Data Min Knowl Disc
DOI 10.1007/s10618-013-0306-1

Repeated labeling using multiple noisy labelers

**Panagiotis G. Ipeirotis · Foster Provost ·
Victor S. Sheng · Jing Wang**

Received: 9 September 2010 / Accepted: 27 February 2013

Abstract This paper addresses the repeated acquisition of labels for data items when the labeling is imperfect. We examine the improvement (or lack thereof) in data quality via repeated labeling, and focus especially on the improvement of training labels for supervised induction of predictive models. With the outsourcing of small tasks becoming easier, for example via Amazon's Mechanical Turk, it often is possible to obtain less-than-expert labeling at low cost. With low-cost labeling, preparing the unlabeled part of the data can become considerably more expensive than labeling. We present repeated-labeling strategies of increasing complexity, and show several main results. (i) Repeated-labeling can improve label quality and model quality, but not always. (ii) When labels are noisy, repeated labeling can be preferable to single labeling even in the traditional setting where labels are not particularly cheap. (iii) As soon as the cost of processing the unlabeled data is not free, even the simple strategy of labeling everything multiple times can give considerable advantage. (iv) Repeatedly labeling a carefully chosen set of points is generally preferable, and we present a set

Responsible editor: Johannes Fürnkranz

P. G. Ipeirotis (✉) · F. Provost · J. Wang
Department of Information, Operations, and Management Sciences, Leonard N. Stern School of Business, New York University, New York, NY, USA
e-mail: panos@stern.nyu.edu

F. Provost
e-mail: fprovost@stern.nyu.edu

J. Wang
e-mail: jwang5@stern.nyu.edu

V. S. Sheng
Department of Computer Science, University of Central Arkansas, Conway, USA
e-mail: ssheng@uca.edu

Published online: 16 March 2013

of robust techniques that combine different notions of uncertainty to select data points for which quality should be improved. The bottom line: the results show clearly that when labeling is not perfect, selective acquisition of multiple labels is a strategy that data miners should have in their repertoire; for certain label-quality/cost regimes, the benefit is substantial.

Keywords Active learning · Data selection · Data preprocessing · Classification · Human computation · Repeated labeling · Selective labeling

1 Introduction

There are various costs associated with the *preprocessing* stage of the KDD process, including costs of acquiring features, formulating data, cleaning data and obtaining expert labeling of data (Turney 2000; Weiss and Provost 2003). For example, in order to build a model to recognize whether two products described on two web pages are the same, one must extract the product information from the pages, formulate features for comparing the two along relevant dimensions, verify that the features are correct for these products, and label product pairs as identical or not; this process involves costly manual intervention at several points. To build a model that recognizes whether an image contains an object of interest, one first needs to take pictures in appropriate contexts, sometimes at substantial cost.

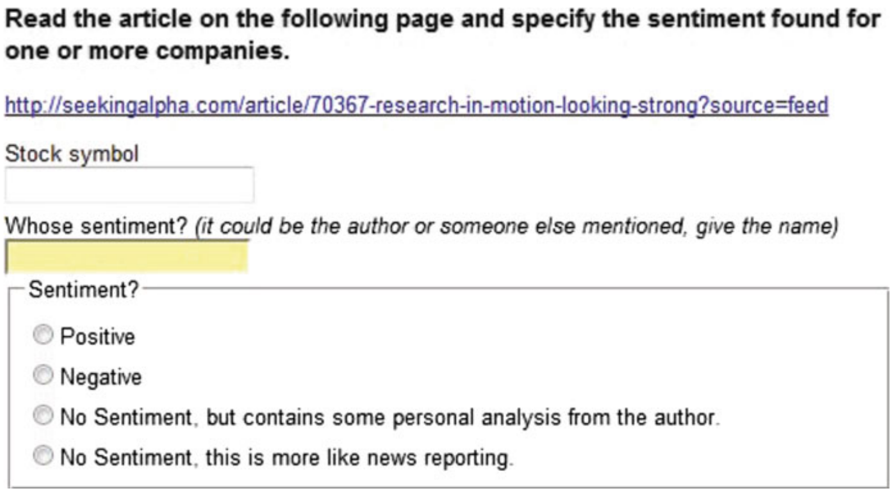
This paper focuses on problems where it is possible to obtain certain (noisy) data values (“labels”) relatively cheaply, from multiple sources (“labelers”). A main focus of this paper is the use of these values as training labels for supervised modeling.[1] For our two examples above, once we have constructed the unlabeled portion of the data point, for relatively low cost one can obtain non-expert opinions on whether two products are the same or whether an image contains a person or a storefront or a building. These cheap labels may be noisy due to lack of expertise, dedication, interest, or other factors. Our ability to perform non-expert labeling cheaply and easily is facilitated by on-line micro-outsourcing systems, such as Amazon’s Mechanical Turk (Snow et al. 2008)[2], which match workers with arbitrary (well-defined) tasks, as well as by creative labeling solutions like the ESP game (von Ahn and Dabbish 2004).[3] These on-line outsourcing systems allow hundreds (or more) of human labelers to look at objects (i.e., articles) and label them, using an interface such as the one in Fig. 1. Using such marketplaces, it is possible to outsource small parts of the process at very low cost—parts that prior to the introduction of such systems would have incurred much higher (in-house) cost, or would have been avoided altogether.

In the face of noisy labeling, as the ratio increases between the cost of preprocessing a data point and the cost of labeling it, it is natural to consider *repeated labeling* (or re-labeling): obtaining multiple labels for some or all data points. This paper explores

[1] This setting is in direct contrast to the setting motivating active learning and semi-supervised learning, where unlabeled points are relatively inexpensive, but labeling is expensive.

[2] http://www.mturk.com

[3] http://www.espgame.org



Read the article on the following page and specify the sentiment found for one or more companies.

http://seekingalpha.com/article/70367-research-in-motion-looking-strong?source=feed

Stock symbol

Whose sentiment? *(it could be the author or someone else mentioned, give the name)*

Sentiment?

○ Positive

○ Negative

○ No Sentiment, but contains some personal analysis from the author.

○ No Sentiment, this is more like news reporting.

Fig. 1 An example of a micro-task submitted to the Amazon Mechanical Turk marketplace

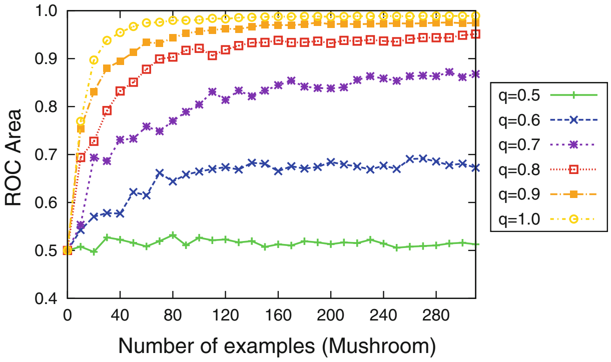


Fig. 2 Learning curves under different quality levels of training data (q is the probability of a label being correct)

whether, when, and for which data points one should obtain multiple, noisy training labels, as well as what to do with them once they have been obtained. Consider a data miner facing the question of how to expend her data preprocessing budget: (i) acquiring multiple cheap labels for existing data to reduce labeling error versus (ii) acquiring potentially informative new data points at higher cost.

Given a data set for supervised classifier induction, ceteris paribus, noisy labels will decrease the generalization performance of induced models (e.g., Quinlan 1986). Figure 2 shows learning curves under different labeling qualities for the classic *mushroom* data set (see Sect. 4.1). Specifically, for the different quality levels of the *training* data,[4] the figure shows learning curves relating the classification accuracy—which is measured by the area under the ROC curve in this paper—of a Weka J48 model (Witten and Frank 2005) to the number of training data. This data set is illustrative because

[4] The test set has perfect quality with zero noise.

with zero-noise labels one can achieve perfect classification after some training, as demonstrated by the $q = 1.0$ curve. Figure 2 illustrates that the performance of a learned model depends both on the quality of the training labels and on the number of training examples. Of course if the training labels are uninformative ($q = 0.5$), no amount of training data helps. As expected, for a given labeling quality ($q > 0.5$), more training examples lead to better performance, and the higher the quality of the training data, the better the performance of the learned model. However, the relationship between the two factors is complex: the marginal increase in performance for a given change along each dimension is quite different for different combinations of values for both dimensions. To this, one must overlay the different costs of acquiring only new labels versus whole new examples, as well as the expected improvement in quality when acquiring multiple new labels.

This paper makes several contributions. First, under gradually weakening assumptions, we assess the impact of repeated-labeling on the quality of the resultant labels, as a function of the number and the individual qualities of the labelers. We derive analytically the conditions under which repeated-labeling will be more or less effective in improving resultant label quality. We then consider the effect of repeated-labeling on the accuracy of supervised modeling. As demonstrated in Fig. 2, the relative advantage of increasing the quality of labeling, as compared to acquiring new data points, depends on the position on the learning curves. We show that there are times when repeated-labeling is preferable compared to getting labels for unlabeled data items, even in the case where one ignores the cost of obtaining the unlabeled part of a data item. Furthermore, when we do consider the cost of obtaining the unlabeled portion, repeated-labeling can give considerable advantage.

We present a comprehensive experimental analysis of the relationships between quality, cost, and technique for repeated-labeling. The results show that even a straightforward, round-robin technique for repeated-labeling can give substantial benefit over single-labeling. We then show that selectively choosing the data items to re-label yields substantial extra benefit. A key question is: How should we select data points for repeated-labeling? We argue that the *uncertainty* of a data item's label is a good indicator of where we should allocate our (repeated) labeling efforts. We present various techniques for measuring the uncertainty, and show how these various techniques improve over round-robin repeated labeling.

Although this paper covers a good deal of ground, there is much left to be done to understand how best to label using multiple, noisy labelers; so, the paper closes with a summary of the key limitations, and some suggestions for future work.

2 Related work

Repeatedly labeling the same data point is practiced in applications where labeling is not perfect (e.g., Smyth et al. 1994a,b). We are not aware of a systematic assessment of the relationship between the resultant quality of supervised modeling and the number of, quality of, and method of selection of data points for repeated-labeling. To our knowledge, the typical strategy used in practice is what we call "round-robin" repeated-labeling, where cases are given a fixed number of labels—so we focus considerable attention in the paper to this strategy. A related important problem is how in practice to assess the generalization performance of a learned model with uncertain labels (Smyth et al. 1994b), which we do not consider in this paper. Prior research has addressed important problems necessary for a full labeling solution that uses multiple noisy labelers, such as estimating the quality of labelers (Dawid and Skene 1979; Donmez et al. 2009, 2010; Smyth 1996; Smyth et al. 1994b), and learning with uncertain labels (Lugosi 1992; Silverman 1980; Smyth 1995). Raykar et al. (2009, 2010) presented a technique that builds on and expands this line of work, and shows how to integrate the process of concurrently building classifier and learning the quality of the labelers. So we treat these topics quickly when they arise, and lean on the prior work.

Repeated-labeling using multiple noisy labelers is different from multiple label classification (Boutell et al. 2004; McCallum 1999), where one example could have multiple *correct* class labels. As we discuss in Sect. 8, repeated-labeling can apply regardless of the number of true class labels. The key difference is whether the labels are noisy. A closely related problem setting is described by Jin and Ghahramani (2002). In their variant of the multiple label classification problem, each example presents itself with a set of mutually exclusive labels, one of which is correct. The setting for repeated-labeling has important differences: labels are acquired (at a cost); the same label may appear many times, and the true label may not appear at all. Again, the level of error in labeling is a key factor.

The consideration of data acquisition costs has seen increasing research attention, both explicitly (e.g., cost-sensitive learning; Turney 2000), (utility-based data mining; Provost 2005) and implicitly, as in the case of active learning (Cohn et al. 1994). Turney (2000) provides a short but comprehensive survey of the different sorts of costs that should be considered, including data acquisition costs and labeling costs. Most previous work on cost-sensitive learning does not consider labeling cost, assuming that a fixed set of labeled training examples is given, and that the learner cannot acquire additional information during learning (e.g., Domingos 1999; Elkan 2001; Turney 1995).

Active learning (Cohn et al. 1994) focuses on the problem of costly label acquisition, although often the cost is not made explicit. Active learning (cf., optimal experimental design, Whittle 1973) uses the existing model to help select additional data for which to acquire labels (Baram et al. 2004; Margineantu 2005; Saar-Tsechansky and Provost 2004). The usual problem setting for active learning is in direct contrast to the setting we consider for repeated-labeling. For active learning, the assumption is that the cost of labeling is considerably higher than the cost of obtaining unlabeled examples (essentially zero for "pool-based" active learning).

Some previous work studies data acquisition cost explicitly. For example, several authors (Kapoor and Greiner 2005; Lizotte et al. 2003; Melville et al. 2004, 2005; Saar-Tsechansky and Provost 2004; Weiss and Provost 2003; Zhu and Wu 2005) study the costly acquisition of feature information, assuming that the labels are known in advance. Saar-Tsechansky et al. (2009) consider acquiring both costly feature and label information.

None of this prior work considers selectively obtaining multiple labels for data points to improve labeling quality, and the relative advantages and disadvantages for improving model performance. An important difference from the setting for traditional active learning is that labeling strategies that use multiple noisy labelers have access to potentially relevant additional information. In our setting, each example has a label multiset which is composed of all the labels that we have acquired for this example. The multisets of existing labels intuitively should play a role in determining the examples for which to acquire additional labels. For example, presumably one would be less interested in getting another label for an example that already has a dozen identical labels, than for one with just two, conflicting labels.

There is a relationship between our work and ensemble learning. However, there is a crucial difference between ensemble learning and crowdsourcing: In ensemble learning, we can create many classifiers and get each classifier to label each example. In crowdsourcing, there is a cost every time we ask a worker to (re-)label an example. Crowdsourcing therefore can be viewed as similar to "budget-constrained" ensemble application, where each ensemble classification decision has a cost. There is another, more subtle difference: in crowdsourcing, each worker has constrained capacity; worker contributions tend to follow a power-law distribution: a large number of workers contribute just a few labels, which makes the estimation of their quality, biases, etc. more challenging.

