Finding Rare Classes: Active Learning with Generative and Discriminative Models

Timothy M. Hospedales, Shaogang Gong, and Tao Xiang

Abstract—Discovering rare categories and classifying new instances of them is an important data mining issue in many fields, but fully supervised learning of a rare class classifier is prohibitively costly in labeling effort. There has therefore been increasing interest both in active discovery: to identify new classes quickly, and active learning: to train classifiers with minimal supervision. These goals occur together in practice and are intrinsically related because examples of each class are required to train a classifier. Nevertheless, very few studies have tried to optimise them together, meaning that data mining for rare classes in new domains makes inefficient use of human supervision. Developing active learning algorithms to optimise both rare class discovery and classification simultaneously is challenging because discovery and classification have conflicting requirements in query criteria. In this paper we address these issues with two contributions: a unified active learning model to jointly discover new categories and learn to classify them by adapting query criteria online; and a classifier combination algorithm that switches generative and discriminative classifiers as learning progresses. Extensive evaluation on a batch of standard UCI and vision datasets demonstrates the superiority of this approach over existing methods.

Index Terms—active learning, rare class discovery, imbalanced learning, classification, generative models, discriminative models

1 Introduction

Many real life problems are characterized by data distributed between vast yet uninteresting background classes, and small rare classes of interesting instances which should be detected. In astronomy, the vast majority of sky survey image content is due to well understood phenomena, and only 0.001% of data is of interest for astronomers to study [1]. In financial transaction monitoring, most are perfectly ordinary but a few unusual ones indicate fraud and regulators would like to find future instances [2]. Computer network intrusion detection exhibits vast amounts of normal user traffic, and a very few examples of malicious attacks [3]. In computer vision based security surveillance of public spaces, observed activities are almost always everyday behaviours, but very rarely there may be a dangerous or malicious activity of interest [4], [5]. All of these classification problems share two interesting properties: highly unbalanced proportions – the vast majority of data occurs in one or more background classes, while the instances of interest for detection are much rarer; and unbalanced prior knowledge – the majority classes are typically known a priori, while the rare classes are not. In order to discover and learn to classify the interesting rare classes, exhaustive labeling of a large dataset would be required to ensure coverage and sufficient representation of all rare classes. However this is often prohibitively expensive as generating each label may require significant time from a human expert.

Active learning strategies can help to discover rare classes [1] or train a classifier [6], [7] with lower label cost. However, training a classifier for a priori undiscovered classes requires both discovery and classifier learning. This is challenging due to the dependence of classifier learning on discovery (training a classifier requires some examples of each class) and the conflict between good discovery and classifier learning criteria (an unlabeled point whose label is likely to reveal a new class is unlikely to improve an existing classifier and vice-versa). The problem of joint discovery and classification via active learning has received little attention despite its importance and broad relevance. The only existing attempt to address this is based on simply applying schemes for discovery and classifier learning in fixed iteration [3]. Sequential or iterative [3] methods effectively treat discovery and classification independently, in that the selection of criteria at each step does not depend on their relative success. They may therefore make inefficient use of queries and perform poorly. For example, spending active learning queries to perfect a particular classifier is useless if the interesting classes are not yet discovered; and spending queries searching for new classes is a poor use of resources if all classes have been discovered.

We address joint discovery and classification by adaptively balancing multiple criteria based on their success both at discovery and improving classification. Specifically, we propose to build a generative-discriminative model pair [8] because as we shall see, generative models naturally provide good discovery criteria and discriminative models naturally provide good classifier learning criteria. As a second contribution, we note that depending on the actual supervision cost and sparsity of rare class examples, the availability of labels will vary across datasets and classes. Given the nature of data

• *The authors are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, UK. Email: {tmh,sgg,txiang}@eecs.qmul.ac.uk*

dependence in generative and discriminative models [8] (in which generative models are often better with very little data; and discriminative models are often better asymptotically) the better classifier will vary across both the dataset and the stage of learning. We address this uncertainty by proposing a classifier switching algorithm to ensure the best classifier is selected for a given dataset and availability of labels. Evaluation on a batch of vision and UCI datasets covering various domains and complexities, shows that our approach consistently and often significantly outperforms existing methods at the important task of simultaneous discovery and classification of rare classes.

2 Related Work

A common approach to rare class detection that avoids supervised learning is outlier detection [4], [9], [10]: building an unconditional model of the data and flagging unlikely instances under this model. Outlier detection has a few serious limitations however: i) It cannot detect instances of non-separable categories, where the interesting classes are embedded in the majority distribution; ii) It does not subsequently exploit any supervision about the true class of flagged outliers, limiting its accuracy – especially in distinguishing rare classes from noise; iii) It is intrinsically binary, treating all data as either normal or outlying. Different rare classes, which may be of varying importance, can not be distinguished.

If it is possible to label some examples of each class, iterative active learning approaches are often used to learn a classifier with minimal supervision effort [6], [7]. Much of the active learning literature is concerned with the relative merits of different criteria for supervision requests. For example, querying points that: are most uncertain [11], [12]; reduce the version space [12], [13]; or reduce direct approximations of the generalization error [14], [15]. Different criteria may function best for different datasets [6], e.g, uncertainty based criteria is often good to refine an approximately known decision boundary, but may be poor if the classes are non-separable (the most uncertain points may be hopeless) or highly multi-modal. This has led to attempts to fuse [7], [16] or select dataset specific criteria online [17]. All these approaches rely on classifiers, and do not generally apply to scenarios in which the target classes have not been discovered yet.

Recently, active learning has been applied to *discovering* rare classes. That is, using selective supervision to quickly find an example of each class. Points may be queried based on e.g., likelihood [1], gradient [18], clustering [19] or nearest neighbor [20] criteria.

