $f \leftarrow \mathcal{F}(D_l)$; $g \leftarrow \mathcal{G}(D_l)$; $u_m \leftarrow U(x'_m \mid f, g)$, for all $x'_m \in D_u$
    $D_s \leftarrow \{x'_m$ with top $S$ scores of $u_m\}$
    $D_l \leftarrow D_l \cup (D_s$, Oracle.label$(D_s))$; $D_u \leftarrow D_u \setminus D_s$
end for
```

also extend the original MMC to a more general one that allows any $\mathcal{F}$ and $\mathcal{G}$ to be used, with an underlying requirement that $\mathcal{G}$ learns per-label decision functions $g^{(k)}$ with real-valued outputs like the binary relevance SVM. Such a $\mathcal{G}$ will be called a margin-based multi-label learner.

An earlier active learning algorithm *binary version space minimization* (BinMin; Brinker, 2006) can also be viewed as a special case of the proposed framework. The original BinMin takes only one learner: the binary relevance SVM. We can view the learner as both the major one $\mathcal{F}$ and the auxiliary one $\mathcal{G}$. Then, the decision functions f and g are the same. BinMin applies the following instance-wise scoring function that only uses g to form the query criterion:

$$U_{\text{BM}}(x' \mid f, g) = \frac{1}{\min_{k=1,\cdots,K} \left| g^{(k)}(x') \right|}.$$

The absolute value of $g^{(k)}(x')$ can be viewed as the confidence margin of the k-th SVM on x'. Yang et al. (2009) show that the original BinMin is practically inferior to MMC because BinMin only considers the worst-case confidence margin from one label rather than taking all labels together when making query decisions. Nevertheless, there are two differences between the original BinMin and the original MMC: the former uses the binary relevance SVM for the same $\mathcal{F}$ and $\mathcal{G}$ while the latter uses a more powerful $\mathcal{F}$; the former uses U_{BM} and the latter uses U_{MMR}. Thus, it is not clear whether the inferior performance should be attributed to the difference of learners, the difference of scoring functions, or both.

Similar to the extended MMC formulation, we can consider an extended BinMin that takes any multi-label learner $\mathcal{F}$ (that can be different from $\mathcal{G}$) and any margin-based multi-label learner $\mathcal{G}$ (that can be different from binary relevance SVM). The extended BinMin can then be fairly compared with the extended MMC using the same $\mathcal{F}$ and $\mathcal{G}$. We will empirically make the comparisons in Section 5.

The simple (non-active) random query criterion is yet another special case of the framework. The criterion is $U_{\text{rand}}(x' \mid f, g) = \text{random}()$, which does not depend on either f or g.

Next, we deepen the study of active learning with auxiliary learner by exploring whether some query criteria other than MMR (the one used by MMC) can improve the active learning performance. We start by deriving some novel query criteria, as discussed in the next section.

4. Query criteria

We first study the derivation steps and properties of *maximum margin reduction* (MMR), the query criterion within MMC. Then, we propose two other query criteria, *Hamming loss reduction* and *soft Hamming loss reduction*.

4.1. Approximate maximum loss reduction

MMR roots from the paradigm of maximum loss reduction. The paradigm is widely used for deriving query criteria in active learning (Tong, 2001; Roy and McCallum, 2001; Yang et al., 2009). In the paradigm, a loss function is used to evaluate the performance of a decision function g. Since the goal of active learning is to improve the prediction performance as fast as possible, we want to reduce the loss function as fast as possible. Thus, maximum loss reduction queries the set $D_s \subseteq D_u$ that can result in the maximum expected loss reduction with respect to g.

Formally speaking, let g be the decision function returned by a multi-label learner $\mathcal{G}$ when trained with D_l; let g_{+s} be the decision function returned by $\mathcal{G}$ when trained with $D_l \cup D_s$. Consider a loss function $L(g, x, y)$ between the decision function g and an instance (x, y). With respect to the conditional probability distribution $P(y \mid x')$ that generates the label-set of x', the expected loss of g over D_u is

$$E = \frac{1}{M} \sum_{x' \in D_u} \sum_{y \in \mathcal{Y}} L\left(g, x', y\right) P(y \mid x'),$$

where $\mathcal{Y} = \{-1, +1\}^K$ contains all possible binary-vector representations of label-sets. The expected loss of g_{+s} over D_u with respect to $P(y \mid x')$ is

$$E_{+s} = \frac{1}{M} \sum_{x' \in D_u} \sum_{y \in \mathcal{Y}} L\left(g_{+s}, x', y\right) P(y \mid x').$$

Maximum loss reduction aims at finding the size-S subset D_s^* that result in the maximum expected loss reduction. That is,

$$\begin{aligned} D_s^* &= \operatorname*{argmax}_{|D_s|=S, D_s \subseteq D_u} \quad \{E - E_{+s}\} \\ &= \operatorname*{argmax}_{|D_s|=S, D_s \subseteq D_u} \quad \Bigg\{ \sum_{x' \in D_s} \sum_{y \in \mathcal{Y}} \big(L(g, x', y) - L(g_{+s}, x', y)\big) P(y \mid x') \\ &\qquad\qquad + \sum_{x' \in D_{u-s}} \sum_{y \in \mathcal{Y}} \big(L(g, x', y) - L(g_{+s}, x', y)\big) P(y \mid x') \Bigg\}. \qquad (2) \end{aligned}$$

where $D_{u-s} = D_u \setminus D_s$.