One specific area of research applying ensembles is particularly closely related: research on noise elimination/mitigation using ensembles. Brodley and Friedl (1999) apply a set of learning algorithms to create an ensemble of classifiers. They decide that certain instances are likely noise and filter them from the data by analyzing the estimations of the classifiers as well as the reported class label. Verbaeten and Assche (2003) study a number of filtering techniques based on ensemble methods like cross-validated committees, bagging and boosting. Rebbapragada and Brodley (2007) use clustering to estimate probabilities over the class labels and then use the confidence on the reported label as a weight during training to mitigate noise in the data.

The work presented in this paper is an extension to a previously published conference paper (Sheng et al. 2008). In the present paper, we present and evaluate two additional algorithms for selectively allocating labeling effort (*NLU* and *NLMU*; see Sects. 6.3 and 6.4). These new algorithms have better theoretical justification and often outperform the techniques in (Sheng et al. 2008). The results in this paper are all new, although many qualitative conclusions are the same as in the conference version. Specifically, in this paper we use as measure of predictive performance the area under the ROC curve (AUC), which is a more robust indicator of performance than the accuracy metric that we used in (Sheng et al. 2008), especially for imbalanced data sets. Additional completely new results include the following. In Sect. 6.4.1 we provide a justification why a technique that relies purely on model uncertainty can improve data quality and predictive performance; this contrasts with the implication in (Sheng et al. 2008) that the reason it works is because it is essentially doing active learning. We also present extensive experimental results that demonstrate when soft-labeling (see Sect. 3.3 for definition) can be beneficial in a setting with noisy labelers (Sect. 5.3) and examine the effect of weighted sampling for selecting examples to label (Sect. 6.6.2). In addition, we show the performance our proposed strategies on a real-world data set.

Since the publication of the conference paper, a significant amount of work has been published in the area. Donmez and Carbonell (2008) presented an active learning model where the labelers are imperfect, have expertise on different parts of the space, and have various costs, depending on the uncertainty of the labeler: Donmez et al. focus on the problem of selecting the labelers to ask for given examples. In contrast to our work, Donmez and Carbonell (2008) do not use repeated labeling to improve the quality of the data. In subsequent extensions, Donmez et al. (2009, 2010) learn the labeler quality in order to better guide the labeling strategy and show that techniques that exploit the labeler quality perform better than the round robin repeated labeling strategy (see Sect. 5). [There is no comparison against the selective labeling strategies that we presented in (Sheng et al. 2008) and which we also discuss in Sect. 6.]

Carpenter (2008) presented a Bayesian model for estimating the quality of the labelers, and Whitehill et al. (2009) presented a graphical model in which both labelers vary in quality and the examples have varying degree of difficulty for being classified. Ipeirotis et al. (2010) described an algorithm that measures the inherent quality of a labeler using a scalar metric that takes into consideration the misclassification costs and also separates the error computation from the potential biases exhibited by the labelers. Again, the focus is not on cost-sensitive acquisition of data, but rather on building Bayesian models that account for the expertise of the workers and the difficulty of labeling given examples.

3 Repeated labeling: the basics

Figure 2 illustrates that the quality of the labels can have a marked effect on classification accuracy. Intuitively, using repeated-labeling to shift from a lower-q curve to a higher-q curve can, under some settings, improve learning considerably. In order to treat this more formally, we first introduce some terminology and simplifying assumptions.

3.1 Notation and assumptions

We consider a problem of supervised induction of a (binary) classification model. The setting is the typical one, with some important exceptions. For each training example $\langle y_i, x_i \rangle$, procuring the *unlabeled* “feature” portion, x_i, incurs cost C_U. The action of *labeling* the training example with a label y_i incurs cost C_L. For simplicity, we assume that each cost is constant across all examples. Each example $\langle y_i, x_i \rangle$ has a true label y_i, but labeling is error-prone. Specifically, each label y_{ij} comes from a labeler j exhibiting an individual labeling quality p_j, which is $Pr(y_{ij} = y_i)$; since we consider the case of binary classification, the label assigned by labeler j will be incorrect with probability $1 - p_j$.

In the current paper, we work under a set of assumptions that allows us to focus on a certain set of problems that arise when labeling using multiple noisy labelers. First, we assume that $Pr(y_{ij} = y_i | x_i) = Pr(y_{ij} = y_i) = p_j$, that is, individual labeling quality is independent of the specific data point being labeled. Second, we assume labelers are conditionally independent from each other. Note that we do *not* assume that all labelers have the same quality: each labeler has an individual quality p_j. We sidestep the issue of knowing p_j: the techniques we present do not rely on this knowledge and are largely agnostic about the quality of the labelers. Inferring p_j

accurately should lead to improved techniques; Dawid and Skene (1979) and Smyth et al. (1996, 1994b) have shown how to use an expectation-maximization framework for estimating the quality of labelers. We also assume for simplicity that each labeler j only gives one label, but that is not a restrictive assumption in what follows. We further discuss limitations and directions for future research in Sect. 8.

3.2 Majority voting and label quality

To investigate the relationship between labeler quality, number of labels, and the overall quality of labeling using multiple labelers, we start by considering the case where, for induction, each repeatedly-labeled example is assigned a *single* "integrated" label $\hat{y}_i$, inferred from the individual y_{ij}'s by majority voting. For simplicity, and to avoid having to break ties, we assume that we always obtain an odd number of labels. The quality $q_i = Pr(\hat{y}_i = y_i)$ of the integrated label $\hat{y}_i$ will be called the *integrated quality*. Where no confusion will arise, we will omit the subscript i for brevity and clarity.

Consider first the case where all labelers are independent and exhibit the same quality, that is, $p_j = p$ for all j. (We would like to stress that the techniques in the paper *do not* rely on this assumption and here we just want to illustrate the improvement in quality when using repeated labeling.). Using $2N + 1$ labelers with uniform quality p, the integrated labeling quality q is:

$$q = Pr(\hat{y} = y) = \sum_{i=0}^{N} \binom{2N+1}{i} \cdot p^{2N+1-i} \cdot (1-p)^i \qquad (1)$$

which is the sum of the probabilities that we have more correct labels than incorrect (the index i corresponds to the number of incorrect labels).

Not surprisingly, from the formula above, we can infer that the integrated quality q is greater than p only when $p > 0.5$. When $p < 0.5$, we have an adversarial setting where $q < p$, and, not surprisingly, the quality decreases as we increase the number of labelers.

Figure 3 demonstrates the analytical relationship between the integrated quality and the number of labelers, for different individual labeler qualities. As expected, the

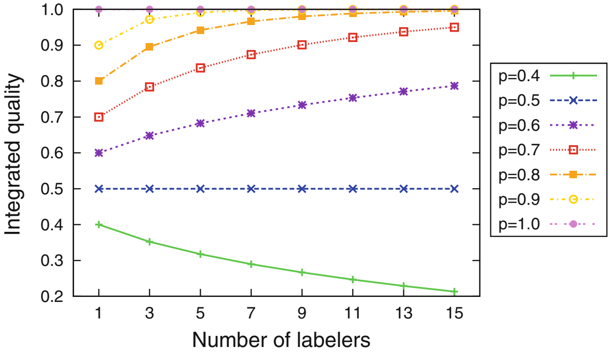


Fig. 3 The relationship between integrated labeling quality, individual quality, and the number of labelers

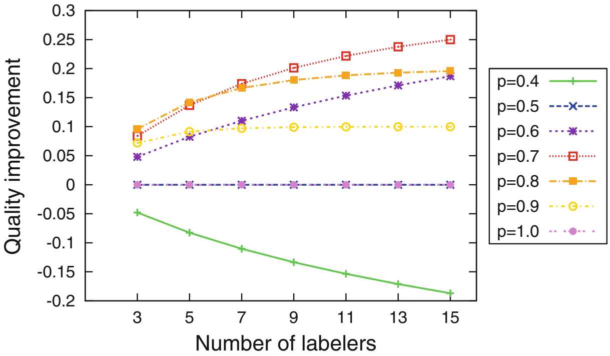


Fig. 4 Improvement in integrated quality compared to single-labeling, as a function of the number of labelers, for different labeler qualities

integrated quality improves with larger numbers of labelers, when the individual labeling quality $p > 0.5$; however, the marginal improvement decreases as the number of labelers increases. Moreover, the benefit of getting more labelers also depends on the underlying value of p. Figure 4 shows how integrated quality q increases compared to the case of single-labeling, for different values of p and for different numbers of labelers. If labeler quality is too high, there is very little room for improvement; if labeler quality is too low, each additional labeler provides limited additional information. In both cases, not much is gained from having more labelers. Therefore, we would expect multiple labeling methods to be most beneficial when labelers have moderate quality (e.g., $p = 0.6$; $p = 0.7$). For example, when $p = 0.9$, there is little benefit when the number of labelers increases from 3 to 11. However, when $p = 0.7$, going just from single labeling to three labelers increases integrated quality by about 0.1, which in Fig. 2 would yield a substantial upward shift in the learning curve (from the $q = 0.7$ to the $q = 0.8$ curve); in short, a small amount of repeated-labeling can have a noticeable effect for moderate levels of noise.

Therefore, for cost-effective labeling using multiple noisy labelers we need to consider: (a) the effect of the integrated quality q on learning, and (b) the number of labelers required to increase q under different levels of labeler quality p; we will return to this later, in Sects. 5 and 6.

3.3 Uncertainty-preserving labeling

Majority voting is a simple and straightforward method for integrating the information from multiple labels, but clearly with its simplicity comes a potentially serious drawback: information is lost about label uncertainty (LU). In principle, an alternative is to move to some form of *"soft" labeling*, with the multiset of labels resulting in a probabilistic label (for example Smyth 1995). One concern with soft labeling is that even in cases where, in principle, modeling techniques should be able to incorporate soft labeling directly (which would be true for techniques such as naive Bayes, logistic regression, tree induction, and beyond), existing software packages do not accommodate soft labels. Fortunately, we can finesse this.

Consider the following straightforward method for integrating labels from multiple labelers. For each unlabeled example x_i, the *multiplied examples* (ME) procedure considers the multiset of existing labels $L_i = \{y_{ij}\}$. *ME* creates one replica of x_i labeled by each unique label appearing in L_i. Then, for each replica, *ME* assigns a weight $1/|L_i|$, where $|L_i|$ is the number of occurrences of this label in L_i. Alternative ways of integrating the labels may rely on computing the uncertainty about the class of the example and assigning the weights appropriately. These weighted replicas can be used in different ways by different learning algorithms: for instance, in algorithms that take weights directly, such as cost-sensitive tree (Ting 2002), or in techniques like naive Bayes that naturally incorporate uncertain labels. Moreover, any importance-weighted classification problem can be reduced to a uniform-weighted classification problem (Zadrozny et al. 2003), often performing better than hand-crafted weighted-classification algorithms. We examine the effect of uncertainty-preserving labeling (a.k.a. "soft labeling") in Sects. 5.3 and 6.6.1.

4 Experimental setup and design

The previous section examined when repeated-labeling can improve data quality. We now consider when repeated-labeling should be chosen for *modeling*. What is the relationship to label quality? (Since we see that for $p = 1.0$ and $p = 0.5$, repeated-labeling adds no value.) How cheap (relatively speaking) does labeling have to be? For a given cost setting, is repeated-labeling much better or only marginally better? Can selectively choosing data points to label improve performance?

4.1 Experimental setup

Practically speaking, the answers to these questions rely on the empirical distributions being modeled, and so we shift to an empirical analysis based on experiments with both simulated and real labelers.

To investigate the questions above, we first present experiments on 8 real-world data sets from (Blake and Merz 1998) and (Zheng and Padmanabhan 2006). These data sets were chosen because they are classification problems with a moderate number of examples, allowing the development of learning curves based on a large numbers of individual experiments. Furthermore, we use only data sets for which the performance (AUC) was above 0.7 running with cross-validation on the original full data set—so that there is room to differentiate different labeling strategies. The data sets are described in Table 1. If necessary, we convert the target to binary (for *thyroid* we keep the negative class and integrate the other three classes into positive; for *splice*, we integrate classes IE and EI; for *waveform*, we integrate classes 1 and 2).

For each data set, 30 % of the examples are held out, in every run, as the test set from which we calculate generalization performance. The rest is the "pool" from which we acquire unlabeled and labeled examples. To simulate noisy label acquisition, we first hide the labels of all examples for each data set. At the point in an experiment when

Table 1 The eight data sets used in the experiments: the numbers of attributes and examples in each, and the split into positive and negative examples

Data set	#Attributes	#Examples	Pos	Neg
kr-vs-kp	37	3,196	1,669	1,527
mushroom	22	8,124	4,208	3,916
sick	30	3,772	231	3,541
spambase	58	4,601	1,813	2,788
splice	61	3,190	1,535	1,655
thyroid	30	3,772	291	3,481
tic-tac-toe	10	958	332	626
waveform	41	5,000	1,692	3,308

a label is acquired, we generate a label according to the labeler quality p: we assign the example's original label with probability p and the opposite value with probability $1 - p$.

After obtaining the labels, we add them to the training set to induce a classifier. For the results presented, models are induced with J48, the implementation of C4.5 (Quinlan 1992) in WEKA (Witten and Frank 2005). The classifier is evaluated on the test set (with the true labels). Each experiment is repeated 50 times with a different random data partition, and average results are reported.

4.2 Design choices for repeated labeling

In our experimental setup, we examine a set of basic design choices that can be varied to create different repeated labeling algorithms. Specifically, the design choices that we explore are the following:

- Choice of next example to (re-)label:
  - Selection policy: The choice of the next example to (re-)label is based on a policy that determines the priority (e.g., uncertainty score) with which we choose the next example. First, in Sect. 5 we explore policies that assume uniform prioritization across all examples. In Sect. 6 we examine versions that use heterogeneous (and dynamic) prioritization schemes.
  - Deterministic versus Sampled order: Is the choice of the next example to (re-)label based on a deterministic order or is the next example sampled from a distribution over the data set? While the selection policy computes the labeling priority of each example, the deterministic/sampled order reflects the way that this information is utilized. For deterministic order, we choose the examples with highest uncertainty scores; for sampled order, each example is sampled probabilistically, with a probability proportional to its uncertainty (Sect. 6.6.2). Most of the results in this paper are based on deterministic order, but we also explore sampling schemes in Sect. 6.6.2.
- Choice of labeling scheme: Use a "hard" label for each example, inferred using majority voting? Or use a "soft" label that preserves the uncertainty about the

class label? We examine the effect of uncertainty-preserving labeling (a.k.a. “soft labeling”) in Sects. 5.3 and 6.6.1.