Most existing studies exploiting active-learning are single-objective: either discovery or classification, but not both. Using active learning to solve discovery and classifier learning together is challenging because even for a single dataset, good discovery and classification criteria are often completely different. Consider the toy

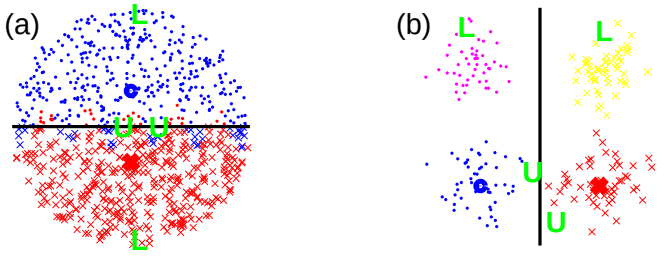


Figure 1. Problems with (a) a complex decision boundary and (b) multiple classes.

scenarios in Figure 1. Here the color indicates the true class, and the symbol indicates the estimated class based on two initial labeled points (large symbols). The black line indicates the initial decision boundary. In Figure 1(a) all classes are known but the decision boundary needs refining. Likelihood sampling (most unlikely point under the learned model) inefficiently builds a model of the whole space (choosing first the points labeled L), while uncertainty sampling selects points closest to the decision boundary (U symbols), leading to efficient refinement. In Figure 1(b) only two classes have been discovered. Uncertainty inefficiently queries around the known class decision boundary (choosing first the points U) without discovering the completely new (and currently incorrectly classified) classes above. In contrast, these are the first places queried by likelihood sampling (L symbols). Evidently, single-criterion approaches are insufficient. Moreover, multiple criteria may be desirable for a single dataset at different stages of learning, e.g., likelihood to detect new classes and uncertainty to learn to classify them. The only existing study addressing both discovery and classification is [3], which non-adaptively iterates over criteria in fixed proportions. However as we will see, such inflexible approaches risk performing poorly due to making inefficient use of the limited supervision.

Our innovation is to adaptively select criteria online, which can increase efficiency dramatically for learning to classify in the presence of undiscovered classes. Typically "exploration" will be preferred while there are easily discoverable classes, and "exploitation" to refine decision boundaries will be preferred when most classes have been discovered. We will however see that this is not the case for every dataset, and that our model can adapt to situations the ideal order is reversed, where one criterion is consistently best, or where it is useful to return to searching for the rarest classes after learning to classify easier ones. This ultimately results in better rare class detection performance than single objective, or non-adaptive methods [3].

Finally, there is the issue of what base classifier to use with active learning. One can categorize classifiers into two broad categories: generative and discriminative. Discriminative classifiers directly learn $p(y|x)$ for class y and data x. Generative classifiers learn $p(x|y)$

and $p(y)$ and then compute $p(y|x)$ via Bayes rule. The importance of this for active learning is that there is some empirical and theoretical evidence that for a given generative-discriminative *pair* (in the sense of equivalent parametric form, e.g., naive Bayes & logistic regression [8], [21] or Gaussian mixtures and support vector machines [22]), generative classifiers often perform better with very few training examples, while discriminative models are often better asymptotically. One intuitive reason why this can occur is that by imposing a stronger parametric constraint $p(x|y)p(y)$ on the data, generative models may overfit less with low data whereas more flexible discriminative models tend to overfit [22]. On the other hand, with ample data, generative model misspecification (e.g., an assumption of Gaussianity in $p(x|y)$ not quite met by the data) will penalize accuracy more compared to a more flexible discriminative model simply representing $p(y|x)$ [21]. The ideal classifier is therefore likely to change at some unknown point during active learning. An automatic way to select the right classifier at test time is therefore crucial. This is especially so in the rare class context where some classes may never obtain more than a few examples. Existing active learning work tends to focus on generative [11], [14] or discriminative [12], [17] classifiers. We develop an algorithm to switch classifiers online in order to get the best of both worlds.

3 Active Discovery and Learning

3.1 Active Learning

Standard learning problems assume an instance space of data $\mathcal{X}$ and labels $\mathcal{Y}$, with joint statistics $p(X, Y)$. The aim is to learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ with low generalization error:

$$E(f) \;=\; \int \sum_{Y} L(f(X), Y) p(X, Y) dX, \qquad (1)$$

where L is a loss function penalizing disagreement between $f(X)$ and Y. In pool based active learning [6], [7], [15], we are given a large set of unlabeled instances $\mathcal{U} = (\mathbf{x}_1, .., \mathbf{x}_m)$ and a small set of labeled instances $\mathcal{L} = ((\mathbf{x}_1, y_1), .., (\mathbf{x}_n, y_n))$. Active learning proceeds by iteratively: i) training a classifier f on $\mathcal{L}$, and ii) using query function $\mathcal{Q}(f, \mathcal{L}, \mathcal{U}) \rightarrow i^*$ to select an unlabeled instance i^* to be labeled, removing $\mathbf{x}_{i^*}$ from $\mathcal{U}$ and adding $(\mathbf{x}_{i^*}, y_{i^*})$ to $\mathcal{L}$. The goal of active learning is to choose instances i^* to label, so as to obtain a low error $E(f)$ classifier f with few iterations. Directly selecting the sequence of ideal i^*s to minimize (1) is usually intractable, so various approximations [6], [7], [14], [23] have been proposed.

The crucial difference between our problem and traditional active learning [6], [7] is that the initial labeled set $\mathcal{L}$ does not cover all possible labels $\mathcal{Y}$. This makes it unclear how to choose i^* to minimize (1) even approximately. We note, however, that the likelihood sampling criterion (4) has been shown effective at discovering new classes [1]; while uncertainty sampling (2) provides a simple greedy approximation [7], [23] to minimizing error (1) – but only if all classes are known in advance. Since success at discovery is necessary for error reduction, combining the criteria appropriately will be essential for good performance. We will bridge this gap by introducing an intuitive adaptive query strategy which balances likelihood and uncertainty criteria according to their success at their respective goals: discovering new classes, and reducing error for known classes.