Directly solving (2) is computationally prohibitive because it requires enumerating over all possible D_s and knowing the exact $P(y \mid x')$ over all 2^K possible y vectors. Existing works (Li et al., 2004; Yang et al., 2009) thus rely on several assumptions to simplify (2), as discussed below.

First, assume that $g(x') \approx g_{+s}(x')$ for all $x' \notin D_l \cup D_s$. In other words, the decision function g does not change much outside $D_l \cup D_s$ when trained with the additional set D_s. Then, for all instance in D_{u-s}, $L(g, x', y) \approx L(g_{+s}, x', y)$ and thus the second term in (2) is approximately 0.

Furthermore, assume that the probability function $P(y \mid x')$ in (2) can be well-approximated by an impulse function at $y = \hat{f}(x')$, where $\hat{f}(x')$ comes from applying a strong multi-label learner (the major learner in our proposed framework) $\mathcal{F}$ on D_l and is supposed to return the most probable label-set of x'. The assumption avoids the computational burden of enumerating over all possible y in (2). Applying both assumptions result in

$$D_s^* \quad = \underset{|D_s|=S, D_s \subseteq D_u}{\operatorname{argmax}} \sum_{x' \in D_s} \Big(L\big(g, x', \hat{f}(x')\big) - L\big(g_{+s}, x', \hat{f}(x')\big) \Big). \tag{3}$$

Equation (3) is a common formulation called *approximate maximum loss reduction*. Nevertheless, it is still computationally expensive because getting g_{+s} for every different choice of D_s needs calling $\mathcal{G}$ on all possible $D_l \cup D_s$ combinations. The MMR criterion takes a specific loss function and relies on another approximation to further simplify (3) to a simpler criterion that uses only an instance-wise scoring function U_{MMR}. Next, we discuss the loss function and the approximation taken by MMR. Then, we propose a novel query criterion based on a different loss function and a different assumption.

4.2. MMR Scoring function

The MMR criterion is derived from a loss function that is based on the total size of the version spaces for all $g^{(k)}$ in the binary relevance SVM. The version space contains all the possible decision functions with respect to the given examples in D_l. A smaller version space indicates less ambiguity in locating a suitable decision function and is hence intuitively better. BinMin (Brinker, 2006) considers the size reduction of the version space in the worst case. The MMR criterion, on the other hand, takes the total size into account. It is argued (Yang et al., 2009) that when taking a binary relevance SVM or any margin-based learner $\mathcal{G}$ and considering a loss function based on the version space,

$$L\big(g, x', \hat{f}(x')\big) - L\big(g_{+s}, x', \hat{f}(x')\big) \approx \sum_{k=1}^{K} \frac{1 - \hat{f}^{(k)}(x') \cdot g^{(k)}(x')}{2}.$$

In other words, MMR can use only the current g (and the current $\hat{f}$) to estimate the loss difference, which avoids the computational burden of getting different g_{+s} in approximate maximum loss reduction (3). The approximation step results in the following instance-wise scoring function, as mentioned in Section 3.

$$U_{\text{MMR}}(x' \mid f, g) = \sum_{k=1}^{K} \left(\frac{1 - \hat{f}^{(k)}(x') \cdot g^{(k)}(x')}{2} \right).$$

The term $\hat{f}^{(k)}(x') \cdot g^{(k)}(x')$ in U_{MMR} can be viewed as the joint ambiguity between $\hat{f}$ and g. The joint ambiguity is large when $\hat{f}^{(k)}(x')$ and $g^{(k)}(x')$ are of different signs (and $g^{(k)}$ is large); the joint ambiguity is small when $\hat{f}^{(k)}(x')$ and $g^{(k)}(x')$ are of the same sign. There is, however, one shortcoming of U_{MMR}. In particular, for an instance $x' \in D_u$, if $\hat{f}(x')$ and $g(x')$ disagree a lot in the k-th label (in terms of the raw value of $g^{(k)}(x')$), the large ambiguity causes the instance to receive a high score. Then, similar to BinMin, the single worst label of an instance dominates the choice of query decisions. On the other hand, if $\hat{f}(x')$ and $g(x')$ agree a lot in the k-th label, the lack of ambiguity on this dimension causes the instance to receive a low score and may not be chosen for query. That is, the single best label of an instance dominates the making of (non-)query decisions. When the numerical ranges of $g^{(k)}(x')$ are different across different labels k, the MMR criterion focuses more on the dimensions k in which $g^{(k)}(x')$ are of wider numerical ranges. Such a focus effectively causes queries to be decided with a few rather than all of the dimensions and affects the performance of MMR. The shortcoming, called sensitivity to a few extreme values, has been observed during our studies on U_{MMR} in MMC. Thus, we propose other scoring functions that avoids the sensitivity shortcoming.