5 Basic repeated-labeling strategies

Figure 5 shows the learning curves for the *mushroom* data set. As a case in point, assume that we have processed 50 examples with quality $q = 0.8$; we have different choices for how to proceed. The two basics choices are: (i) get more examples with a single label each (horizontal arrow), or (ii) improve the quality of the existing examples by repeatedly labeling the existing examples (vertical arrow).

We call the first strategy *single labeling* (*SL*): getting as many examples as possible, one label for each example. The second strategy is the most straightforward repeated labeling strategy: assign additional labels to the labeled examples, in a round-robin fashion. We can keep *adding labels* to a *fixed* number of examples, until exhausting our labeling budget. We call this strategy *fixed round-robin* (*FRR* in short). The *FRR* strategy strategy corresponds to the vertical arrow in Fig. 5. A slight generalization of *FRR* is to always give the next label to the example with the fewest labels; we call this labeling strategy *generalized round-robin* (*GRR* in short). For the experiments below, we assume that we receive a new example every k labels, so in this case *GRR* simply assigns k labels to each new example. We evaluate these basic labeling strategies (*SL*, *FRR*, and *GRR*) in the next sections.

5.1 Fixed round-robin strategy *FRR*

We assume for this section that we select randomly a *fixed* set of examples from the unlabeled pool and *FRR* repeated-labeling re-labels examples in a *fixed round-robin* fashion: Specifically, given a fixed set L of to-be-labeled examples (a subset of the entire set of examples) the next label goes to the example in L with the fewest labels, with ties broken according to some rule (in our case, by cycling through a fixed order).

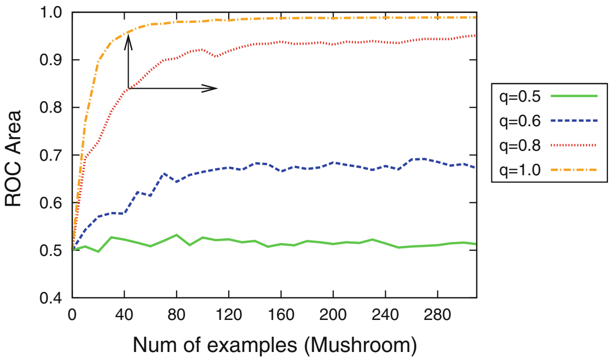
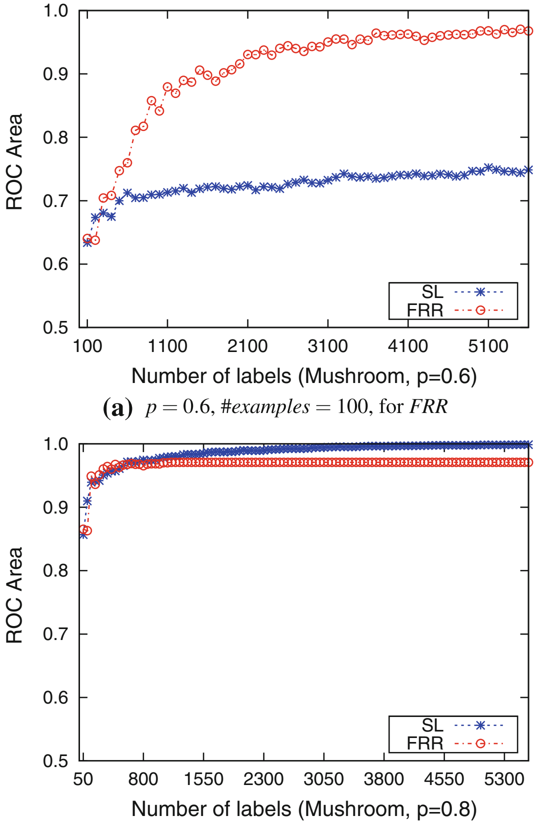


Fig. 5 Learning curves under different quality levels of training data (q is the probability of a label being correct)



(a) $p = 0.6$, *#examples* $= 100$, for *FRR*

(b) $p = 0.8$, *#examples* $= 50$, for *FRR*

Fig. 6 Comparing the increase in accuracy for the *mushroom* data set as a function of the number of labels acquired, when the cost of an unlabeled example is negligible. The simplest repeated-labeling strategy *FRR* with majority vote starts with an existing set of examples and only acquires additional labels for them, and single labeling (*SL*) acquires additional examples. Other data sets show similar results

Figure 6 shows the generalization performance of *FRR* with majority vote, compared to that of single labeling, *SL*, as a function of the *number of labels* acquired for a fixed labeler quality. Both *FRR* and *SL* start with the same number of single-labeled examples. Then, *FRR* starts acquiring additional labels *only* for the *existing* examples, while *SL* acquires new examples and labels them (once).

Generally, the decision regarding whether to invest in another whole training example or another label depends on the gradient of generalization performance, as a function of obtaining another label or a new example. (We will return to this when we discuss future work.) Figure 6 shows, for our example problem, scenarios where each strategy is preferable to the other. Consider Fig. 6a. From Fig. 2 we see that for $p = 0.6$, and with 100 examples, there is a lot of headroom for repeated-labeling to improve generalization performance by improving the overall labeling quality. Figure 6a indeed shows that for $p = 0.6$, repeated-labeling does improve generalization performance (per label) as compared to single-labeling new examples. On the other hand, for high initial quality or steep sections of the learning curve, *FRR* may not compete with single labeling. Figure 6b shows that single labeling performs better than *FRR* when we have a fixed set of 50 training examples with labeling quality $p = 0.8$. Particularly, *FRR* could not further improve its performance after a certain amount of labeling (cf., the $q = 1$ curve in Fig. 2). This happens because each of the fixed set of examples ends up having perfect quality (so further repeated labeling cannot help) and the size of the fixed example set (for training) simply is not sufficient to reach higher accuracy.

The results for other data sets are similar to Fig. 6: under noisy labels, the fixed round-robin repeated-labeling *FRR* can perform better than single-labeling when there are enough training examples, i.e., after the learning curves are not so steep (cf., Fig. 2).

5.2 Generalized round-robin strategies *GRR*, introducing costs

We illustrated above that repeated-labeling (viz., *FRR*) is a viable alternative to single-labeling, when the labels that we get are noisy. In our comparison of *FRR* with *SL*, we effectively ignored the cost of acquiring the "feature" part of each new example for *SL*. However, as described in the introduction, often the cost of (noisy) label acquisition C_L is low compared to the cost C_U of acquiring an unlabeled example. In this case, clearly repeated-labeling should be considered: using multiple labels can shift the learning curve up significantly.

We now study the setting where we have the choice of either:

- acquiring a new training example for cost $C_U + C_L$, (C_U for the *unlabeled* portion, and C_L for the label), or
- get another label for an existing example for cost C_L.

To compare any two strategies on equal footing, we calculate generalization performance "per unit cost" of acquired data; we then compare the different strategies for combining multiple labels, under different individual labeling qualities. We start by defining the data acquisition cost C_D:

$$C_D = C_U \cdot T_r + C_L \cdot N_L \tag{2}$$

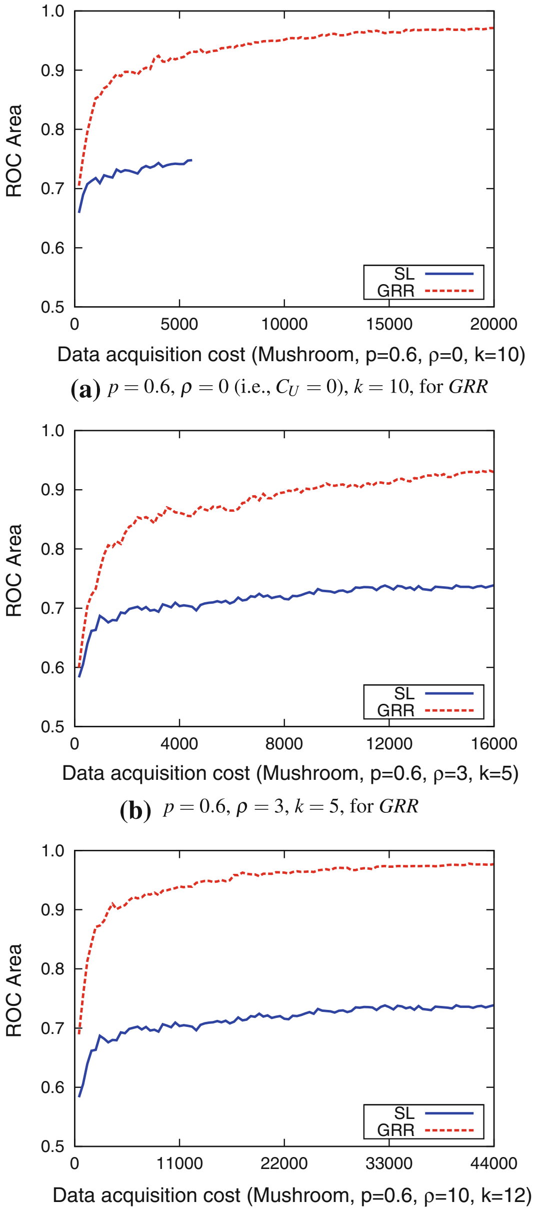
to be the sum of the cost of acquiring T_r unlabeled examples ($C_U \cdot T_r$), plus the cost of acquiring the associated N_L labels ($C_L \cdot N_L$). For single labeling we have $N_L = T_r$, but for repeated-labeling $N_L > T_r$.

We extend the setting of Sect. 5.1 slightly and we consider the generalized round-robin strategy, *GRR*, which can acquire and label new examples; single labeling *SL* is unchanged. For each new example acquired, repeated labeling acquires a fixed number of labels k, and in this case $N_L = k \cdot T_r$. Thus, for *GRR*, in these experiments the cost setting can be described compactly by the cost ratio $\rho = \frac{C_U}{C_L}$, and in this case $C_D = \rho \cdot C_L \cdot T_r + k \cdot C_L \cdot T_r$, i.e.,

$$C_D \propto \rho + k \tag{3}$$

Figure 7 shows the generalization performance of the *GRR* round-robin repeated-labeling strategy with majority vote compared to that of single labeling *SL*, as a function of data acquisition cost. Figure 7a shows the case where the unlabeled part of a new

Fig. 7 Comparing the increase in accuracy for the *mushroom* data set as a function of data acquisition cost. *SL* is single labeling; *GRR* is generalized round-robin repeated-labeling, acquiring one new training example at a time, and using majority voting. Other data sets show similar results



(a) $p = 0.6$, $\rho = 0$ (i.e., $C_U = 0$), $k = 10$, for *GRR*

(b) $p = 0.6$, $\rho = 3$, $k = 5$, for *GRR*

(c) $p = 0.6$, $\rho = 10$, $k = 12$, for *GRR*

example is still free ($\rho = 0$, i.e., $C_U = 0$) for both *GRR* and *SL*. Here we just have in the horizontal axis the data acquisition cost instead of the number of labels. In this example, *GRR* gets ten labels per example. For the same cost, *SL* gets ten examples. In this higher noise scenario, it is much better to get examples with ten labels, as this reduces significantly the noise. Another issue is the size of the pool of available examples. For the *mushroom* data set, we have eight thousands examples; so, *SL* runs out of examples to label and this is the reason for the early termination shown in the *SL* curve. This case shows that *GRR* can be better than single labeling even when the labels are not particularly cheap.

Figure 7b, c illustrates scenarios where there is a cost for obtaining the unlabeled part of the example, with $\rho = 3, k = 5$ and $\rho = 10, k = 12$, respectively. We can see that *GRR* outperforms *SL* substantially, even though *GRR* allocates a significant amount of resources to re-labeling the examples. The results are similar across other data sets, in high-noise settings.

The high-level conclusion: as the cost ratio ρ increases, the improvement of *GRR* over *SL* also increases. So when the labeler quality is low, and the labels actually are cheap, we can actually get substantial advantage from using a repeated labeling strategy, such as *GRR*.

5.3 Different label integration methods

In all experiments shown above, both for *FRR* and *GRR*, we use majority voting for creating the integrated label from multiple labels. Obviously, some information (such as label uncertainty) is lost during this process. Soft labeling (described above) does not lose this information, and therefore may improve performance. In this section, we examine these two label integration methods (majority voting and soft labeling) by applying them to *GRR*. *GRR* with majority voting is called *MV* for short, and *GRR* with soft labeling is called *ME* for short. For *ME*, we generate *multiple examples* to preserve the uncertainty of the label multiset as described in Sect. 3.3.

Figure 8 plots the generalization accuracy of the models as a function of data acquisition cost. Here $\rho = 3$, $k = 5$, and we see very clearly that, for $p = 0.6$, both versions of repeated-labeling are preferable to single labeling. *MV* and *ME* outperform *SL* consistently (on all but waveform, where *MV* ties with *SL*) and, interestingly, the comparative performance of repeated-labeling tends to increase as one spends more on labeling. Furthermore, from the results in Fig. 8, we can see that the uncertainty-preserving repeated-labeling *ME* outperforms *MV* in all cases, to greater or lesser degrees.

In other results (not shown) we see that when labeling quality is substantially higher (e.g., $p = 0.8$), repeated-labeling still is increasingly preferable to single labeling as ρ increases; however, we no longer see an advantage for *ME* over *MV*. These results suggest that when labeler quality is low, inductive modeling often can benefit from the explicit representation of the uncertainty incorporated in the multiset of labels for each example. When labeler quality is relatively higher, this additional information apparently is superfluous, and straight majority voting is sufficient.