3.1.1 Query Criteria

Uncertainty. The intuition behind uncertainty sampling is that if the class of a point is highly uncertain, obtaining a label should improve discrimination between the two classes. Uncertainty is typically quantified by posterior entropy, which for binary classification reduces to selecting the point whose posterior is closest to $p(y|\mathbf{x}) = 0.5$. The posterior $p(y|\mathbf{x})$ of every point in $\mathcal{U}$ is evaluated and the uncertain points queried,

$$i^* \;=\; \underset{i}{\operatorname{argmax}} \left(-\sum_{y_i} p(y_i|\mathbf{x}_i; \theta) \log p(y_i|\mathbf{x}_i; \theta) \right), \qquad (2)$$

$$p_u(i) \;\propto\; \exp\left(\beta \sum_{y_i} p(y_i|\mathbf{x}_i; \theta) \log p(y_i|\mathbf{x}_i; \theta) \right). \qquad (3)$$

Rather than selecting a single maxima (2), a normalized *degree of preference* for every point can be expressed by putting the entropy into a Gibbs function (3). For non-probabilistic SVM classifiers, $p(y|\mathbf{x})$ can be approximated based on the distance to the margin at each point [6].

Likelihood. A complementary query criteria is that of low likelihood $p(\mathbf{x}|y)$. Such points are badly explained by the current (generative) model, and may reflect an as yet unseen class [1]. This may involve marginalizing over the class or selecting the maximum likelihood label,

$$i^* \;=\; \underset{i}{\operatorname{argmin}} \left(\max_{y_i} p(\mathbf{x}_i|y_i; \theta) \right), \qquad (4)$$

$$p_l(i) \;\propto\; \exp\left(-\beta \max_{y_i} p(\mathbf{x}_i|y_i; \theta) \right). \qquad (5)$$

The uncertainty measure in (2) is in spirit *discriminative* (in focusing on decision boundaries), although $p(y|\mathbf{x})$ can obviously be realized by a generative classifier. In contrast, the likelihood measure in (4) is intrinsically *generative*, in that it requires a density model $p(\mathbf{x}|y)$ of each class y, rather than just the decision boundary. The uncertainty measure is usually poor at finding new classes, as it focuses on known decision boundaries, and the likelihood measure is usually good at finding new classes, while being poorer at refining decision boundaries between known classes (Figure 1). Neither of these are always the case, however. For example, a risk of the uncertainty measure is that it can perform poorly if parts of the data are highly non-separable – it will

indefinitely query impossible to separate areas of space; meanwhile, the likelihood measure could still improve known-class classification if the classes are multi-modal – it will explore different modes. Our adaptation method in Section 3.3 will allow the criteria to flexibly applied according to which is more likely to reduce error (1) at the current stage of learning on a particular dataset.

3.1.2 Density Weighting

There is one potential concern with how uncertainty, and especially likelihood sampling relate to the unconditional density of the data $p(\mathbf{x})$. It may be that the most uncertain or unlikely points are in low-density regions of the input space, so learning a good model there is not a good use of supervision since few test points will be drawn there and the generalization error (1) will not be significantly improved. At worst, uncertain or unlikely points may simply be actual noise rather than interesting rare classes. If this is the case, a good solution is to additionally weight the criteria by the unconditional density of the data $p(\mathbf{x})$ which we will readily obtain in the process of building a generative model of the data (see Section 3.2). We can then define a density-weighted variant of uncertainty or likelihood sampling as:

$$p_{wu}(i) \propto \exp\left(\beta \sum_{y_i} p(y_i|\mathbf{x}_i;\theta)\log p(y_i|\mathbf{x}_i;\theta)p(\mathbf{x}_i)\right), \quad (6)$$

$$p_{wl}(i) \propto \exp\left(-\beta \max_{y_i} p(\mathbf{x}_i|y_i;\theta)p(\mathbf{x}_i)\right). \quad (7)$$

Various studies [16], [24], [25], [26] have advocated density weighting of query criteria to improve generalization by ensuring points are both informative and representative. Notably [25] suggests weighted uncertainty sampling early in the learning process, moving toward unweighted uncertainty sampling later in the learning process. The intuition is that it is worth refining the decision boundary in areas relevant to many points first, and only move onto learning about sparse areas once dense areas are reliably modeled. In our case we can leverage this idea by simply including the density weighted variant of each criteria another option to be adapted (see Section 3.3). Next we discuss specific procedures for learning the required generative model $p(\mathbf{x}|y)p(y)$ and discriminative model $p(y|\mathbf{x})$.

3.2 Generative-Discriminative Model Pairs

We use a Gaussian mixture model (GMM) for the generative model and a support vector machine (SVM) for the discriminative model. These were chosen because they may both be incrementally trained (for active learning efficiency), and they are a complementary generative-discriminative *pair* in that (assuming a radial basis SVM kernel) they have equivalent classes of decision boundaries [22], but are optimized with different criteria during learning. Given these models, we will initially use the GMM and SVM to compute the likelihood and uncertainty criteria respectively, (although we will see later that this is not always the best strategy).

3.2.1 Incremental GMM Estimation

For online GMM learning, we use the constant time incremental agglomerative algorithm from [10]. To summarize the procedure, for the first $n = 1..N$ training points observed with the same label y, $\{\mathbf{x}_n, y\}_n^N$, we incrementally build a model $p(\mathbf{x}|y)$ for y using kernel density estimation with Gaussian kernels $\mathcal{N}(\mathbf{x}_n, \Sigma)$ and weight $w_n = \frac{1}{n}$. d is the dimension of $\mathbf{x}$.

$$\begin{aligned} p(\mathbf{x}|y) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \\ &\quad \cdot \sum_{n=1}^{N} w_n \exp -\frac{1}{2}\left((\mathbf{x}-\mathbf{x}_n)^T\Sigma^{-1}(\mathbf{x}-\mathbf{x}_n)\right). \end{aligned} \quad (8)$$

After some maximal number of Gaussians N_{max} is reached, merge two existing Gaussians i and j by moment matching [27] as follows

$$\begin{aligned} w_{(i+j)} &= w_i + w_j, \\ \mu_{(i+j)} &= \frac{w_i}{w_{(i+j)}}\mu_i + \frac{w_j}{w_{(i+j)}}\mu_j, \\ \Sigma_{(i+j)} &= \frac{w_i}{w_{(i+j)}}\left(\Sigma_i + (\mu_i - \mu_{(i+j)})((\mu_i - \mu_{(i+j)})^T\right) \\ &\quad + \frac{w_j}{w_{(i+j)}}\left(\Sigma_j + (\mu_j - \mu_{(i+j)})((\mu_j - \mu_{(i+j)})^T\right). \end{aligned} \quad (9)$$