4.3. Hamming loss reduction

We couple (3) with a different loss function that does not depend on a margin-based learner (for estimating the size of the version space) and hence avoids the sensitivity shortcoming of MMR. The loss function is the Hamming loss, which is a commonly-used loss function for multi-label classification (Zhang and Zhou, 2006, 2007; Tai and Lin, 2010). Consider a multi-label example (x, y) and a decision function g, the (scaled) Hamming loss of g on the example is defined as

$$L(g, x, y) = \sum_{k=1}^{K} [\![\hat{g}^{(k)}(x) \neq y^{(k)}]\!].$$

The loss function only depends on $\hat{g}$, which does not need to be generated from a margin-based learner.

When the Hamming loss function is plugged into (3), we derive a new criterion for multi-label active learning.

$$D_s^* \quad = \quad \operatorname*{argmax}_{|D_s|=S, D_s \subseteq D_u} \sum_{x' \in D_s} \sum_{k=1}^{K} \Big([\![\hat{g}^{(k)}(x') \neq \hat{f}^{(k)}(x')]\!] - [\![\hat{g}_{+s}^{(k)}(x') \neq \hat{f}^{(k)}(x')]\!] \Big). \tag{4}$$

Nevertheless, equation (4) still depends on $\hat{g}_{+s}$, which results in the computational burden. To simplify (4) to an instance-wise scoring function, we consider an approximation of the term

$$\sum_{x' \in D_s} \sum_{k=1}^{K} [\![\hat{g}_{+s}^{(k)}(x') \neq \hat{f}^{(k)}(x')]\!].$$

In particular, for any $x' \in D_s$, let y be the (most-probable) label-set vector that the labeling oracle returns. Then,

$$0 \leq \sum_{x' \in D_s} \sum_{k=1}^{K} [\![\hat{g}_{+s}^{(k)}(x') \neq \hat{f}^{(k)}(x')]\!] \leq \sum_{x' \in D_s} \sum_{k=1}^{K} [\![\hat{g}_{+s}^{(k)}(x') \neq y^{(k)}]\!] + \sum_{x' \in D_s} \sum_{k=1}^{K} [\![y^{(k)} \neq \hat{f}^{(k)}(x')]\!]$$

The second term on the right-hand side is close to 0 because $\hat{f}^{(k)}$ has been assumed to be a good approximator of $P(y \mid x')$ in approximate maximum loss reduction. We further assume that the first term, which is the partial training error of $\hat{g}_{+s}$, to be close to 0. The assumption happens when $\mathcal{G}$ is powerful enough to return a decent $\hat{g}_{+s}$. Using the assumptions, we derive the *Hamming loss reduction* (HLR) criterion to be

$$D_s^* \quad = \quad \operatorname*{argmax}_{|D_s|=S, D_s \subseteq D_u} \sum_{x' \in D_s} \sum_{k=1}^{K} \Big([\![\hat{g}^{(k)}(x') \neq \hat{f}^{(k)}(x')]\!] \Big).$$

The criterion can be performed by selecting top-scored S instances from D_u using the instance-wise scoring function

$$U_{\text{HLR}}(x' \mid f, g) = \sum_{k=1}^{K} \Big([\![\hat{g}^{(k)}(x') \neq \hat{f}^{(k)}(x')]\!] \Big).$$

The scoring function U_{HLR} can be interpreted as querying the instances that two multi-label classifiers $\hat{f}$ and $\hat{g}$ disagree the most in terms of the Hamming distance. The interpretation makes HLR a sibling of to the traditional query by committee algorithm for active learning in binary classification (Seung et al., 1992), which queries the instances that a committee of binary classifiers disagree the most.