6 Selective repeated-labeling strategies

So far, we have considered repeated labeling as a uniform process across all examples. Now, we examine (i) whether selective allocation of labeling resources can further improve performance, and (ii) if so, how should the examples be selected. For

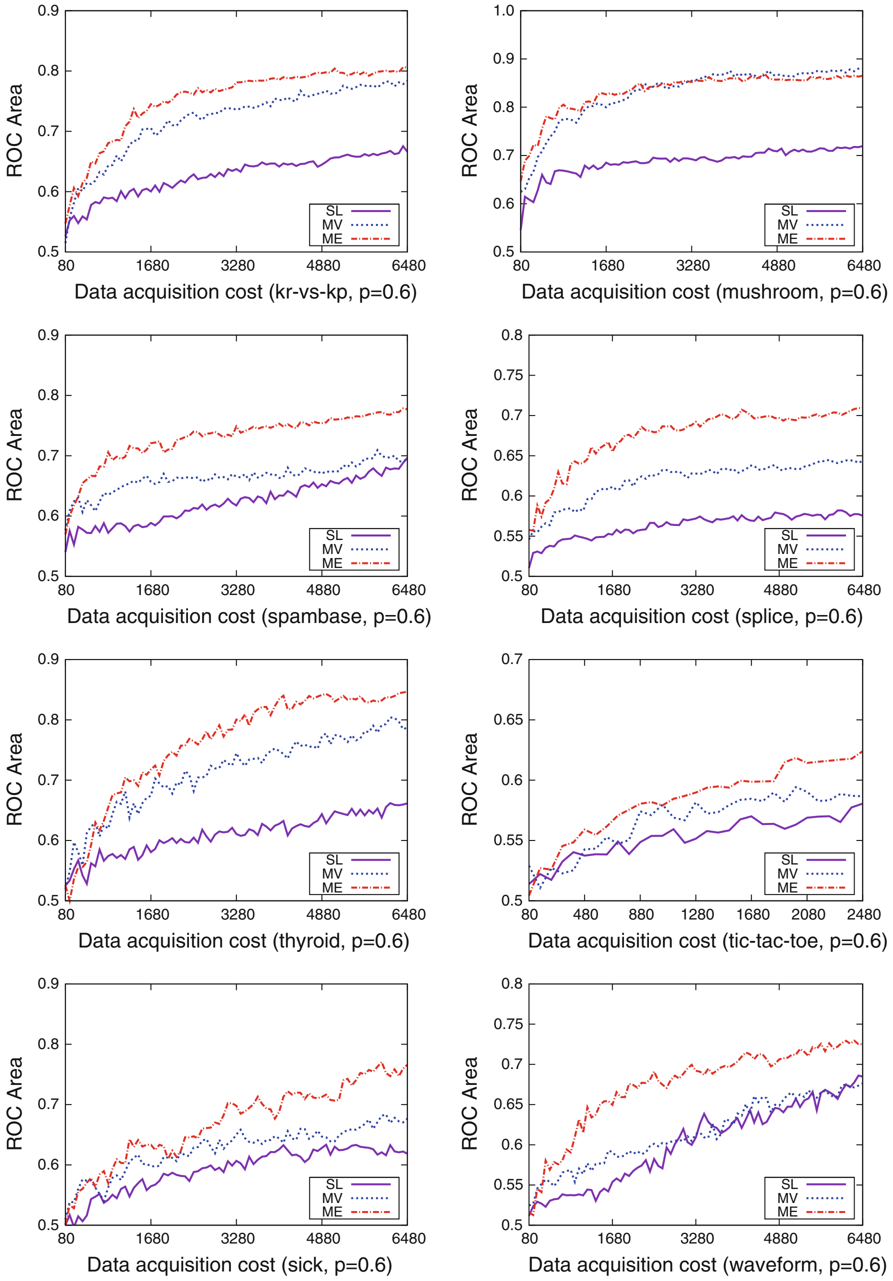
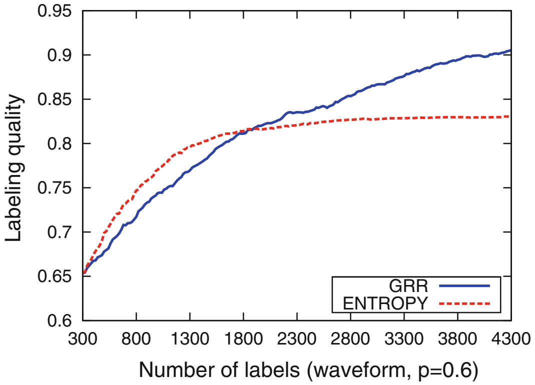


Fig. 8 Increase in model accuracy as a function of data acquisition cost for the eight data sets; ($p = 0.6$, $\rho = 3$, $k = 5$). *SL* is single labeling; *MV* is repeated-labeling with majority voting, and *ME* is uncertainty-preserving repeated-labeling. Both *MV* and *ME* are based on *GRR*

Fig. 9 What not to do: data quality improvement for an entropy-based selective repeated-labeling strategy versus round-robin repeated-labeling



example, intuitively it would seem better to augment the label multiset $\{+, -, +\}$ than to augment $\{+, +, +, +, +\}$.

6.1 What not to do

The example above suggests a straightforward procedure for selective repeated-labeling: acquire additional labels for those examples where the current multiset of labels is impure. Two natural measures of purity are:

- the entropy of the multiset of labels, and
- how close the frequency of the majority label is to the decision threshold (here, 0.5).

For our binary classification setting, these two measures generate the same example ranking. Unfortunately, there is a clear problem: under noise these measures do not really measure the uncertainty in the estimation of the class label. For example, $\{+, +, +\}$ is perfectly pure, but the true class is not certain (e.g., with $p = 0.6$ one is not 95 % confident of the true label). Applying a small-sample shrinkage correction (e.g., Laplace) to the probabilities is not sufficient.

Figure 9 (dashed line) demonstrates how labeling quality increases as a function of assigned labels, using the (Laplace-corrected) entropy-based estimation of uncertainty (*ENTROPY*). For small amounts of repeated-labeling, the *ENTROPY* technique does indeed select useful examples to label, but the fact that the estimates are not true estimates of uncertainty hurts the procedure in the long run—generalized round-robin repeated-labeling (GRR) from Sect. 5 outperforms the entropy-based approach. This happens because with ENTROPY most of the labeling resources are wasted, with the procedure labeling a small set of examples very many times. Note that with a high level of noise, the long-run label mixture will be quite impure, even though the true class of the example may be quite certain (e.g., consider the case of 600 positive labels and 400 negative labels with $p = 0.6$). Less impure, but incorrect, label multisets are never revisited.

6.2 Estimating label uncertainty (LU) using example-specific labeler quality

Instead, of relying on entropy measurements, we introduce a different approach: For a given multiset of labels, we compute a Bayesian estimate of the *labeling quality uncertainty* (LU) for each example. LU computes the uncertainty in true label of the integrated label under the following assumptions (some of which will be relaxed in the next section):

- All labelers have the same quality, *when labeling a given example*.[5]
- The labeling quality is always above 0.5 (i.e., we do not have adversarial labelers).
- We presume the prior distribution over the true label (quality) $p(y)$ to be uniform in the [0.5, 1] interval, but given that we do not know the true label y, the prior distribution over $p(y)$ becomes effectively uniform in the [0, 1] interval.

Labeling based on LU focuses the labeling efforts on examples for which we are uncertain about the quality of labeling, which given the assumptions, proxies for uncertainty in the true label. Specifically, we would like to estimate the uncertainty that the $p(y)$ of the example is on the "correct side" of the labeling decision threshold (and, therefore, that the majority of the votes also correspond to the correct label).

Consider a Bayesian estimation of the probability that y_m is incorrect. Here we do not assume that we know (or have estimated well) the labeler quality. Thus, after observing p positive labels and n negative labels, the posterior probability $p(y)$ follows a Beta distribution $B(p+1, n+1)$ (Gelman et al. 2003). LU computes the level of uncertainty as the tail probability below the labeling decision threshold. Formally, the uncertainty is equal to the CDF at the decision threshold of the Beta distribution, which is given by the regularized incomplete beta function:

$$I_x(\alpha,\beta) = \sum_{j=a}^{\alpha+\beta-1} \frac{(\alpha+\beta-1)!}{j!(\alpha+\beta-1-j)!} x^j (1-x)^{\alpha+\beta-1-j} \tag{4}$$

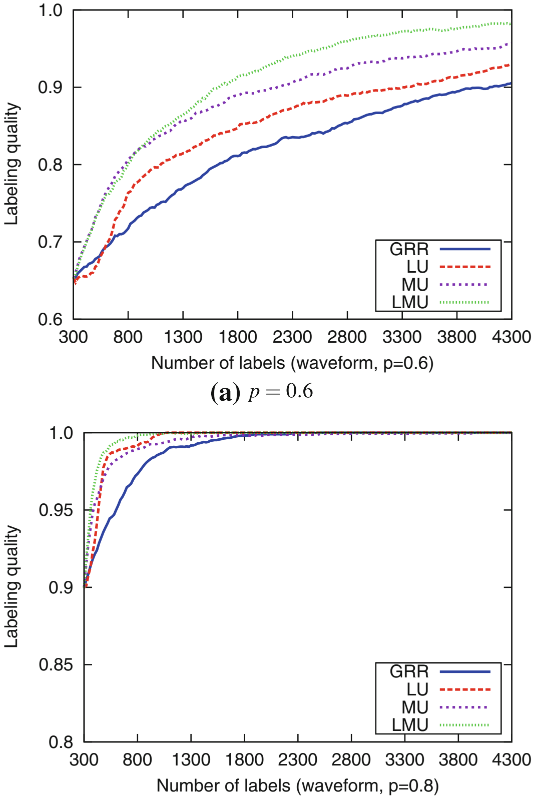
In our case, the decision threshold is $x = 0.5$, and $\alpha = p+1, \beta = n+1$. Thus, we set:

$$S_{LU} = \min\{I_{0.5}(p+1, n+1), 1 - I_{0.5}(p+1, n+1)\} \tag{5}$$

We compare selective repeated-labeling based on S_{LU} to round-robin repeated-labeling (GRR), which we showed to perform well in Sect. 5. To compare repeated-labeling strategies, we followed the experimental procedure of Sect. 5, with the following modification. Since we are asking whether LU can help with the selection of examples for which to obtain additional labels, each training example starts with three initial labels (selected as above). Then, each repeated-labeling strategy iteratively selects examples for which it acquires additional labels (since we need odd number of labels to use majority voting, we add two labels at a time in these experiments).

[5] We do *not* assume that the quality is the same across all examples. In fact, *LU* indirectly relies on the assumption that the labeling quality is different across examples.

Fig. 10 The data quality improvement of the four strategies (*GRR*, *LU*, *MU*, and *LMU*) for the *waveform* data set



(a) $p = 0.6$

(b) $p = 0.8$

Comparing selective repeated-labeling using S_{LU}[6] to *GRR*, we observed similar patterns across all twelve data sets; therefore we only show the results for the *waveform* data set (Fig. 10; ignore the *MU* and *LMU* lines for now, we discuss these techniques below), which are representative. The results indicate that *LU* performs better than *GRR*, identifying the examples for which repeated-labeling is more likely to improve quality.

6.3 Estimating example-specific label uncertainty directly

The assumptions behind the *LU* method are approximately satisfied in many situation, but are violated dramatically in one common setting: in the data, at least one class appears very infrequently. In this case, it may be the case that even intelligent, rational labelers are technically adversarial—i.e., their labeling quality for an infrequently appearing class is below 0.5. For example, they may be performing a Bayesian estimation of the class label and the prior on the true class is simply too small.

[6] As a shorthand we will simply call that Label Uncertainty (LU).

Instead of relying on the assumption of non-adversarial labelers, we can estimate the uncertainty of the integrated label directly, taking advantage of the estimation of the labeler quality distribution just computed (Sect. 6.2). Specifically, suppose that we have an example that has been labeled $l_p + l_n$ times, receiving l_p positive labels and l_n negative ones. If we had available the quality of the labelers p (assuming for simplicity a common p; labeler-and-example-specific $p_{i,j}$ would entail the natural expansion), it is straightforward to compute the probability $Pr(y_i|l_p, l_n)$, i.e., infer the true label given the l_p positive labels and l_n negative labels. Specifically, we have:

$$\begin{aligned} Pr(+|l_p, l_n) &= \frac{Pr(l_p, l_n|+) \cdot Pr(+)}{Pr(l_p, l_n)} = p^{l_p} \cdot (1-p)^{l_n} \frac{Pr(+)}{Pr(l_p, l_n)} \\ Pr(-|l_p, l_n) &= \frac{Pr(l_p, l_n|-) \cdot Pr(-)}{Pr(l_p, l_n)} = p^{l_n} \cdot (1-p)^{l_p} \frac{Pr(-)}{Pr(l_p, l_n)} \end{aligned} \tag{6}$$

In reality, the labeler quality p is unknown. However, given a sequence of l_p positive labels and l_n negative labels, we can apply Bayesian estimation to compute the distribution of possible values for p. After seeing l_p positive and l_n negative labels, the quality p, *for that example* follows a beta distribution. The distribution is $B(l_p + 1, l_n + 1)$ if $l_p > l_n$, or follows a beta distribution $B(l_n + 1, l_p + 1)$ if $l_p < l_n$. In other words:

$$Pr(q) = \begin{cases} \frac{\Gamma(l_p + l_n + 2)}{\Gamma(l_p + 1)\Gamma(l_n + 1)} \cdot q^{l_p} \cdot (1-q)^{l_n} : l_p \geq l_n \\ \frac{\Gamma(l_p + l_n + 2)}{\Gamma(l_p + 1)\Gamma(l_n + 1)} \cdot q^{l_n} \cdot (1-q)^{l_p} : l_p < l_n \end{cases} \tag{7}$$

Assuming $l_p \geq l_n$, and without loss of generality, we use the derivation of $Pr(+|l_p, l_n)$ from above, and integrate over all possible values of p:

$$\begin{aligned} Pr(+|l_p, l_n) &= \frac{Pr(l_p, l_n|+) \cdot Pr(+)}{Pr(l_p, l_n)} \\ &= \int_0^1 q^{l_p} \cdot (1-q)^{l_n} \frac{Pr(+)}{Pr(l_p, l_n)} Pr(q) dq \\ &= \frac{Pr(+)}{Pr(l_p, l_n)} \int_0^{1.0} q^{l_p} \cdot (1-q)^{l_n} \cdot \frac{\Gamma(l_p + l_n + 2)}{\Gamma(l_p + 1)\Gamma(l_n + 1)} \cdot q^{l_p} \cdot (1-q)^{l_n} dq \\ &= \frac{Pr(+)}{Pr(l_p, l_n)} \cdot \frac{\Gamma(l_p + l_n + 2)}{\Gamma(l_p + 1)\Gamma(l_n + 1)} \int_0^1 q^{2l_p} \cdot (1-q)^{2l_n} dq \\ &= \frac{Pr(+)}{Pr(l_p, l_n)} \cdot \frac{\Gamma(l_p + l_n + 2)}{\Gamma(l_p + 1)\Gamma(l_n + 1)} \frac{\Gamma(2l_n + 1)\Gamma(2l_p + 1)}{\Gamma(2 + 2l_n + 2l_p)} \end{aligned}$$

and similarly for $Pr(-|l_p, l_n)$. Using $Pr(+|l_p, l_n)$ and $Pr(-|l_p, l_n)$, and for $l_p \geq l_n$ we get:

$$Pr(+|l_p, l_n) = \left(1 + \frac{1 - Pr(+)}{Pr(+)} \cdot \frac{\left(\Gamma(l_n + l_p + 1)\right)^2}{\Gamma(2l_n + 1) \cdot \Gamma(2l_p + 1)}\right)^{-1} \tag{8}$$

Since l_p and l_n are integers, and $\Gamma(k) = (k-1)!$ for $k \in N$, we have:

$$Pr(+|l_p, l_n) = \left(1 + \frac{1 - Pr(+)}{Pr(+)} \cdot \frac{\left((l_n + l_p)!\right)^2}{(2l_n)! \cdot (2l_p)!}\right)^{-1} \tag{9}$$

where l_p is the number of positive labels, l_n is the number of negative labels (with $l_p \geq l_n$), $Pr(+)$ is the prior probability for the positive class, and $\Gamma(\cdot)$ is the Gamma function. (For $l_p < l_n$, it is symmetric.) So, our New Label Uncertainty (***NLU***) metric should be now $S_{NLU} = \min\{Pr(+|l_p, l_n), 1 - Pr(+|l_p, l_n)\}$.