The components to merge are chosen by the selecting the pair (G_i, G_j) whose replacement $G_{(i+j)}$ is most similar, in terms of the Kullback-Leibler divergence:

$$i^*, j^* = \underset{i,j}{\operatorname{argmin}} C_{ij} \quad (10)$$

$$C_{ij} = w_i\mathcal{KL}(G_i||G_{(i+j)}) + w_j\mathcal{KL}(G_j||G_{(i+j)}), \quad (11)$$

where the divergence between two multivariate Gaussians of dimension d is:

$$\begin{aligned} \mathcal{KL}(G_i||G_j) &= \frac{1}{2}\left(\log\frac{|\Sigma_j|}{|\Sigma_i|} + \mathrm{Tr}(\Sigma_j^{-1}\Sigma_i)\right. \\ &\quad \left. + (\mu_i - \mu_j)\Sigma_q^{-1}(\mu_i - \mu_j)^T - d\right). \end{aligned} \quad (12)$$

Importantly for active learning online, merging Gaussians and updating the cost matrix requires constant $\mathcal{O}(N_{max})$ computation every iteration once the initial cost matrix has been built. In contrast, learning a GMM with latent variables requires multiple expensive $O(n)$ expectation-maximization iterations [1]. The initial covariance parameter Σ is assumed to be uniform diagonal $\Sigma = \mathbf{I}\sigma$, and is estimated by leave-one-out cross validation using the large pool of unlabeled data in $\mathcal{U}$.

$$\hat{\sigma} = \underset{\sigma}{\text{argmax}} \left(\prod_{n \in \mathcal{U}} \sigma^{-\frac{d}{2}} \sum_{x \neq x_n} \exp -\frac{1}{2\sigma^2}(\mathbf{x} - \mathbf{x}_n)^2 \right). \quad (13)$$

Given the learned models $p(\mathbf{x}|y, \theta)$, we can classify $\hat{y} \leftarrow f(\mathbf{x})$, where

$$\begin{aligned} f(\mathbf{x}) &= \underset{y}{\text{argmax}} p(y|\mathbf{x}), \\ p(y|\mathbf{x}) &\propto \sum_i w_i \mathcal{N}(\mathbf{x}; \mu_{i,y}, \Sigma_{i,y}) p(y). \end{aligned} \quad (14)$$

3.2.2 SVM

We use a standard SVM approach with RBF kernels, treating multi-class classification as a set of 1-vs-all decisions, for which the decision rule [22] is given (by an equivalent form to (14)) as

$$f(\mathbf{x}) = \underset{y}{\text{argmax}} \left(\sum_{\mathbf{v}_i \in SV_y} \alpha_{y,i} \mathcal{N}(\mathbf{x}; \mathbf{v}_i) + \alpha_{y,0} \right), \quad (15)$$

and $p(y|\mathbf{x})$ can be computed via an optimization based on the binary posterior estimates [28].

3.3 Adapting Active Query Criteria

Our first concern is how to adaptively combine the query criteria online for discovery and classification. Our algorithm involves probabilistically *selecting* a query criteria Q_k according to some weights $\mathbf{w}$ ($k \sim \text{Multi}(\mathbf{w})$) and then sampling the query point from the distribution $i^* \sim p_k(i)$ ((3), (5-7)).[1] The weights $\mathbf{w}$ will be adapted based on the discovery and classification performance ϕ of our active learner at each iteration. In an active learning context, [17] show that because labels are few and biased, cross-validation is a poor way to assess classification performance, and suggest the unsupervised measure of binary *classification entropy* (CE) on the unlabeled set $\mathcal{U}$ instead. This is especially the case in the rare class context where there is often only one example of a given class, so cross-validation is not well defined. To overcome this problem, we generalize CE to multi-class entropy (MCE) of the classifier $f(\mathbf{x})$ and take it as our indication of classification performance,

$$H = -\sum_{y=1}^{n_y} \frac{\sum_i \text{I}(f(\mathbf{x}_i) = y)}{|\mathcal{U}|} \log_{n_y} \frac{\sum_i \text{I}(f(\mathbf{x}_i) = y)}{|\mathcal{U}|}, \quad (16)$$

where n_y is the number of classes observed so far. The intuition here is that in a rare-class scenario with extreme class imbalance, classifiers are typically at risk of bias toward the majority class. A classifier with a higher entropy on the unlabeled data shows less bias and is therefore likely to be generalize better than classifier with a more biased response. Next, we must also explicitly reward the discovery of new classes to jointly optimize classification and discovery. To drive future adaptation of query criteria, we therefore define a reward function $\phi_t(i)$ upon querying point i at time t as,

$$\phi_t(i) = \alpha \text{I}(y_i \notin \mathcal{L}) + (1 - \alpha) \frac{(e^{H_t} - e^{H_{t-1}}) - (1 - e)}{2e - 2}. \quad (17)$$

The first term rewards discovery of a new class, and the second term rewards an increase in MCE (where the constant factors ensure the range is 0 to 1) after labeling point i. The parameter α is the weighting prior for discovery vs. classification. Given the reward function $\phi_t(i)$, we define an update for the future weight w_{t+1} of each active criterion k,

$$w_{t+1,k}(q) \propto \lambda w_{t,k} + (1 - \lambda - \epsilon) \phi_t(i) \frac{p_k(i)}{p(i)} + \epsilon. \quad (18)$$

Here we define an exponential decay (first term) of the weight in favor of (second term) the current performance ϕ weighted by how strongly criteria k recommended the chosen point i, compared to the joint recommendation $p(i) = \sum_k p_k(i)$. λ is the forgetting factor which determines how quickly the weights adapt. The third term encourages exploration by diffusing the weights so every criterion is tried occasionally. In summary, this approach adaptively selects more frequently those criteria that have been successful at discovering new points and/or increasing MCE, thereby balancing discovery and classifier improvement so as to improve overall performance.