4.4. Soft Hamming loss reduction

When using a margin-based auxiliary learner like the binary relevance SVM, the scoring functions U_{MMR} and U_{HLR} take the two extremes of using the margin information. The function U_{MMR} takes all the margin information into account, and thus can be affected by a few extreme margin values. The function U_{HLR}, on the other hand, does not use any margin information. To explore whether the margin information is important in multi-label active learning, we design another instance-wise scoring function that can be viewed as a mixture of U_{MMR} and U_{HLR}. In particular, the scoring function takes the margin information from the decision function $g^{(k)}(x')$ and the estimated label $\hat{f}^{(k)}(x')$ like U_{MMR}. But instead of using the value $g^{(k)}(x') \cdot \hat{f}^{(k)}(x')$ directly in the instance-wise scoring function (as U_{MMR} does), the value is clipped to the range $[-b, b]$ to remove the influence of extreme margin values. Define

$$U_{\text{SHLR}}(x' \mid f, g) = \sum_{k=1}^{K} \frac{b - clip\left(\hat{f}^{(k)}(x') \cdot g^{(k)}(x'), b\right)}{2b},$$

where $clip(A, b) = \max(\min(A, b), -b)$. When $b \to \infty$, we see that U_{SHLR} is similar to U_{MMR} because all the margin information is preserved when computing the instance-wise scoring function. On the other hand, when $b \to 0$,

$$\begin{aligned} \sum_{k=1}^{K} \frac{b - clip(\hat{f}^{(k)}(x') \cdot g^{(k)}(x'), b)}{2b} &\approx \sum_{k=1}^{K} \frac{b - b \cdot (2[\![\hat{f}^{(k)}(x') = \hat{g}^{(k)}(x')]\!] - 1)}{2b} \\ &= \sum_{k=1}^{K} [\![\hat{f}^{(k)}(x') \neq \hat{g}^{(k)}(x')]\!]. \end{aligned}$$

That is, U_{SHLR} can be viewed as a relaxed version of U_{HLR} that takes some margin information into account. We propose a query criterion that uses U_{SHLR} as the scoring function to select the top-scored S instances from D_u, and name the criterion *soft Hamming loss reduction* (SHLR).

When comparing MMR, HLR and SHLR (with $b \geq 1$), we see that they are exactly the same if the auxiliary learner $\mathcal{G}$ is not margin-based. In that case, $\hat{g}(x) = g(x)$ and all three criteria result in exactly the same query decisions. The three criteria thus all fit the paradigm of query by committee of size two. When the auxiliary learner is margin-based, like the binary relevance SVM, the MMR criterion takes all the margin information; the HLR criterion ignores all the margin information; the SHLR criterion takes partial margin information. Next, we make a fair empirical comparison and see if the different amount of margin information leads to different performance in active learning.

Table 1: Details of each data sets

Data set	# Instances	# Features	# Labels	Data set	# Instances	# Features	# Labels
Rcv1	3000	47236	103	Y!Ed	6030	27534	33
Y!Ar	3712	23146	26	Y!En	6356	32001	21
Y!Bu	5710	21924	30	yeast	2000	103	14
Y!Co	6270	34096	33	scene	2000	294	6

5. Experiment

We evaluate the proposed query criteria and compare them with MMR on real-world benchmark data sets. In Section 5.2, we use the same major/auxiliary learner as the original MMC approach. Then, in Sections 5.3 and 5.4, we use other major/auxiliary learner combinations to validate the usefulness of the general framework.

5.1. Setting

Data sets. We consider eight real-world data sets in the experiments. The first data set is the "topics" task of RCV1-V2 (Lewis et al., 2004), which is a text classification task with instances being documents from Reuters newswire stories. The data set is downloaded from LIBSVM-Tools[1] and will be denoted as rcv1. We select 3000 instances from subsets to form our data pool. There are 47236 features and 103 classes in rcv1 data set.[2]

We use five more data sets on classifying web pages from the top directory of Yahoo!.[3] Each data set is associated with one top directory, and each instance within the data set is a web page that is labeled as one or more sub-directories. We take the Arts (Y!Ar), Business (Y!Bu), Computers (Y!Co), Education (Y!Ed) and Entertainment (Y!En) sets. The rcv1 and the five Yahoo! data sets have also been used for evaluating the MMC approach (Yang et al., 2009) and are thus included in our experiments. For these six data sets, the instances (documents) are transformed to the TF-IDF format and normalized to a unit-length vector.

In addition to the text-classification data sets that are used for evaluating the MMC approach (Yang et al., 2009), we consider two general multi-label classification data sets, yeast and scene. The two sets are popularly used for evaluating multi-label learners. (Zhang and Zhou, 2006; Read et al., 2008; Grodzicki et al., 2008; Tai and Lin, 2010). The yeast data set is a biological one on protein classification; the scene data set is a visual one on still-scene classification. The two data sets are both also downloaded from LIBSVM-Tools.

The detail information of each data set is in Table 1. In Yahoo! data sets, we directly take their training data as our data pool. In scene and yeast data sets, we select 2000 instances from the combination of training and testing data to form our data pool.