From Eq. 9, we can see that we need to know the prior probability $Pr(+)$ to calculate the posterior probability $Pr(+|l_p, l_n)$, which is typically unknown. However, we can use a marginal maximum likelihood algorithm, to estimate $Pr(+)$:

1. Pick a random prior for positive class (e.g., $Pr(+) = 0.5$)
2. Compute the conditional probabilities $Pr(x_i = +|l_p, l_n)$ for each example x_i, using the current prior estimate
3. Compute the $Pr(+)$ as the average value of $Pr(x_i = +|l_p, l_n)$
4. Go to step 2 (stopping when some criterion is reached)

Experiments show that *NLU* improves *LU*, in terms of labeling quality and the model performance (accuracy), on the data sets where the class distribution is unbalanced, such as *sick and thyroid* (Fig. 11). For the data sets (*kr-vs-kp, mushroom, spambase, splice, tic-tac-toe, and waveform*), *NLU* has a similar performance to *LU* (Fig. 12). The results indicate that *LU* works well for balanced data sets, as suggested by the discussion above—for balanced data sets the assumptions behind the calculation of LU are likely to be satisfied. However, for unbalanced data sets, the non-adversarial assumption behind LU is likely to be violated. In this case, the direct uncertainty estimation of *NLU*—which does not make the non-adversarial assumption, and instead takes into consideration the prior class distribution during the uncertainty estimation—should improve the results. The experiments demonstrate that it indeed does so.

We can also compare the performance of *LU* and *NLU* based on their ability to identify and separate the correctly and incorrectly labeled examples. Ideally, all incorrectly labeled examples should have higher uncertainty scores than the correctly labeled ones.

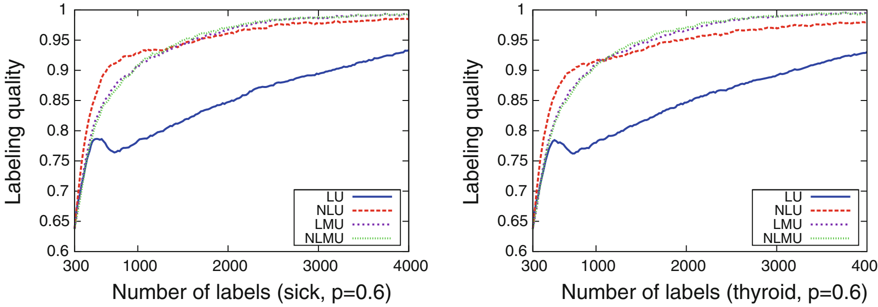


Fig. 11 Comparing the strategies (*NLU* and *NLMU*) with their previous version (*LU* and *LMU*) in terms of the improvement of integrated labeling quality on imbalanced data sets (*sick* and *thyroid*)

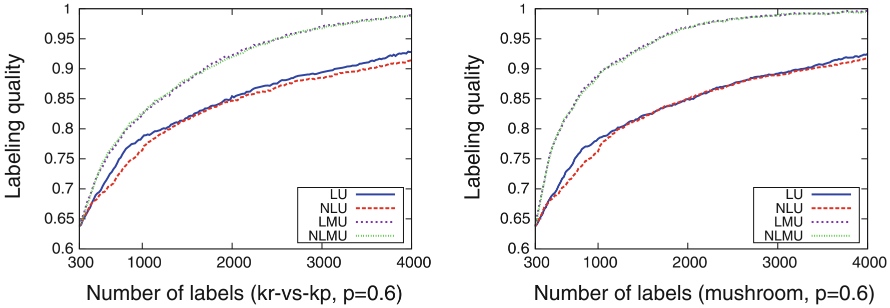


Fig. 12 Comparing the strategies (*NLU* and *NLMU*) with their previous version (*LU* and *LMU*) in terms of the improvement of integrated labeling quality on balanced data sets (*kr-vs-kp* and *mushroom*)

The method that better separates the two classes will be a better choice. In order to examine the differences of *LU* and *NLU*, we employ the Mann-Whitney-Wilcoxon (MWW) test, a non-parametric test for assessing whether two independent samples of observations can be separated. Specifically, we conduct the following experiment:

1. Pick a random prior for the positive class (e.g., $Pr(+) = 0.7$)
2. Generate n positive and negative examples (in our case, $n = 300$), following the prior value from the previous step.
3. Assign three initial labels to each example; the labeler quality is p.
4. Assign two labels to each of the 10 examples selected by the GRR strategy.[7]
5. Compute the uncertainty scores for *LU*, and *NLU*; also, we compute the "*Gold*" uncertainty score, which assumes knowledge of the quality of the labelers and uses Eq. 6 to compute the uncertainty for each example. Note that this is the best possible uncertainty metric that we can have when we only know about the multiset of the assigned labels.
6. Compute the uncertainty scores for correctly and incorrectly labeled examples, and compute the MWW statistic for the two sets of points (the correctly and incorrectly labeled examples). Higher values of MWW indicate better separating ability.
7. Go to step 4 until the total number of labels are used up.

Figure 13 gives us the results of the experiment. When the dataset is balanced (i.e., $(Pr(+) \approx 0.5)$, the performance of *LU*, *NLU*, and *Gold* are similar. This indicates that *LU* and *NLU*, are almost optimal. However, in imbalanced data sets, *NLU* outperforms *LU* by a wide margin. *NLU* has a performance much closer to *Gold*, especially when we have only a few labels per example.

These results lead to a striking conclusion: at least as far as these experiments can be generalized, *NLU* is getting most of the information that is possible out of the label multisets alone. If we want to improve selective repeated labeling substantially over *NLU*, we need to turn to a different source of information.

[7] We do *not* use selective labeling strategies for this experiment, as we want to keep the labeling allocation strategy constant, and independent of the two uncertainty scoring strategies. The goal is to see which uncertainty score can separate best the correctly from the incorrectly labeled examples.

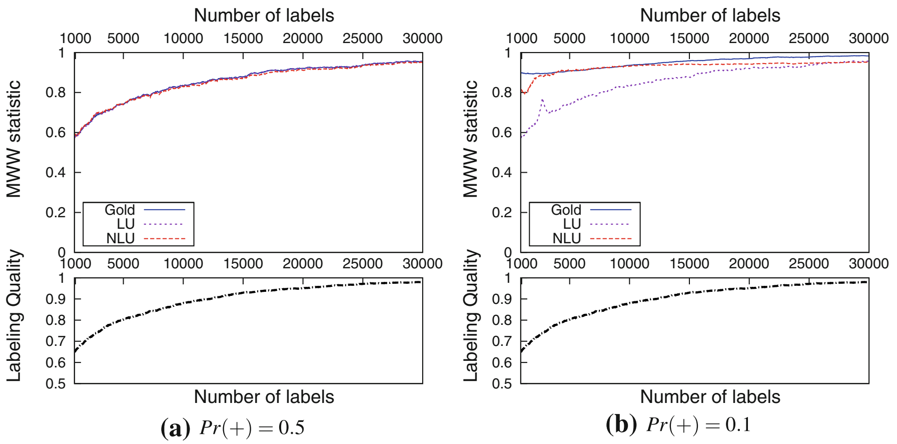


Fig. 13 MWW statistic for three uncertainty scores (*LU*, *NLU* and *Gold*) and the corresponding labeling quality ($p = 0.6$)

6.4 Using model uncertainty

The two techniques described above, *LU* and *NLU*, assume that the different examples that are being labeled are independent of each other. In other words, the labels assigned to one example do not give any information about the labels that should be assigned to a different example. However, in many settings we already may be assuming that similar examples will be labeled similarly—that's a key assumption behind our using supervised learning methods to build predictive models.

Let us now turn to a method for taking advantage of this assumption to improve selective repeated labeling. Consider the following idea. Assume that examples are labeled similarly, to some extent, to the examples "near" to them in example space. Machine learning provides many ways to define nearness. For the sake of concreteness in our discussion, let's assume that there are different regions of the example space (that can be found by a machine learning method), and within these regions examples have some probability of belonging to the class that is different from that of the overall population. For simplicity, let's assume that examples are generated at random from these regions (possibly based on a region-specific sampling rate), and that this comprises the complete data-generating process. This mimics the assumption, for example, that would justify many classification tree and rule-learning methods.

Now, what happens if one or more training examples is drawn from a particular region and mis-labeled? In this case, in expectation the resultant learned model will give a slightly different estimated probability of class membership for the examples in that region than if all the training data were labeled correctly. And in expectation, the probability is more likely than not to move away from certainty—i.e., toward the region's minority class. (Because the mis-labeled example is more likely than not to be of the region's majority class!) Thus, we may be able to find mislabeled examples for relabeling by looking for examples for which a learned *model* has more uncertainty in its labeling.

This idea seems very similar to, and in fact was inspired by, active learning. However, it works for a very different reason (which we will demonstrate below). Unlike traditional active learning, which builds models from the labeled examples and uses these models to predict the uncertainty of each *unlabeled* example, our *MU* strategy applies a strategy similar to that of Brodley and Friedl (1999), but different in an important way which we will describe presently. It builds models on the noisy data and uses the models to predict the uncertainty of each example *within these same noisy data*. (Contrast this with active learning, where the *model* is applied to a different, unlabeled set of data.)

So, specifically, we learn a classification model using the existing labeled examples, and use the resulting classifier to get information about the model's uncertainty in the class for each of these same examples. One can envision different ways to instantiate this approach. We devise and experiment with a simple one here: producing a measure of "model uncertaintly" for an example. An example becomes a candidate for (re-)labeling when a learned model's uncertainty about its class is high.

More specificaly, the measure Model Uncertainty (MU) ignores the current multiset of labels. It learns a model to estimate the probability of class membership for each example. The *MU* score is computed as:

$$S_{MU} = 0.5 - |Pr(+|x, H) - 0.5| \tag{10}$$

where $Pr(+|x, H)$ is the probability of classifying the example x into $+$ by the learned model H.

As we will show more formally next, *MU* is well justified under the simplifying assumptions we have discussed. Whether it works in real-world settings is an empirical question.

Proposition 1 *Assume that data are generated by region of the example space, as described above, and given a classifier that estimates the probability of binary class membership $\hat{Pr}(+|x \in r)$ for any region r of the example space better than random guessing. Ceteris paribus, in expectation, the training examples in a region of the space with a mislabeled training example are more likely to have a higher S_{MU} than those in a region with no mislabeled example. Therefore, in expectation, a mislabeled training example is likely to be selected for relabeling by MU before the examples in all regions with no mislabeled training examples.*

Proof sketch Consider N regions of the example space R, $r_1, r_2, \ldots, r_N \in R$, which satisfy $Pr(+|x \in r_1) = Pr(+|x \in r_2) = \cdots = Pr(+|x \in r_N)$. Assume for this proof sketch that the data generating process selects training examples uniformly at random from the regions and then they are labeled correctly by labelers with the exception of one example t. Without loss of generality, assume $+$ is the majority class in the region (the alternate case is symmetric) and therefore since the classifier is better than random guessing, in expectation $\hat{Pr}(+|x \in r_i) > 0.5$ for each r_i.

Now, consider training data T in which one example t in r_1 is mislabeled, and the rest are not. There are two cases, depending on whether or not $t \in +$.

Case (i): If the true label of t is positive, in expectation the value of $|Pr(+|x \in r_1) -0.5|$ decreases; the training data from the region have a smaller proportion of positive examples, but in expectation still more than half.[8] Therefore, $S_{MU}(x_i \in r_1) > S_{MU}(x_j \in r_{k \neq 1})$.

Case (ii): If the true label of t is negative, in expectation the value of $|Pr(+|x \in r_1) - 0.5|$ *increases*. In this case, $S_{MU}(x_i \in r_1) < S_{MU}(x_j \in r_{k \neq 1})$.

However, since the majority class is positive, in expectation, case (i) will occur more frequently than case (ii). Therefore, it is more likely than not that $S_{MU}(x_i \in r_1) > S_{MU}(x_j \in r_{k \neq 1})$. Therefore, the mislabeled example will be likely to be relabeled by *MU* before any $x' \in r_{k \neq 1}$. □

In our experiments, model H is a *random forest* (Breiman 2001) of ten models (generated by Weka, averaging the class membership probabilities). Notice that our *MU* is quite different from the ensemble methods used in Verbaeten and Assche (2003). In their paper, the binary choice of whether or not to eliminate an example is based specifically on a consensus (or majority vote) filter built from a committee of classifiers. However, our *MU* method works for any class-probability estimator which can give uncertainty scores. We use a *random forest* here simply because that it would in general give us reasonably good probability estimates.

We compare *MU* to other strategies below, but first let us address its obvious drawback. By ignoring the label set, *MU* has the complementary problem to *LU*: even if the model is uncertain about a case, should we acquire more labels if the existing label multiset is very certain about the example's class? The investment in these labels would be wasted, since they would have a small effect on either the integrated labels or the learning.