3.4 Adaptive Selection of Classifiers

Although we broadly expect the generative GMM classifier to have better initial performance, and the discriminative SVM classifier to have better asymptotic performance, the best classifier will vary with dataset and active learning iteration. The question is how to combine these classifiers [29] online for best performance given a specific training supervision budget. Cross-validation to determine reliability is infeasible because of lack of data; however we can again resort to the MCE over the training set $\mathcal{U}$ (16). In our experience, MCE is indeed indicative of generalization performance, but relatively crudely and non-linearly so. This makes approaches based on MCE weighted posterior fusion unreliable. We therefore choose a simpler but more robust approach which *switches* the final classifier at the end of each iteration to the one with higher MCE, aiming to perform as well as the better classifier for any supervision budget. Specifically, after each training iteration, having learned the n points in $\mathcal{L}$ and obtained parameters θ^n_{SVM} and θ^n_{GMM}, we compute multi-class classification entropies $H_{gmm}(f_{gmm}(\mathcal{U}; \theta^n_{gmm}))$ and $H_{svm}(f_{svm}(\mathcal{U}; \theta^n_{svm}))$ over the train set $\mathcal{U}$. If training is then terminated and we are asked to predict a test point $\mathbf{x}^*$, we predict as

1. We choose this method because each criterion has very different "reasons" for its preference. An alternative is querying a product or mean [17] of the criteria. That risks querying a merely moderately unlikely and uncertain point – neither outlying nor on a decision boundary – which is useless for either classification or discovery.

Algorithm 1 Active Learning for Discovery and Classification

Active Learning

Input: Initial labeled $\mathcal{L}$ and unlabeled $\mathcal{U}$ samples. Classifiers $\{f_c\}$, query criteria $\{\mathcal{Q}_k\}$, weights $\mathbf{w}$.

1) Build unconditional GMM $p(\mathbf{x})$ from $\mathcal{L} \cup \mathcal{U}$ (8)-(12)
2) Estimate σ by cross-validation on $p(\mathbf{x})$ (13)
3) Train initial GMM and SVM classifiers on $\mathcal{L}$

Repeat as training budget allows:

1) Compute query criteria $p_{lik}(i)$ (5) and $p_{unc}(i)$ (3)
2) Sample query criteria to use $k \sim \text{Multi}(\mathbf{w})$
3) Query point $i^* \sim p_k(i)$, add $(\mathbf{x}_{i^*}, y_{i^*})$ to $\mathcal{L}$
4) Update classifiers with label i^* (14) and (15)
5) Update query criteria weights $\mathbf{w}$ (17) and (18)
6) Compute entropies H_{gmm} and H_{svm} (16)
7) If $H_{gmm} > H_{svm}$: select classifier $f_{gmm}(\mathbf{x})$ (19)
8) Else: select $f_{svm}(\mathbf{x})$ (19)

Testing

Input: Testing samples $\mathcal{U}^*$, selected classifier c.

1) Classify $x \in \mathcal{U}^*$ with $f_c(x)$ ((14) or (15))

$$f(\mathbf{x}^*) = \begin{cases} f_{gmm}(\mathbf{x}^*; \theta^n_{gmm}) & H_{gmm} > H_{svm} \\ f_{svm}(\mathbf{x}^*; \theta^n_{svm}) & H_{gmm} \leq H_{svm} \end{cases}. \quad (19)$$

Additionally, the process of multi-class posterior estimation for SVMs [28] requires cross-validation and is inaccurate with limited data. To compute the uncertainty criterion (3) at each iteration, we therefore use posterior of the classifier determined to be more reliable by MCE, rather than always using the discriminative model posterior. This ensures that uncertainty sampling is as accurate as possible in both low and high data contexts.

3.5 Summary

Algorithm 1 summarizes our approach. Our algorithm has four parameters: Gibbs parameter β, discovery vs. classification prior α, forgetting rate λ and exploring rate ϵ. None of these were tuned to obtain good results; we set them all crudely to intuitive values for all experiments, $\beta = 100$, $\alpha = 0.5$, $\lambda = 0.6$ and $\epsilon = 0.02$. The GMM and SVM classifiers both have regularization hyperparameters N_{max} and (C, γ). These were not optimized[2], but set at standard values $N_{max} = 32$, $C = 1, \gamma = 1/d$.

4 Experiments

4.1 Illustrative Example

We first illustrate the operation of our model (Algorithm 1) by way of a synthetic example in Figure 2. This dataset contains a majority class organized in a ring (Figure 2(a), dots) and five Gaussian distributed rare classes (Figure 2(a), other symbols) around the majority class in geometrically descending prior proportion. On iteration 1 (Figure 2(a)) only the majority class has been sampled (Figure 2(a), large bold symbols indicate observations). The preferred points under the likelihood criterion (5) are those far from the current samples, (Figure 2(a), likelihood row), while the uncertainty criterion has no basis to choose yet, so there is no preference (Figure 2(a), uncertainty plot). On iteration 4 (Figure 2(b)) the likelihood criterion discovers an outlying rare class. The classification accuracy and hence likelihood criterion weight (18) are thus increased (Figure 2(b), top). The local region to this class is no longer preferred by the likelihood criterion (Figure 2(b), likelihood) and the region between the two known classes is preferred by the uncertainty criterion (Figure 2(b), uncertainty). In the next three iterations, the likelihood criteria is applied repeatedly and the other three outer rare classes are found (Figure 2(c)). Accuracy is greatly improved, and the likelihood criteria is weighted strongly due to its success (Figure 2(c), top). With four rare class regions accounted for, the remaining low-likelihood domain is in the central ring (Figure 2(c), likelihood). The likelihood criterion discovers the final rare class (Figure 2(d)) on iteration 13 – taking slightly longer because being within the ring, this rare class is near many majority distribution points. With no new classes to discover, the uncertainty criterion generates better rewards (via greater MCE increase (17)) and begins dominate, effectively refining the decision boundary (Figure 2(e)) and raising accuracy to 98%. Finally, by iteration 64, the model is close to peak performance, and without further reward, the weights of each criteria are returning to equilibrium. The the majority model has surpassed its maximum complexity $N_{max} = 32$. The larger blobs in Figure 2(f) now illustrate the pairs of points chosen for fusion (10).