Active learning environment. For each data set, we randomly choose 500 instances with labels as the initial labeled pool D_l and leave the remaining instances as the unlabeled pool D_u. In each round, each active learning algorithm is asked to query $S = 20$ instances. We consider a total of $R = 50$ rounds. That is, the final labeled pool would contain 1500 instances. Each experiment is repeated for 6 trials with the same initial D_l for each algorithm and the average result of the 6 trials

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html
[2] The actual number of classes that comes with at least one example is 101.
[3] http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.gz

are reported. The active learning settings is same as MMC paper, except that the paper repeats the experiments for 10 trials.

Evaluation measure. We consider two different evaluation measures. One is the micro-averaged F1-score, which is between 0 and 1, with higher value representing better performance. The other is the Hamming loss (as discussed in Section 4.3 for deriving HLR), which is also between 0 and 1 when normalized with respect to K. For the Hamming loss, lower value represents better performance. The original MMC paper only uses the F1-score to evaluate the performance. We also include the Hamming loss because of its popularity in evaluating multi-label learners (Zhang and Zhou, 2006, 2007; Tai and Lin, 2010).

Major/Auxiliary learners. We consider three different combinations of major/auxiliary learners. The first one takes stacking with logistic regression (SLR) as the major learner and binary relevance SVM (BR) as the auxiliary one. We take the combination to fairly compare MMC with the proposed HLR and SHLR criteria.

The second combination takes a different major learner, Classifier Chain (CC; Read et al., 2009) with SVM, while keeping BR as the auxiliary learner. CC is a leading multi-label classification algorithm. Similar to BR, each class in CC is also modeled with a single binary SVM and thus CC can provide margin information. Nevertheless, CC models the relationship between classes by a randomly-generated chain of them instead of treating the classes in parallel like BR. To simplify the experiment setting, we take a fix order for CC, and do not use the ensemble version. The combination aims at exploring the flexibility of choosing a major learner in the proposed framework.

The third combination takes SLR as the major learner and replaces the auxiliary learner with CC. The combination helps validate the flexibility of choosing an auxiliary learner in the proposed framework.

Note that all learners that we consider rely on SVM directly (BR, CC) or indirectly (SLR). We take LIBSVM (Chang and Lin, 2011) as the SVM solver, use the linear kernel and set the regularization parameter C to 1. This setting is as the same as the one used by MMC. Internally, SLR needs the probability output from SVM, which is done by the default probability output routine (Lin et al., 2007) in LIBSVM.

Query Criteria. We compare five different query criteria to clarify their usefulness for Algorithm 2. The criteria are random, BinMin (both described in Section 3.2), MMR (discussed in Section 4.2), our proposed HLR (Section 4.3) and SHLR (Section 4.4). Because SVM is taken as the base binary classifiers and the hinge point in the common SVM loss function is also 1, we fix the b parameter of SHLR to be 1.

5.2. Comparison using SLR/BR as Major/Auxiliary

When using SLR as the major learner and BR as the auxiliary learner in the proposed framework, we can equivalently recover the original MMC approach (Yang et al., 2009) by taking the MMR criterion. In this section, we fairly compare the MMC approach to approaches that use other query criteria. Figure 1(a) lists the resulting F1-score of the different query criteria on the `rcv1` data set. The horizontal axis indicates the number of rounds; the vertical axis is the F1-score achieved in each round. Figure 1(a) clearly demonstrates that over all the five criteria, SHLR is the best choice for `rcv1` while MMR and HLR are the runner-ups. After a few rounds, we see that MMR, HLR and SHLR start to outperform the two baseline criteria: BinMin and Random. The results justify

Table 2: ending F1-score when using SLR/BR as major/auxiliary learners

data set	`rcv1`	`Y!Ar`	`Y!Bu`	`Y!Co`	`Y!Ed`	`Y!En`	`yeast`	`scene`
starting F1-score	0.701	0.312	0.705	0.473	0.316	0.445	0.639	0.702
Random	0.764	0.396	0.738	0.528	0.402	0.517	0.647	0.730
BinMin	0.790	0.327	0.734	0.524	0.338	0.467	0.690	0.876
MMR	0.842	0.437	**0.774**	0.570	**0.425**	0.544	0.710	0.904
HLR	0.842	0.427	0.760	**0.574**	0.420	**0.551**	0.715	0.860
SHLR	**0.858**	**0.438**	0.771	0.570	**0.425**	0.543	**0.726**	**0.925**

that it is beneficial to not only take an auxiliary learner during active learning, but also query by the disagreement between the auxiliary learner and the major learner.

In the earlier rounds in Figure 1(a), HLR is slightly worse than MMR, which suggests that the margins $g^{(k)}(x')$ can be useful in determining better queries. On the other hand, in the latter rounds, the F1-scores achieved by HLR and MMR are similar, which demonstrates that HLR can be a simple but competitive alternative over MMR.