A hybrid strategy, called Label and Model Uncertainty (LMU), combines the uncertainty scores from *LU* and *MU*, to *avoid examples where either model is certain*. This is done by computing the score S_{LMU} as the geometric average[9] of S_{LU} and S_{MU}. That is:

$$S_{LMU} = \sqrt{S_{MU} \cdot S_{LU}} \tag{11}$$

Similarly, we combined the score of *NLU* with the score of *MU*, to create the New Label and Model Uncertainty (NLMU) method, for which:

$$S_{NLMU} = \sqrt{S_{MU} \cdot S_{NLU}} \tag{12}$$

Figure 10 demonstrates the improvement in data quality when using model information, which is typical of the results across our data sets. We can observe that the *LMU* model strongly dominates all other strategies. In high-noise settings ($p = 0.6$) *MU* also performs well compared to *GRR* and *LU*, indicating that when noise is high,

[8] Since the Proposition and proof sketch are mainly to give theoretical motivation to MU, let's assume that the induction algorithm is no worse than a standard classification tree learner.

[9] Subsequent to these experiments, we also experimented with other approaches for combining probabilities from multiple sources, following the discussion in Clemen and Winkler (1990). For our experiments, taking the geometric mean was the best performing and most robust approach for combining the uncertainty scores, even after transforming the uncertainty scores into proper probability estimates.

using learned models helps to focus the investment in improving quality. In settings with low noise ($p = 0.8$), *LMU* continues to dominate, but *MU* no longer outperforms *LU* and *GRR*.

6.4.1 *Why model uncertainty* MU *works*

Following the discussion above, we expect that incorrectly labeled examples will tend to have higher *MU* uncertainty scores, compared to correctly labeled ones. Figure 14 shows a representative result: We computed the uncertainty scores of correctly and incorrectly labeled examples for the *bmg* dataset ($m = 10$, $p = 0.6$). Figure 14 illustrates that the uncertainty scores of the correctly labeled examples are centered around 0, while the distribution of scores for incorrectly labeled examples has a much higher mean.

To assess the relative contribution of this self-healing property of *MU*, we compare our *MU* with versions corresponding to traditional active learning. Specifically, the latter uses 10-fold cross-validation; every fold is treated as active learning's unlabeled set, and the remaining 9 as the labeled set (training set) for building models. [We refer to this strategy as *MUCV*; it is essentially the same as the strategy of Brodley and Friedl (1999).] *MUCV* follows a similar procedure as *MU*. However, unlike *MUCV* that builds a random forest using the training data, our *MU* strategy builds a random forest from the whole data without separating the data into training and testing sets. This comparison isolates the contribution of applying the model back to the training data (for self-healing), as opposed to active learning's attempt to concentrate on parts of the space near the classification boundary, or that otherwise are not (yet) modeled well. As a point of further comparison, we also build a strategy called Model Uncertainty with Oracle (*MUO*): we use all the original data with perfect labels (provided by a perfect Oracle, but unknown to the re-labeling procedure except through the resultant model) and build a random forest; we use this random forest to predict the uncertainty of each example in the noisy data. For all strategies, the uncertainty score is defined as in Eq. 10.

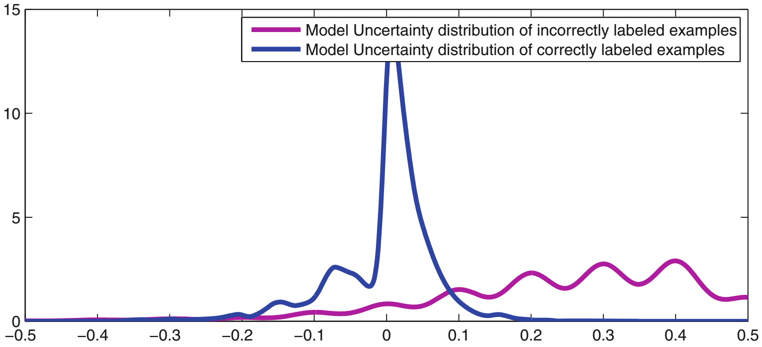


Fig. 14 Why MU works: model uncertainty distribution of correctly and incorrectly labeled examples for the *bmg* data set

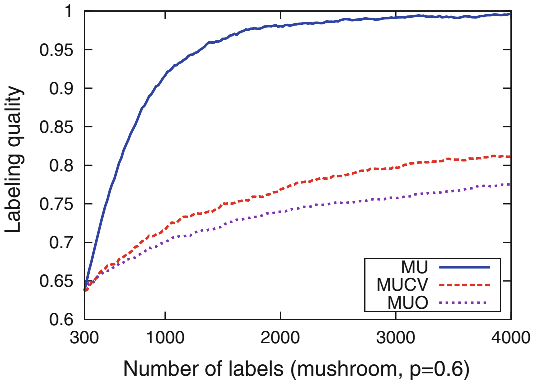


Fig. 15 The data quality improvement on three versions of model uncertainty (*MU*, *MUCV*, and *MUO*) for the *mushroom* data set

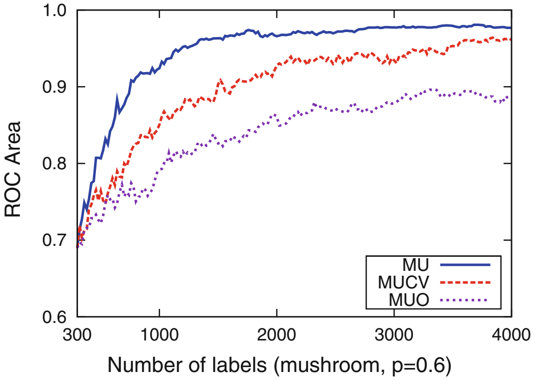


Fig. 16 The model quality improvement on three versions of model uncertainty (*MU*, *MUCV*, and *MUO*) for the *mushroom* data set

Figures 15 and 16 show representative experimental results for the three versions of model uncertainty, using the *mushroom* data set. Figures 15 and 16, show that *MUCV* and *MUO* are not nearly as effective as *MU* for repeated labeling. *MU* dynamically finds examples that seem to be causing problems with the modeling, and improves their label quality by acquiring more labels. In contrast, *MUO* ignores the characteristics of the noisy data completely and applies the statistical models learned from the original noise-free training data, so *MUO* always acquires more labels for the same small set of examples—producing static models unaffected by the additional labels. *MUCV* is also a dynamic process. However, the benefit of identifying examples that cannot be classified easily (as with active learning) is not nearly as large as the benefit of self-healing with *MU*.

6.5 Classification performance with selective repeated-labeling

So, finally, let us assess whether selective repeated-labeling accelerates learning (i.e., improves model generalization performance, in addition to data quality). Again, experiments are conducted as described above, except here we compute generalization

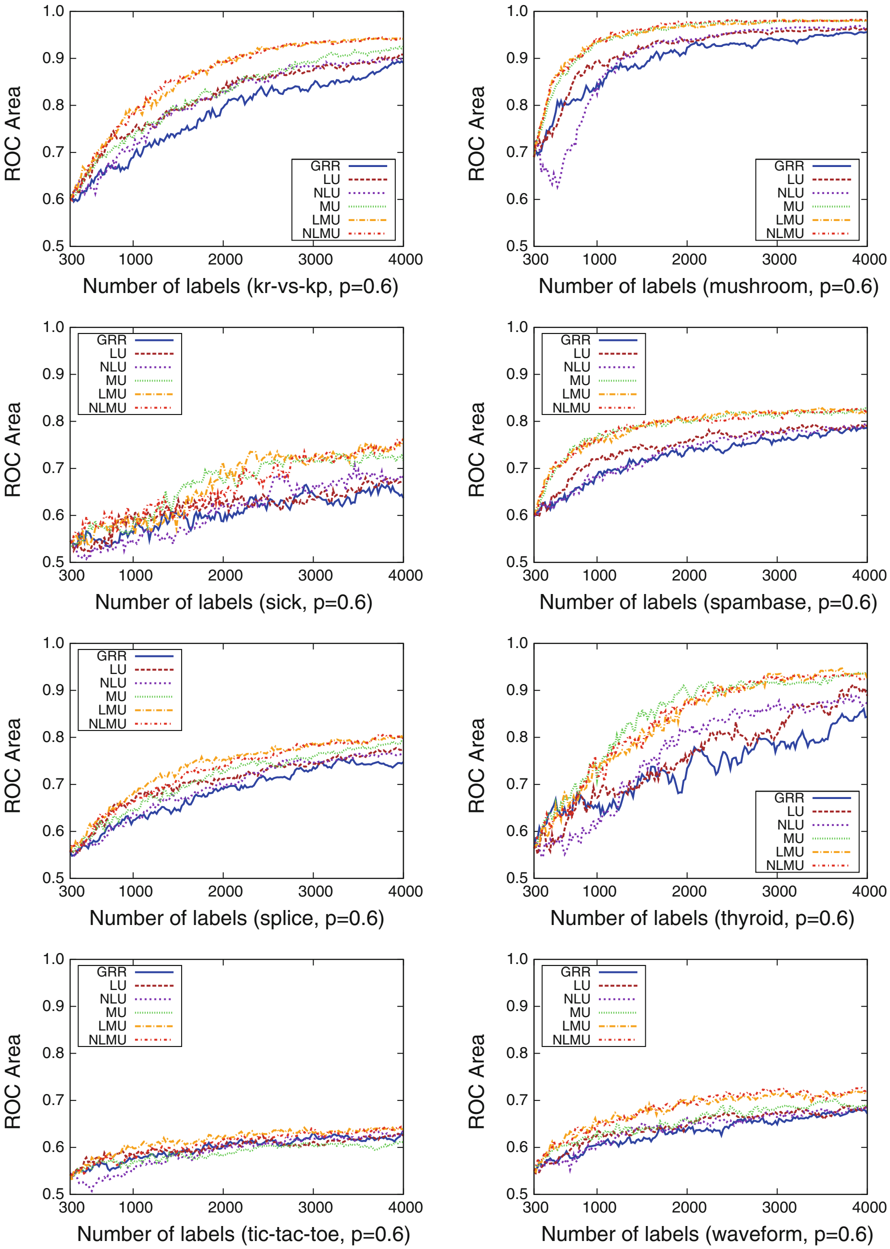


Fig. 17 Accuracy as a function of the number of labels acquired for the six selective repeated-labeling strategies for the eight data sets ($p = 0.6$)

accuracy averaged over the held-out test sets (as described in Sect. 4.1). In Fig. 17, we show the performances of these four strategies in terms of classification accuracy on the eight data sets.

Table 2 Average AUC of the six strategies over eight data sets, for $p = 0.6$

Data Set	GRR	LU	NLU	MU	LMU	NLMU
kr-vs-kp	*0.891*	0.908	0.902	0.925	0.941	**0.942**
Mushroom	*0.956*	0.964	0.966	**0.980**	**0.980**	**0.980**
Sick	*0.637*	0.674	0.685	0.717	**0.764**	0.761
Spambase	*0.786*	0.789	0.796	0.825	0.818	**0.825**
Splice	*0.747*	0.776	0.766	0.790	**0.800**	0.797
Thyroid	*0.844*	0.889	0.875	**0.938**	0.926	0.924
Tic-tac-toe	0.625	0.627	0.637	*0.611*	0.637	**0.644**
Waveform	*0.672*	0.689	0.686	0.689	0.716	**0.723**
Average	*0.770*	0.789	0.789	0.809	0.823	**0.825**

For each data set, the best performance is in boldface and the worst in italics

We report values for $p = 0.6$, a high-noise setting that can occur in real-life training data.[10] Table 2 summarizes the results of the experiments, reporting accuracies (AUCs) averaged across the acquisition iterations for each data set, with the maximum AUC across all the strategies highlighted in bold, the minimum AUC italicized, and the grand averages reported at the bottom of the columns.

Here are the main findings from the results. The two basic methods that use LU (*LU* and *NLU*) are consistently better than round-robin repeated-labeling, achieving higher accuracy for every data set. (Recall that in the previous section, round-robin repeated-labeling was shown to be substantially better than the baseline single labeling in this setting.) The performance of model uncertainty alone (*MU*) is more variable: in one case it has the best accuracy, but in another case it does not even reach the accuracy of round-robin repeated-labeling. Overall, combining label and model uncertainty (*LMU* and *NLMU*) produce the best approaches: in these experiments, they always outperform round-robin repeated-labeling, and as hypothesized, generally they are better than the strategies based on only one type of uncertainty (in the case of *LU* and *NLU*, the corresponding combined strategy is better in every case; statistically significant by a sign test at $p < 0.01$).

6.6 Alternative selective strategies

In previous sections, we have studied the performance of the selective repeated labeling strategies. The experimental results show that all selective strategies perform better than single labeling. In this section, we examine some alternative strategies, that can be used in conjunction with the techniques that we presented so far. Specifically, we discuss the case of using "soft labels" and the case of using weighted sampling to select

[10] From Provost and Danyluk (1995): *"No two experts, of the five experts surveyed, agreed upon diagnoses more than 65 % of the time. This might be evidence for the differences that exist between sites, as the experts surveyed had gained their expertise at different locations. If not, however, it raises questions about the correctness of the expert data."*

the example to label, instead of picking the examples in absolute order according to their uncertainty scores.

6.6.1 Soft-labeling

Majority voting is the most straightforward method for integrating multiple labels for each example. In contrast to majority voting, soft-labeling (refer to Sect. 3.3) retains the uncertainty in the multiset of labels. Specifically, the *ME* technique produces a fractionally weighted example for each unique label in the set. In Sect. 5.3, we saw that soft-labeling can improve the performance of the GRR strategy. Now, we investigate whether soft-labeling can improve selective re-labeling.

Curiously, in our experimental results, we did not observe consistent improvements in generalization performance by incorporating soft-labeling, as compared to the majority-voting counterparts. The results from the data sets *mushroom, splice, and thyroid* are representative and shown in Figs. 18, 19 and 20. For some data sets (e.g., *mushroom*), the performance of soft-labeling and majority voting are similar. For other data sets (e.g., *splice*), soft-labeling performs a little better, while on others (e.g., *thyroid*), soft-labeling reduces performance significantly.

Based on these results and those from above, we can conclude that soft-labeling is a strategy to consider in environments with high noise and when using basic round-robin labeling strategies. When selective labeling is employed, the benefits of using soft-labeling apparently diminish, and so far we do not have the evidence to recommend using soft-labeling.