4.2 UCI Data

4.2.1 Evaluation Procedure

In this section, we test our method on 7 standard datasets from the UCI repository [30]. These datasets were selected because they contained multiple classes in naturally unbalanced proportions, thereby representing real discovery and classification problems. In every case we started with one labeled point from the largest class and the goal was to discover and learn to classify the remaining classes. Table 1 summarizes the properties of each dataset. Performance was evaluated by two measures at each active learning iteration: i) the percentage of distinct classes in the training dataset discovered and ii) the average classification accuracy over all classes. Note that in contrast to (1), this accuracy measure ensures that ability to classify each rare class is weighted equally with the majority class despite the fewer rare class points. Moreover, it means that undiscovered rare classes automatically penalize accuracy. Accuracy was evaluated

2. Standard SVM optimization exploits cross-validation, but this is not feasible in this study as there are usually only a few examples of each rare class for most of the learning process. In any case the aim of our approach is to adapt between two imperfect classifiers.

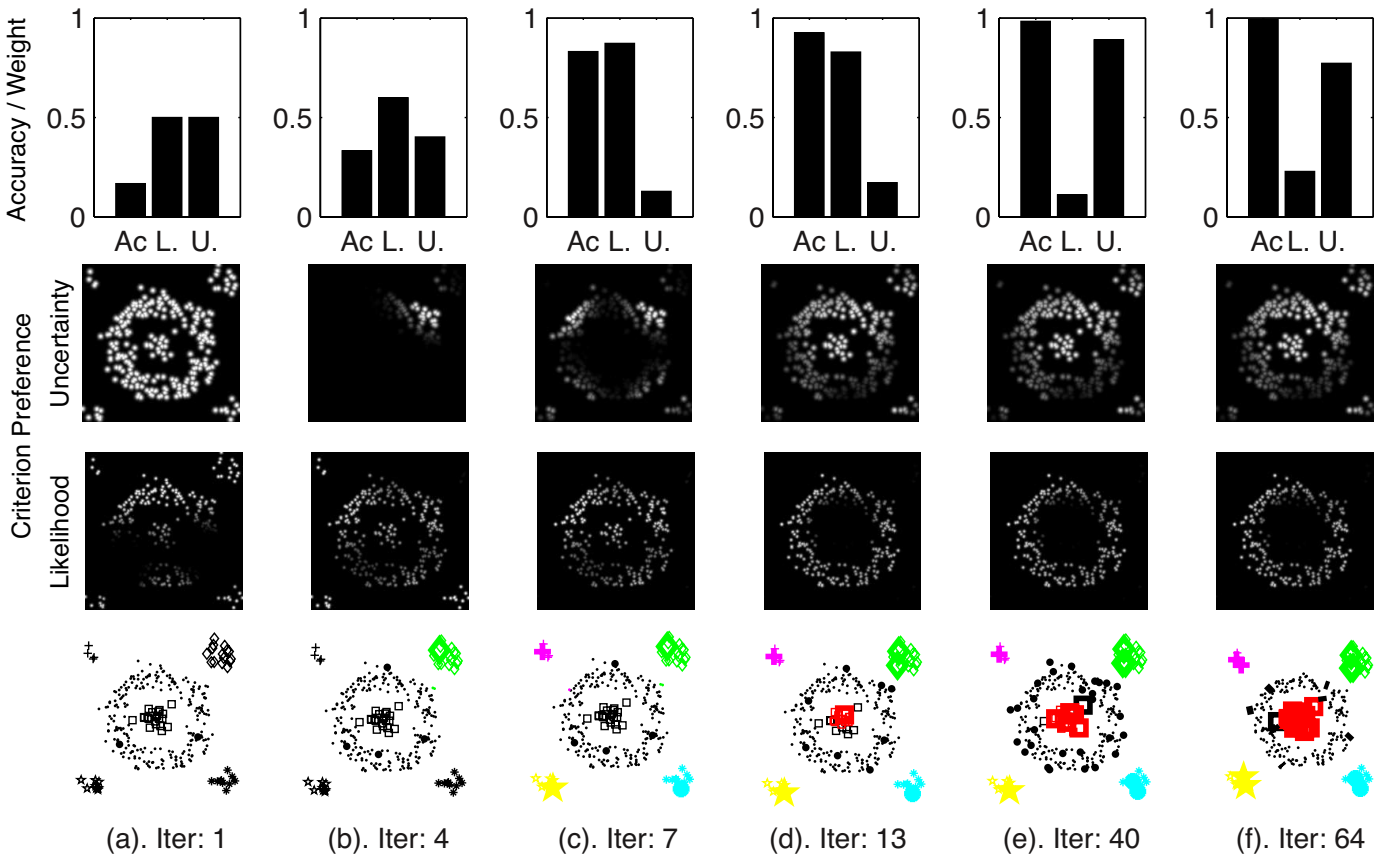


Figure 2. Illustrative synthetic example of rare class discovery and classification. Bottom: true class indicated by symbols, observed points by bold symbols, predicted class by shade/color. Second row: badly explained points preferred by likelihood criteria. Third row: ambiguous points preferred by the uncertainty criteria. Fourth row: accuracy and likelihood vs uncertainty criteria weighting.

Dataset	N	d	N_c	S%	L%
Ecoli	336	7	8	1.5%	42%
Pageblocks	5473	10	5	.5%	90%
Glass	214	10	6	4%	36%
Covertype	5000	10	7	3.6%	25%
Shuttle	20000	9	7	.01%	78%
Thyroid	7200	21	3	2.5%	92%
KDD99	33650	23	15	.04%	51%

Table 1
UCI dataset properties. (N) number of instances. (d) dimension of data. (N_c) number of classes. (S%/L%) proportions of smallest and largest classes.

by 2-fold cross-validation, averaged over 50 runs from random initial conditions – except for the shuttle dataset which is provided with a dedicated test set.