SHLR and MMR perform similarly in the earlier rounds in Figure 1(a). That is, not many instances are affected by the clipping operation in SHLR in the earlier rounds. Nevertheless, in the latter rounds, more instances can come with very negative or very positive joint ambiguity values $\hat{f}^{(k)}(x') \cdot g^{(k)}(x')$ and thus affects the MMR criterion significantly. On the other hand, the clipping operation in SHLR regularizes the extreme values when making query decisions. Figure 1(a) demonstrates that SHLR performs better than MMR during the latter rounds, and validates that the clipping operation is helpful. The figure also indicates that SHLR is better than HLR by using the detailed margin information to reach better performance in the earlier rounds.

In Table 2, we list the F1-score of the decision function f obtained in the final round. The best query criterion for each data set is marked in bold. We also rank the query criteria by their performance for each data set and list them in Table 3, with tied cases receiving the mean rank of the tied positions.

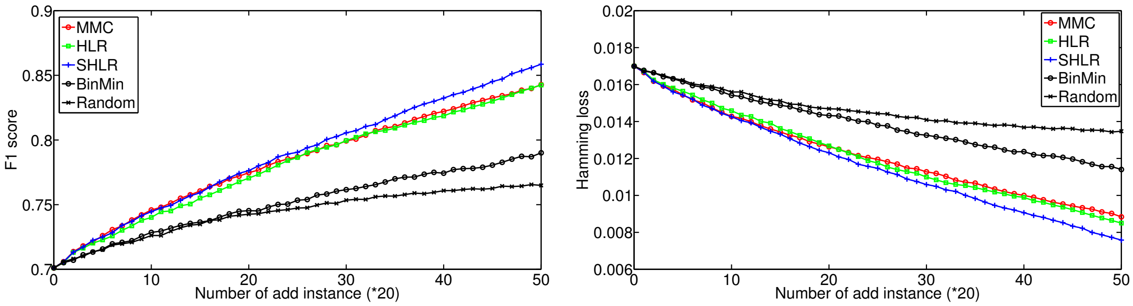
BinMin versus random. The average ranks of BinMin and random across all data sets are similarly large in Table 3, which echoes the finding in the MMC paper (Yang et al., 2009). In particular, the results suggest that querying based on a few worst-case labels cannot improve much over the (non-active) random baseline.

MMR versus BinMin and random. On the other hand, MMR outperforms BinMin and random on all data sets, which again echos that finding of the MMC paper (Yang et al., 2009).

MMR versus HLR. HLR is better than MMR on three data sets, worse on four, and ties with MMR on the other. In terms of the average rank across all data sets in Table 3, HLR is slightly worse than MMR. That is, HLR is competitive but not a better alternative over MMR.

MMR versus SHLR. In Table 2, SHLR is better than MMR on four data sets, worse on two, and ties with MMR on the other two. In terms of the average rank across all data sets in Table 3, SHLR is slightly better than MMR. The results demonstrate that SHLR can be a new state-of-the-art query criterion. SHLR is also the best query criterion in five out of the eight data sets. The results validate the importance of the proposed framework—better performance can be achieved by considering a novel criterion that appropriately use the information from the major and the auxiliary learners.

Next, we examine the results on the Hamming loss. Figure 1(b) shows the number of rounds versus the Hamming loss achieved by different query criteria on the `rcv1` data set; Table 4 lists



(*a*) F1-score

(*b*) Hamming loss

Figure 1: Using SLR/BR as major/auxiliary learners on `rcv1`

Table 3: ranking of ending-F1-score when using SLR/BR as major/auxiliary learners

data set	`rcv1`	`Y!Ar`	`Y!Bu`	`Y!Co`	`Y!Ed`	`Y!En`	`yeast`	`scene`	average
Random	5	4	4	4	4	4	5	5	4.375
BinMin	4	5	5	5	5	5	4	4	4.625
MMR	2.5	2	1	2.5	1.5	2	3	2	2.063
HLR	2.5	3	3	1	3	1	2	3	2.313
SHLR	1	1	2	2.5	1.5	3	1	1	**1.625**

Table 4: ending Hamming loss and average rank when using SLR/BR as major/auxiliary learners

data set	`rcv1`	`Y!Ar`	`Y!Bu`	`Y!Co`	`Y!Ed`	`Y!En`	`yeast`	`scene`	average rank
starting Hamming loss	.0169	.0801	.0274	.0431	.0525	.0730	.2125	.1033	
Random	.0134	.0787	.0253	.0413	.0538	.0655	.2069	.0936	4.625
BinMin	.0114	.0917	.0236	.0411	.0575	.0708	.1820	.0422	4.375
MMR	.0088	**.0693**	**.0203**	.0362	.0523	.0603	.1684	.0327	2.188
HLR	.0085	.0716	.0215	**.0360**	.0529	**.0602**	.1668	.0482	2.313
SHLR	**.0075**	.0694	**.0203**	**.0360**	**.0516**	.0605	**.1596**	**.0256**	**1.500**

the Hamming loss after the final round and shows average rank of each criterion in terms of the Hamming loss.