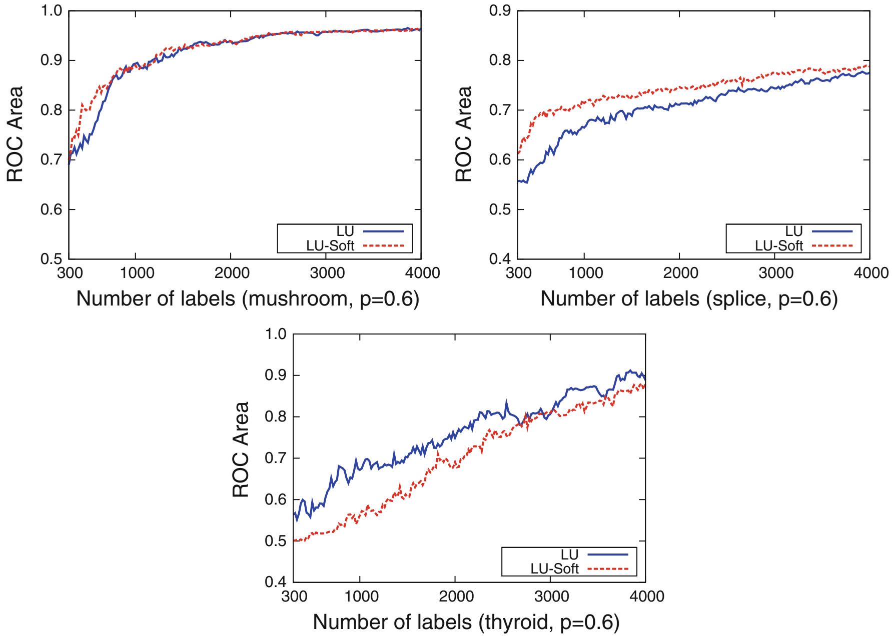


Fig. 18 The accuracy improvement of soft-labeling on *LU* on the *mushroom, splice, thyroid* data sets

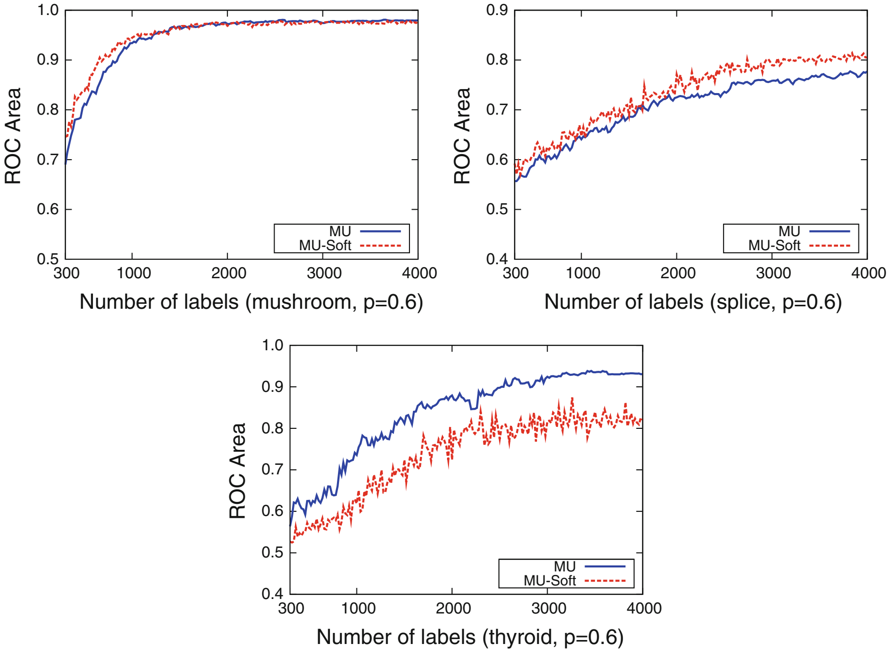


Fig. 19 The accuracy improvement of soft-labeling on *MU* on the *mushroom, splice, thyroid* data sets

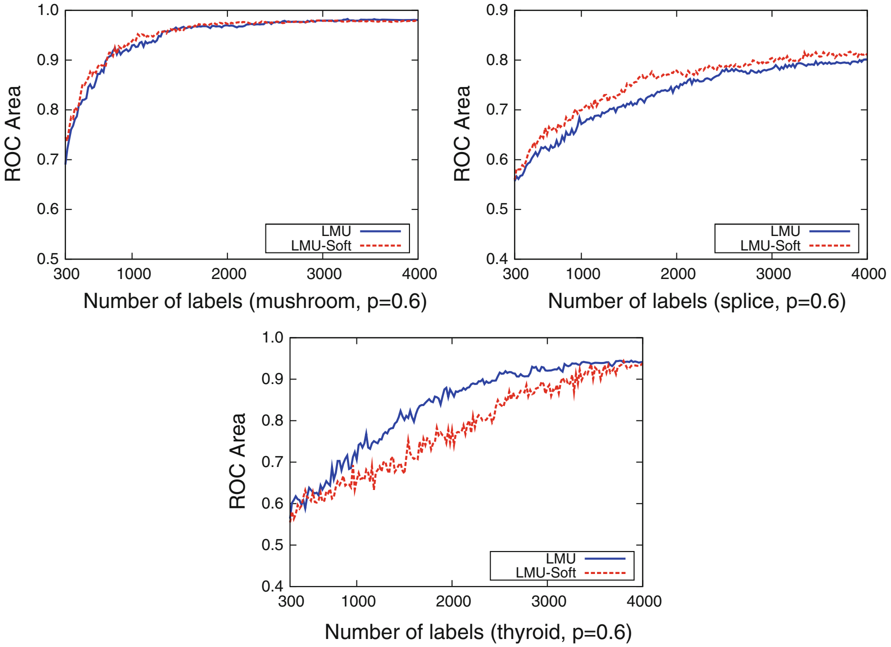


Fig. 20 The accuracy improvement of soft-labeling on *LMU* on the *mushroom, splice, thyroid* data sets

6.6.2 Weighted sampling

Weighted sampling has been shown previously to be a useful tool for improving the performance of active learning (Saar-Tsechansky and Provost 2004). In this section, we further study whether weighted sampling can also improve the performance of the selective repeated-labeling strategies.

So far, all selective repeated-labeling strategies always acquired new labels for the most uncertain examples, in absolute priority. Now we study the technique proposed in (Saar-Tsechansky and Provost 2004) where an example is selected probabilistically, with a probability proportional to its uncertainty: if the uncertainty score of an example is s_i (where s_i is computed following S_{MU}, S_{LU}, S_{LMU}, S_{NLU}, S_{NLMU}), then the probability of picking that example for labeling is $\frac{s_i}{\sum_j s_j}$. Figure 21 shows the *labeling quality* of the selective repeated-labeling strategies with and without weighted sampling for the *mushroom* data set. Figure 22 shows the change of the *accuracy* of the selective repeated-labeling strategies with and without weighted sampling for the *mushroom* data set. (We only show the experimental results of the *mushroom* data set, but the results are representative across all data sets.)

We can see that (this version of) weighted sampling does not improve the performance (labeling quality and accuracy) of the selective repeated-labeling strategies. The three selective repeated-labeling strategies with deterministic selection order perform significantly better than the ones with weighted sampling. The weighted sampling

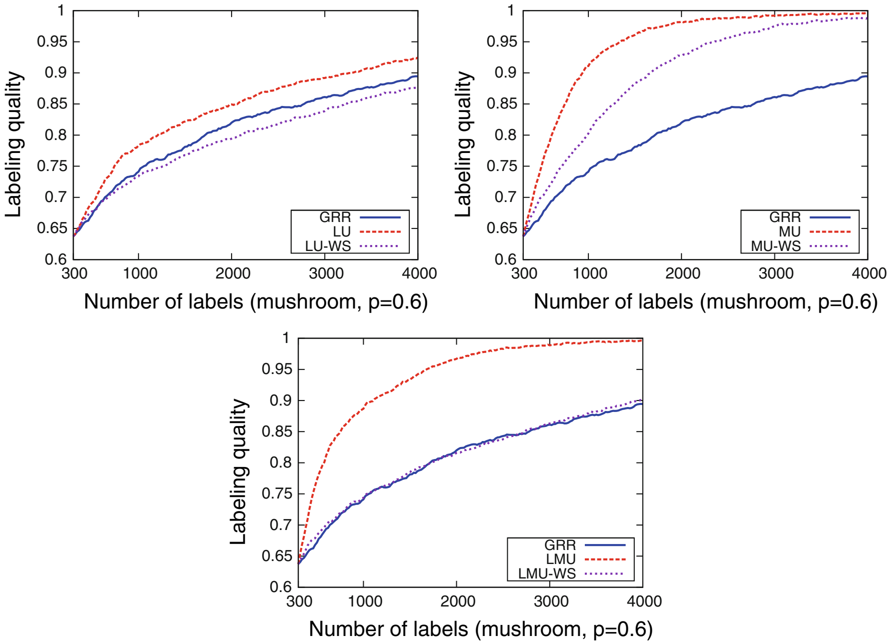


Fig. 21 The labeling quality of the three selective repeated-labeling strategies with/without weighted sampling for the *mushroom* data set

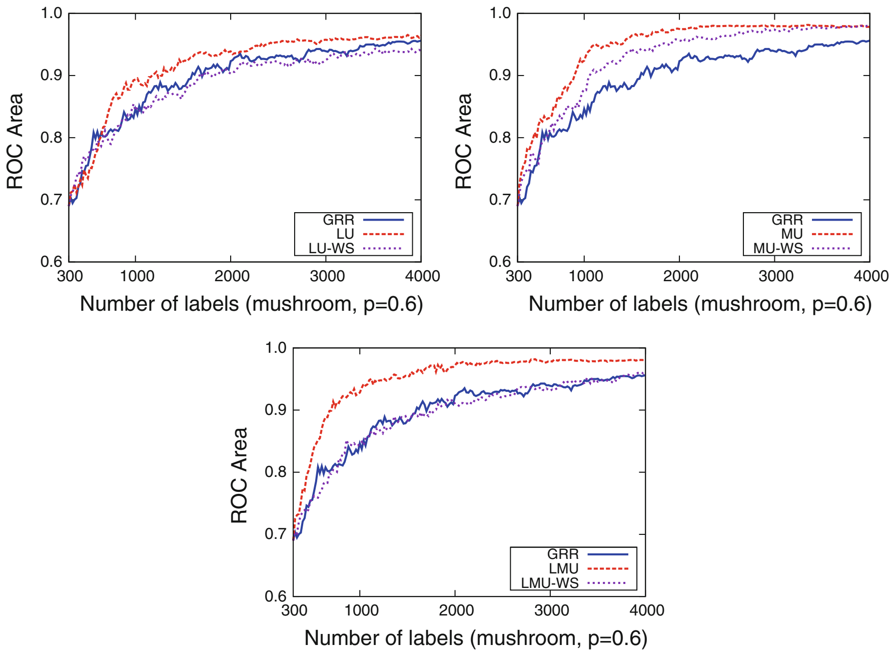


Fig. 22 Accuracy of the three selective repeated-labeling strategies with/without weighted sampling for the *mushroom* data set

makes the three repeated-labeling strategies worse. Intuitively, this happens because weighted sampling allocates resources to examples that are only marginally uncertain. Due to the large number of examples with low uncertainty, the weighted sampling strategies end up allocating significant amount of labeling resources to re-label examples that are perfectly good and do not need any further labels. Note that the present setting is quite different from that of active learning: here we have direct, albeit noisy, information on the class of the example. Of course, there are alternative ways to use the uncertainty scores to form a sampling distribution. It may be that a version that has a sharper peak at the uncertain end of the spectrum would be effective; such tinkering is left to future work.

7 Evaluation on real-world data

So far, we have presented experiments on real, benchmark data sets but with simulated labelers; the main reason was the need to examine the performance of our algorithms under a variety of settings. In this section, we present an additional experimental evaluation using both real data *and* real labelers, who may also exhibit characteristics that are not explicitly accounted for in our algorithms (e.g., different levels of noise, correlated errors, some examples that are inherently harder than others, etc.).

To get a better understanding of how our repeated-labeling strategies perform in practical settings, we use a real-world data set labeled by Amazon Mechanical Turk

Table 3 MTurk-spam data set: The number of attributes, examples, and the split into positive and negative examples

Data set	#Examples	Pos	Neg
MTurk-spam	4,111	1,600	2,511

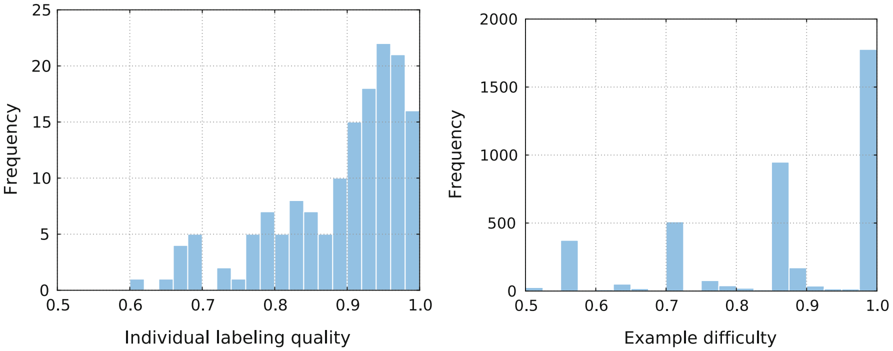


Fig. 23 Histogram of individual labeling quality and example difficulty for the "MTurk-spam" data set

workers. This data set includes a total of 4,111 descriptions of tasks posted on the Amazon Mechanical Turk marketplace. (In other words, the workers on Mechanical Turk, as part of our experiment, had to examine other tasks that were posted on the market.) We asked the workers to examine whether task is asking workers to perform an action that is designed to game social media metrics (e.g., "follow me on Twitter," "like my video on YouTube," etc.). We collected a total of 32,752 labels. Table 3 summarizes characteristics of the data set.