4.2.2 Discovery and Classification

We compared the following algorithms: **S/R**: A baseline SVM classifier with random queries. **G/G**: GMM classification, querying the GMM likelihood criterion (4). **S/S**: SVM classifier using the SVM uncertainty criterion (2). This corresponds to the strategy in [12], [31]. **S/GSmix:** SVM classifier alternating GMM likelihood and SVM uncertainty criteria. Note that this corresponds to the approach in [3]. **S/GSonline**: SVM classifier querying GMM likelihood & SVM uncertainty criteria fused by the method in [17]. **S/GSadapt**: SVM classification with adaptive fusion of GMM likelihood & SVM uncertainty criteria by our method in (16)-(18). **GSsw/GSadapt**: Our full model including online switching of GMM and SVM classifiers, as detailed in Algorithm 1.

Shuttle (Figure 3(a)). Our methods S/GSadapt and GSsw/GSadapt, exploit likelihood sampling early on for fast discovery, and hence early classification accuracy. They then switch to uncertainty sampling later on, and hence achieve higher asymptotic accuracy than the pure likelihood based G/G method. Figure 4(a) illustrates this process via the query criteria weighting (18) for one typical run. The likelihood criterion discovers a new class early, leading to higher weight (17) and rapid discovery of the remaining classes. After 75 iterations, with no new classes to discover, the uncertainty criterion obtains greater reward (17) and dominates, efficiently refining classification performance.

To provide some context for our discovery rate results, we re-plot the discovery rates for this dataset reported by some contemporary discovery studies [1], [18],

[19][3]. [1] exploits low likelihood similarly to our model, but under-performs due to the heuristic of spending a large fraction of the query budget on randomly selected points. The low likelihood criterion selects points based on dissimilarity to known classes. In contrast, [18] relies on a local gradient to identify rare classes. This will be successful depending on how much the dataset "looks like" a uniform majority class background with spikes of rare class data. In this case our low likelihood approach outperformed [18], but the better method will vary depending which assumption is better met by the data.

Thyroid (Figure 3(b)) This dataset has only two rare classes to be found, so classification chance is 33%. Our GSsw/GSadapt model has best classification performance here because of our two key innovations: adaptive sampling (Section 3.3) and switching classifiers (Section 3.4). The switching classifier permits GSsw/GSadapt to match the initially superior classification performance of the G/G likelihood-based model, and asymptotic performance of the SVM based models. Figure 5(a) illustrates switching via the average (training) classification entropy and (testing) classification accuracy of each of the classifiers composing GSsw/GSadapt. The GMM classifier entropy is greater than the SVM entropy for the first 25 iterations. This is approximately the period over which the GMM classifier has better performance than the SVM classifier, so switching classifier on training entropy allows the classifier pair to perform as well as the best classifier for that iteration. The adaptive weighting allows GSsw/GSadapt to rapidly switch to exploiting uncertainty sampling after the few classes have been discovered (Figure 4(b)) (contrast the shuttle dataset, which required more extensive use of likelihood sampling to discover all the classes). In contrast non-adaptive S/GSmix (corresponding to [3]) continues to "waste" half its observations on likelihood samples once all classes are discovered, and so is consistently and asymptotically outperformed by our GSsw/GSadapt.

Glass (Figure 3(c)). Again, our GSsw/GSadapt approach is competitive at discovery, and best at classification because it matches the good initial performance of the GMM classifier and asymptotic performance of the SVM classifiers by exploiting classifier switching (Figure 5(b)). Note the dramatic improvement over the SVM models in the first 50 iterations. **Page Blocks** (Figure 3(d)). Here our adaptive classification methods are not the fastest to detect all classes, but they still show good overall classification. **Covertype** (Figure 3(e)) Our GSsw/GSadapt model performs best because it explores new classes as quickly as purely likelihood based G/G (green) model, but outperforms G/G later on by also refining the decision boundary. Interestingly, SVM classifiers perform poorly in general on this dataset (up to

3. Vatturi et al. [19] use a complex discovery criterion based on hierarchically clustering the data, in contrast to our simpler flat clustering based likelihood criteria. Our framework is agnostic to the details of the procedure used to generate the query preference p_l, and we expect that using their criteria would improve the rest of our results accordingly.