The observations from the Hamming loss are similar to the observations from the F1-score. SHLR remains to be the best choice and is followed by MMR and HLR. BinMin and random keeps being in the back. The results further confirm that the observed strength and weakness of the query criteria can hold across different evaluation measures.

5.3. Comparison using CC/BR as Major/Auxiliary

Next, we compare the query criteria using a different combination of major/auxiliary learners. In particular, we take Classifier Chain (CC) (Read et al., 2009) as the major learner instead of SLR. The figures on `rcv1` are similar to Figures 1(*a*) and 1(*b*) and are thus not included because of page limits.

Table 5 lists the F1-score and average rank of each criterion. We again observe that the difference between MMR, HLR and SHLR is small and SHLR enjoys a small edge. Table 6 lists the results on the Hamming loss. While SHLR is still the best in terms of the Hamming loss, HLR does not perform as well. We suspect that the inferior performance can be caused by the similarity between the CC and BR learners in nature. In particular, when the two learners are similar, the detailed margin information may be helpful in distinguishing worth-querying instances from others. HLR does not use the detailed margin information and hence could result in worse queries.

Table 5: ending F1-score and average rank when using CC/BR as major/auxiliary learners

data set	rcv1	Y!Ar	Y!Bu	Y!Co	Y!Ed	Y!En	yeast	scene	average rank
starting F1-score	0.488	0.188	0.676	0.424	0.171	0.133	0.621	0.688	
Random	0.619	0.201	0.698	0.461	**0.221**	**0.299**	0.639	0.721	3.500
BinMin	0.611	0.124	0.724	**0.472**	0.163	0.276	0.679	0.874	3.500
MMR	0.824	**0.209**	**0.759**	0.458	0.211	0.186	**0.696**	0.874	2.813
HLR	0.814	0.190	0.722	0.469	0.220	0.276	0.686	0.827	3.063
SHLR	**0.832**	0.206	**0.759**	0.464	0.218	0.187	**0.696**	**0.919**	**2.125**

Table 6: ending Hamming loss and average rank when using CC/BR as major/auxiliary learners

data set	rcv1	Y!Ar	Y!Bu	Y!Co	Y!Ed	Y!En	yeast	scene	average rank
starting Hamming loss	.0267	.0740	.0282	.0419	.0612	.0648	.2086	.1095	
Random	.0203	.0631	.0268	.0394	.0546	.0572	.1971	.0962	4.500
BinMin	.0198	.0768	.0236	.0376	**.0464**	.0571	.1766	.0423	3.250
MMR	.0091	**.0583**	..0206	**.0347**	.0483	**.0565**	.1693	.0455	1.813
HLR	.0095	.0649	.0241	.0396	.0566	.0566	.1718	.0598	3.875
SHLR	**.0084**	.0584	**.0205**	.0350	.0506	**.0565**	**.1676**	**.0276**	**1.563**

Table 7: ending F1-score and average rank when using SLR/CC as major/auxiliary learners

data set	rcv1	Y!Ar	Y!Bu	Y!Co	Y!Ed	Y!En	yeast	scene	average rank
starting F1-score	0.701	0.312	0.705	0.473	0.316	0.445	0.639	0.702	
Random	0.764	0.396	0.738	0.528	0.402	0.517	0.647	0.730	4.625
BinMin	0.819	0.403	0.737	0.550	0.398	0.492	0.705	0.928	4.125
MMR	0.841	0.431	0.766	0.568	0.417	0.540	0.720	0.910	2.563
HLR	0.847	0.427	0.761	**0.570**	0.417	**0.552**	0.726	0.844	2.313
SHLR	**0.854**	**0.432**	**0.768**	0.567	**0.420**	0.544	**0.734**	**0.934**	**1.375**

Table 8: ending Hamming loss and average rank when using SLR/CC as major/auxiliary learners

data set	rcv1	Y!Ar	Y!Bu	Y!Co	Y!Ed	Y!En	yeast	scene	average rank
starting Hamming loss	.0169	.0801	.0274	.0431	.0525	.0730	.2125	.1033	
Random	.0134	.0787	.0253	.0413	.0538	.0655	.2069	.0936	4.750
BinMin	.0102	.0773	.0234	.0394	.0545	.0677	.1719	.0243	4.000
MMR	.0090	.0703	.0208	.0362	.0522	.0615	.1622	.0306	2.438
HLR	.0090	.0718	.0214	.0363	.0529	**.0601**	.1605	.0540	2.688
SHLR	**.0084**	**.0700**	**.0206**	**.0361**	**.0520**	.0607	**.1543**	**.0223**	**1.125**

An interesting observation comes from comparing the results from Tables 4 and 6. In particular, the best Hamming loss of the CC/BR combination is better than the best one of the SLR/BR combination across almost all the data sets. On the other hand, when comparing the results from Tables 2 and 5, the SLR/BR combination appears to be a better choice. Thus, the flexibility in the proposed framework is useful: an appropriate combination of major/auxiliary learners can be chosen to improve a particular evaluation measure of interest.