We ran the experiment using CrowdFlower:[11] CrowdFlower is a commercial system that helps requesters to find trust-worthy workers. CrowdFlower achieves the goal by checking the labels that workers assign to examples with labels already known to the employer (often referred to as "gold tests"). By checking how often workers assign the correct labels, CrowdFlower is able to eliminate workers with extremely low quality. The disadvantage of the gold-based filtering is the need to generate sufficient amounts of gold data for the filtering: with only a small number of gold tests, the tests tend to repeat and spammers can estimate which examples serve the role of gold tests. The labeler quality in the CrowdFlower-filtered data set is high, with an average labeler quality being $p \approx 0.85$. Figure 23 shows the distributions of labeler quality and example difficulty. Individual labeler quality is measured as the fraction of examples that are correctly classified by a particular labeler (where correctness is defined by the majority of high-quality CrowdFlower workers), and example difficulty is measured as the fraction of correct labels for a particular example.

While on CrowdFlower we can get high worker quality (by eliminating workers that fail gold tests), in many crowdsourcing environments, including Mechanical Turk,

[11] http://crowdflower.com

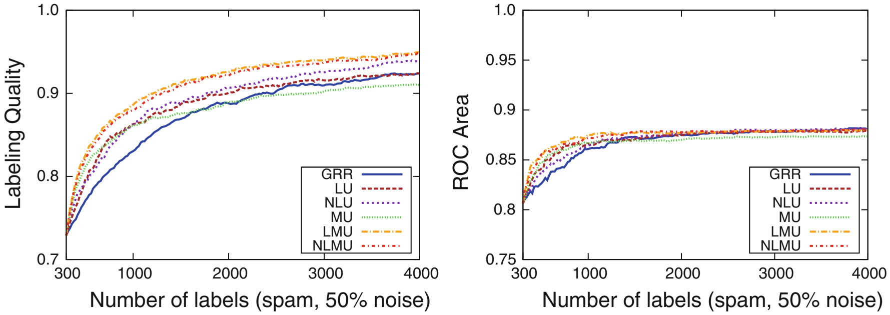


Fig. 24 The labeling quality and accuracy as a function of the number of labels acquired for the six selective repeated-labeling strategies for the "MTurk-spam" data set

we often have a very significant amount of noise. To experiment with various levels of noise, we used our data set and randomly add a fraction of noisy labels to the data set, and also add "spammer" workers that contribute only noise. In particular, whenever we need a new label for an example, we do the following (suppose the noise level is α):

1. Draw a random number between 0 and 1: If the number is larger than α, go to step 2; otherwise, go to step 3.
2. Randomly draw with replacement, a label among the ones assigned to the objects by the workers.
3. Artificially add a label to the example with an accuracy of 0.5 (to simulate the existence of spammers).

Figure 24 shows the labeling quality and accuracy for the six repeated-labeling strategies, with the use of linear SVMs as the learning algorithm and with noise level $\alpha = 0.5$. In this setting, half of the workers contributing just noise, while the other half have an average quality of $p \approx 0.85$, as described above; this results in a setting with highly heterogeneous worker qualities. Our selective repeated-labeling strategies *LMU* and *NLMU* outperform *GRR* for both the integrated labeling quality and the overall classification accuracy. The results are qualitatively similar for different levels of α as well. The behavior of our algorithms on our real data is very similar to the behavior of our algorithms on the benchmark data sets with simulated labelers, which adds some evidence to the superiority of the proposed selective repeated-labeling strategies. It is encouraging to see the selective and repeated labeling algorithms working better than existing baselines, even when the assumptions of independence and of equal difficulty of the examples do not hold. Future work could elaborate further.

8 Conclusions, limitations, and future work

Repeated-labeling is a tool that should be considered whenever labeling might be noisy, but can be repeated. We showed that under a wide range of conditions, it can improve both the quality of the labeled data directly, and the quality of the models learned from the data. In particular, *selective* repeated-labeling seems to be preferable, taking into account both labeling uncertainty and model uncertainty.

Our focus in this paper has been on improving data quality for supervised learning; however, the results have implications for data mining generally. We showed that selective repeated-labeling improves the data quality directly and substantially. This could be helpful for many data mining applications.

This paper makes important assumptions that should be visited in future work, in order for us to understand practical repeated-labeling and realize its full benefits.

- The techniques we present can be applied no matter if the labelers have the same or different qualities. Furthermore, the NLU and NLMU selective labeling algorithm explicitly account for the quality of the labelers being different across different data points. We have not experimented extensively with the effects of labelers of varying qualities. Moreover, good estimates of individual labelers' qualities inferred by observing the assigned labels (Dawid and Skene 1979; Ipeirotis et al. 2010; Smyth 1996; Smyth et al. 1994b) could allow more sophisticated selective repeated-labeling strategies.
- The methods and experiments in this paper consider binary classification only. However, many classification problems have multiple classes. Model uncertainty applies directly to the multiclass setting as well. LU can be extended directly to the multiclass setting by replacing the Beta distribution with the Dirichlet distribution (the multivariate generalization of the Beta distribution).
- It would be interesting to see if labelers exhibit higher quality in exchange for a higher payment. Some recent work (Mason and Watts 2009) indicates that this may not be the case. It would be interesting to observe empirically how individual labeler quality varies as we vary C_U and C_L, and to build models that dynamically increase or decrease the amounts paid to the labelers, depending on the quality requirements of the task. Morrison and Cohen (2005) determine the optimal amount to pay for noisy information in a decision-making context, where the amount paid affects the level of noise.
- We also assumed that C_L and C_U are fixed and indivisible. Clearly there are domains where C_L and C_U would differ for different examples, and could even be broken down into different acquisition costs for different features. Thus, repeated-labeling may have to be considered in tandem with costly feature-value acquisition. Indeed, feature-value acquisition may be noisy as well, so one could envision a generalized repeated-labeling problem that includes both costly, noisy feature acquisition and label acquisition.
- In this paper, we consider the labeling process to be a noisy process over a *single, true label*. An alternative, practically relevant setting is where the label assignment to a case is inherently uncertain. (For example, assesments on whether a racy celebrity gossip website should be classified into an "adult-only" category or into "parental-guidance"). This is a separate setting where repeated-labeling could provide benefits, but we leave it for future analysis.
- In our repeated-labeling strategy we compared repeated-labeling vs. single labeling, and did not consider any hybrid scheme that can combine the two strategies. A promising direction for future research is to build a *"learning curve gradient"-based* approach that decides dynamically which action will give the highest marginal accuracy benefit for the cost. Such an algorithm would compare on-the-fly the

expected benefit of acquiring new examples versus selectively repeated-labeling existing, noisy examples and/or features.

Despite these limitations, we hope that this study provides a solid foundation on which future work can build. Furthermore, we believe that both the analyses and the techniques introduced can have immediate, beneficial practical application.

Acknowledgments This work was supported by the National Science Foundation under Grant No. IIS-0643846 and IIS-1115417, by an NSERC Postdoctoral Fellowship, by an NEC Faculty Fellowship, by a Google Focused Award, and a George Kellner Fellowship. Thanks to Carla Brodley, John Langford, and Sanjoy Dasgupta for enlightening discussions and comments.

References

Baram Y, El-Yaniv R, Luz K (2004) Online choice of active learning algorithms. J Mach Learn Res 5:255–291

Blake CL, Merz CJ (1998) UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. Accessed 11 Mar 2013

Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recognit 37(9):1757–1771

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167

Carpenter B (2008) Multilevel bayesian model of categorical data annotation. http://lingpipe-blog.com/lingpipe-white-papers/. Accessed 11 Mar 2013

Clemen RT, Winkler RL (1990) Unanimity and compromise among probability forecasters. Manag Sci 36(7):767–779

Cohn DA, Atlas LE, Ladner RE (1994) Improving generalization with active learning. Mach Learn 15(2):201–221

Dawid AP, Skene AM (1979) Maximum likelihood estimation of observer error-rates using the EM algorithm. Appl Stat 28(1):20–28

Domingos P (1999) MetaCost: a general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining (KDD-99). pp 155–164

Donmez P, Carbonell JG (2008) Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: Proceedings of the 17th ACM conference on information and knowledge management (CIKM 2008). pp 619–628

Donmez P, Carbonell JG, Schneider J (2009) Efficiently learning the accuracy of labeling sources for selective sampling. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2009). pp 259–268

Donmez P, Carbonell JG, Schneider J (2010) A probabilistic framework to learn from multiple annotators with time-varying accuracy. In: Proceedings of the SIAM international conference on data mining (SDM 2010). pp 826–837

Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the seventeenth international joint conference on, artificial intelligence (IJCAI-01). pp 973–978

Gelman A, Carlin JB, Stern HS, Rubin DB (2003) Bayesian data analysis, 2nd edn. Chapman and Hall/CRC, Boca Raton

Ipeirotis PG, Provost F, Wang J (2010) Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD workshop on human computation (HCOMP 2010). pp 64–67

Jin R, Ghahramani Z (2002) Learning with multiple labels. In: Advances in neural information processing systems 15 (NIPS 2002). pp 897–904

Kapoor A, Greiner R (2005) Learning and classifying under hard budgets. In: ECML 2005, 16th European conference on machine learning. pp 170–181

Lizotte DJ, Madani O, Greiner R (2003) Budgeted learning of naive-bayes classifiers. In: 19th conference on uncertainty in artificial intelligence (UAI 2003). pp 378–385

Lugosi G (1992) Learning with an unreliable teacher. Pattern Recognit 25(1):79–87

Margineantu DD (2005) Active cost-sensitive learning. In: Proceedings of the nineteenth international joint conference on, artificial intelligence (IJCAI-05). pp 1622–1613

Mason W, Watts DJ (2009) Financial incentives and the performance of crowds. In: Proceedings of the human computation workshop (HCOMP 2009). pp 77–85

McCallum A (1999) Multi-label text classification with a mixture model trained by EM. In: AAAI'99 workshop on text learning

Melville P, Saar-Tsechansky M, Provost FJ, Mooney RJ (2004) Active feature-value acquisition for classifier induction. In: Proceedings of the 4th IEEE international conference on data mining (ICDM 2004). pp 483–486

Melville P, Provost FJ, Mooney RJ (2005) An expected utility approach to active feature-value acquisition. In: Proceedings of the 5th IEEE international conference on data mining (ICDM 2005). pp 745–748

Morrison CT, Cohen PR (2005) Noisy information value in utility-based decision making. In: Proceedings of the 1st international workshop on utility-based data mining (UBDM'05). pp 34–38

Provost F (2005) Toward economic machine learning and utility-based data mining. In: Proceedings of the 1st international workshop on utility-based data mining (UBDM'05). p 1

Provost F, Danyluk AP (1995) Learning from bad data. In: Proceedings of the ML-95 workshop on applying machine learning, in practice. pp 27–33

Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106

Quinlan JR (1992) C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc, San Mateo

Raykar VC, Yu S, Zhao LH, Jerebko A, Florin C, Valadez GH, Bogoni L, Moy L (2009) Supervised learning from multiple experts: whom to trust when everyone lies a bit. In: Proceedings of the 26th annual international conference on machine learning (ICML 2009). pp. 889–896

Raykar VC, Yu S, Zhao LH, Valadez GH, Florin C, Bogoni L, Moy L (2010) Learning from crowds. J Mach Learn Res 11(7):1297–1322

Rebbapragada U, Brodley CE (2007) Class noise mitigation through instance weighting. In: 18th European conference on machine learning (ECML'07). pp. 708–715

Saar-Tsechansky M, Provost F (2004) Active sampling for class probability estimation and ranking. Mach Learn 54(2):153–178

Saar-Tsechansky M, Melville P, Provost F (2009) Active feature-value acquisition. Manag Sci 55(4):664–684

Sheng VS, Provost F, Ipeirotis P (2008) Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the fourteenth ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2008). pp. 614–622

Silverman BW (1980) Some asymptotic properties of the probabilistic teacher. IEEE Trans Inf Theory 26(2):246–249

Smyth P (1995) Learning with probabilistic supervision. In: Petsche T (ed) Computational learning theory and natural learning systems, vol. III: selecting good models. MIT Press, Cambridge

Smyth P (1996) Bounds on the mean classification error rate of multiple experts. Pattern Recognit Lett 17(12):1253–1257

Smyth P, Burl MC, Fayyad UM, Perona P (1994a) Knowledge discovery in large image databases: Dealing with uncertainties in ground truth. In: Knowledge discovery in databases: papers from the 1994 AAAI, workshop (KDD-94). pp 109–120

Smyth P, Fayyad UM, Burl MC, Perona P, Baldi P (1994b) Inferring ground truth from subjective labelling of Venus images. In: Advances in neural information processing systems 7 (NIPS 1994). pp 1085–1092

Snow R, O'Connor B, Jurafsky D, Ng AY (2008) Cheap and fast–but is it good? Evaluating non-expert annotations for natural language tasks. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP'08). pp 254–263

Ting KM (2002) An instance-weighting method to induce cost-sensitive trees. IEEE Trans Knowl Data Eng 14(3):659–665

Turney PD (1995) Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. J Artif Intell Res 2:369–409

Turney PD (2000) Types of cost in inductive concept learning. In: Proceedings of the ICML-2000 workshop on cost-sensitive, learning. pp 15–21

Verbaeten S, Assche AV (2003) Ensemble methods for noise elimination in classification problems. In: Fourth international workshop on multiple classifier systems. Springer, pp 317–325 von Ahn L, Dabbish L (2004) Labeling images with a computer game. In: Proceedings of the 2004 conference on human factors in computing systems (CHI 2004). pp 319–326

Weiss GM, Provost FJ (2003) Learning when training data are costly: the effect of class distribution on tree induction. J Artif Intell Res 19:315–354

Whitehill J, Ruvolo P, fan Wu T, Bergsma J, Movellan J (2009) Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Advances in neural information processing systems 22 (NIPS 2009). pp 2035–2043

Whittle P (1973) Some general points in the theory of optimal experimental design. J R Stat Soc Ser B 35(1):123–130

Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann Publishing, San Francisco

Zadrozny B, Langford J, Abe N (2003) Cost-sensitive learning by cost-proportionate example weighting. In: Proceedings of the 3th IEEE international conference on data mining (ICDM 2003). pp 435–442

Zheng Z, Padmanabhan B (2006) Selectively acquiring customer information: a new data acquisition problem and an active learning-based solution. Manag Sci 52(5):697–712

Zhu X, Wu X (2005) Cost-constrained data acquisition for intelligent data preparation. IEEE Trans Knowl Data Eng 17(11):1542–1556