5.4. Comparison using SLR/CC as Major/Auxiliary

Next, we compare the query criteria using SLR/CC as major/auxiliary learners. Table 7 lists the F1-score and average rank of each criterion; Table 8 lists the Hamming loss. The figures on rcv1 are similar to Figures 1(*a*) and 1(*b*) and are thus not included because of page limits. The observations from the figures and tables are consistent with the observations using SLR/BR. SHLR clearly reaches the best performance, MMR and HLR are similar, and BinMin and random cannot do well.

When comparing the results of SLR/CC (Tables 7 and 8) with the results of SLR/BR (Tables 2 and 4), we see that the SLR/CC combination outperforms SLR/BR on Y!En, yeast and scene data sets in terms of both the F1-score and Hamming loss. The results again justify that the proposed general framework creates a room for improving active learning performance by appropriately choosing major/auxiliary learners.

6. Conclusion

We extended the state-of-the-art MMC approach to a more general framework called active learning with auxiliary learner. We then studied the properties of MMC's query criterion, MMR, and designed two different query criteria to alleviate the extreme-value sensitivity of MMR. The first criterion, HLR, does not require any margin information from the auxiliary learner; the second criterion, SHLR, removes the influence on extreme margin-values by clipping.

We conducted experiments to fairly compare the query criteria on various real-world data sets. The experiments showed that SHLR is usually the best query criterion across different data sets and different combinations of major/auxiliary learners. Thus, SHLR should be the most favorable choice in practice. In addition, we observed that the flexibility of the proposed framework allows properly using different combinations of major/auxiliary learners to reach better performance. The observation justified the validity and usefulness of the framework. One important future direction is to explore how to properly choose a suitable combination of major/auxiliary learners *before* running the active learning algorithm.

Acknowledgements

We thank the anonymous reviewers and the first author's thesis committee (Hung, 2011) for valuable suggestions. This work is supported by National Science Council of Taiwan via the grant NSC 99-2628-E-002-017.

References

Klaus Brinker. On active learing with multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, pages 206–213, 2006.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30, 2004.

Rafał Grodzicki, Jacek Mańdziuk, and Lipo Wang. Improved multi-label classification with neural networks. In *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, pages 409–416, 2008.

Chen-Wei Hung. Multi-label active learning with auxiliary learner. Master's thesis, National Taiwan University, 2011.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Xuchun Li, Lei Wang, and Eric Sung. Multi-label SVM active learning for image classification. In *Proceedings of the 11th International Conference on Image Processing*, pages 2207–2210, 2004.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Journal of Machine Learning Research*, 68:267–276, 2007.

Andrew McCallum. Multi-label text classification with a mixture model trained by EM. In *Proceedings of the Text Learning Workshop on 15th National Conference of Artificial Intelligence*, 1999.

Andrew McCallum and Kamal Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 350–358, 1998.

Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Multi-label classification using ensembles of pruned sets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 995 –1000, 2008.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, pages 441–448, 2001.

Burr Settles. Active learning literature survey. Technical report, 2010.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 287–294, 1992.

Farbound Tai and Hsuan-Tien Lin. Multi-label classification with principle label space transformation. In *Proceedings of the 2nd International Workshop on Learning from Multi-Label Data*, pages 45–52, 2010.

Simon Tong. *Active Learning: Theory and Applications.* PhD thesis, Stanford University, 2001.

Simon Tong, Daphne Koller, and Pack Kaelbling. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:999–1006, 2000.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehouse and Mining*, 3:1–13, 2007.

Vladimir N. Vapnik. *The nature of statistical learning theory*. 1995.

Mei Wang, Xiangdong Zhou, and Tat-Seng Chua. Automatic image annotation via local multi-label classification. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval*, pages 17–26, 2008.

Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining*, pages 917–926, 2009.

Cha Zhang and Tsuhan Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4:260–268, 2002.

Min-Ling Zhang and Zhi-Hua Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18: 1338 –1351, 2006.

Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40:2038–2048, 2007.