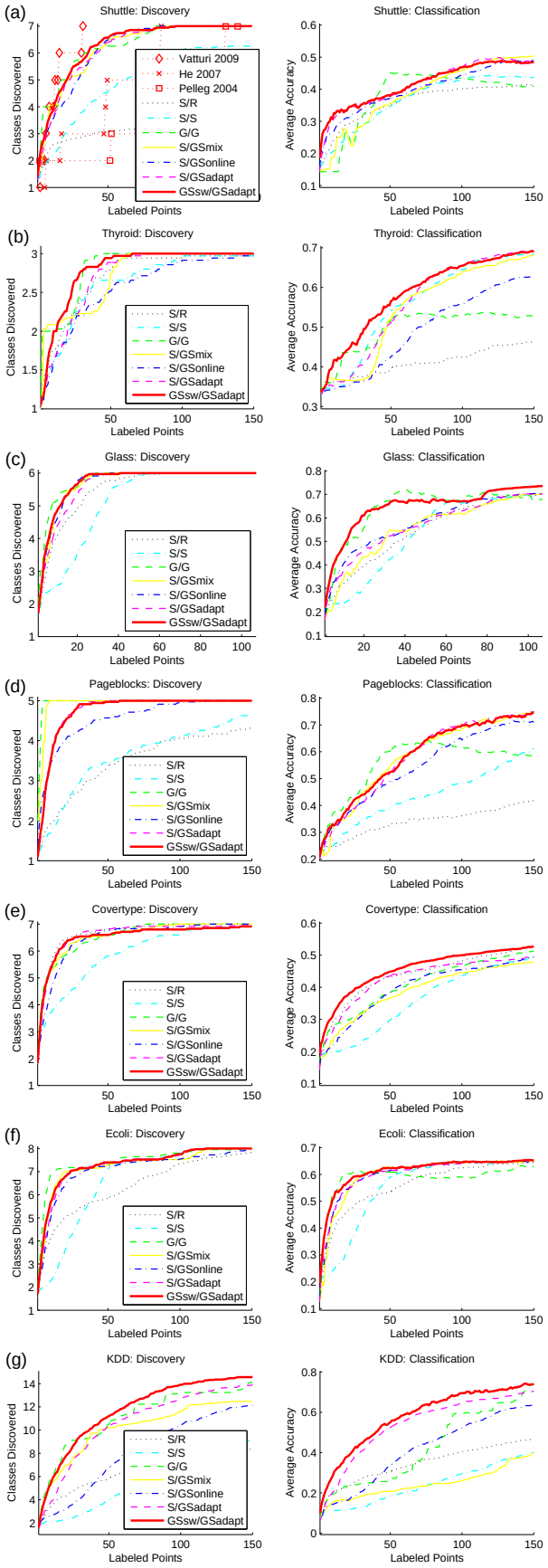


Figure 3. Discovery and classification performance for UCI datasets. (a) Shuttle, (b) Thyroid, (c) Glass, (d) Pageblocks, (e) Covertype, (f) Ecoli, (g) KDD.

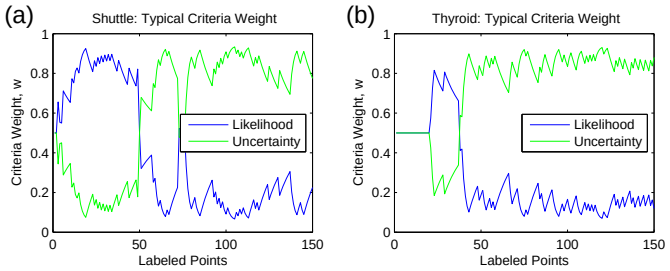


Figure 4. Criteria weight adaptation for (a) shuttle and (b) thyroid datasets.

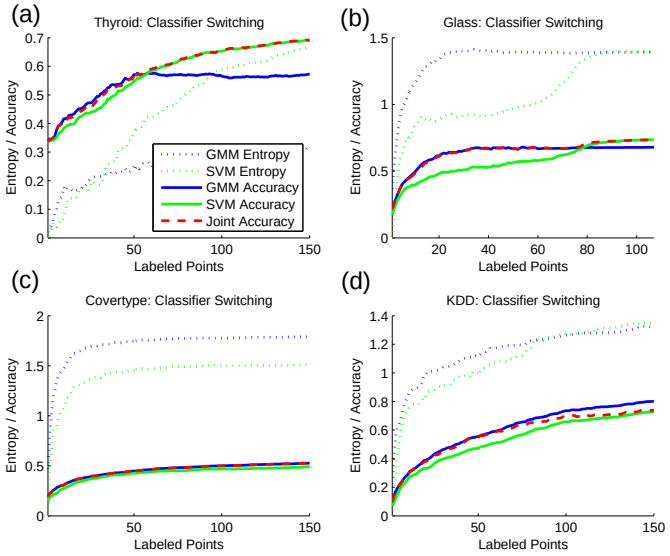


Figure 5. Classifier switching by multi-class classification entropy.

the first 150 data points). Based on the MCE classifier switching criteria, our GSsw/GSadapt model uses GMM classification throughout (Figure 5(c)), which helps it to outperform all the other SVM based models. **KDD** (Figure 3(g)). The KDD network intrusion dataset is the largest UCI dataset in dimension, number of classes, and extent of class skew. Our proposed models are competitive at discovery rate, and best at classification. In this example, our MCE-based classifier switching criteria fails to perform exactly as desired. The GMM classifier is better throughout, however by iteration 100 the MCE of the SVM classifier is greater, and our GSsw/GSadapt model switches to SVM classification prematurely (Figure 5(d)). In total the combined MCE-based switching classifier used by GSsw/GSadapt outperformed both of the individual component GMM and SVM classifiers (in terms of AUC) for 5 of the 7 datasets.

In summary the G/G method using likelihood criterion was usually the most efficient at discovering classes – as expected. However, it was usually asymptotically weaker at classifying new instances. This is because the generative model mis-specification tends to cost more with increasing amounts of data [8]. S/S, solely using uncertainty criteria, was always poor at discovery (and hence classification). Alternating between likelihood and uncertainty sampling, S/GSmix (corresponding to [3]) did a fair job of both discovery and classification on average, but under-performed our adaptive models due to its inflexibility. S/GSonline (corresponding to [17]) was generally better than random or S/S, and had decent asymptotic performance, but was not the quickest learner. Our first model S/GSadapt, which solely adapted the multiple active query criteria, was competitive at discovery, but sometimes not the best at classification in early phases with very little data. This is due to exclusively using the discriminative SVM classifier. Finally, adding generative-discriminative classifier switching, our complete GSsw/GSadapt model was consistently the best classifier over all stages of learning.

4.2.3 *Quantitative Performance Summary*

The standard approach to quantitatively summarizing the (time-varying) performance of active learning algorithms is to compute the area under their classification curve (AUC) during learning [17], [25]. Table 2 quantitatively summarizes the performance of each model in terms of the AUC means and standard deviations over the trials. The left columns represent prior approaches, and right columns represent the models introduced in this paper. Of the comparison models, there is no consistent best performer with G/G, S/S, S/GSmix and S/GSonline performing best on 3, 1, 2 and 1 datasets respectively. Moreover, each model performs poorly (last or second to last) on at least one dataset. This supports our earlier insight that a big challenge of this problem is the strong dataset dependence of the ideal query criteria.

Overall, our first contribution S/GSadapt performs competitively in each case, and our second model GSsw/GSadapt performs best on all datasets. The performance standard deviations of all models are fairly large, reflecting the significance of the random initialization and selection, but we note that the standard deviations of our GSsw/GSadapt are among the lowest, indicating consistent good performance. Finally, we indicate the statistical significance of the performance improvement of our GSsw/GSadapt over each comparison model as computed by two-sided t-test.

4.3 Vision Data

In this section we apply our approach to two vision datasets, the MNIST handwritten digits dataset[4] and the human gait dataset[5]. **MNIST digits:** This dataset has 60,000 examples of 28x28 pixel handwritten digit images in ten classes. We reduce the number of dimensions to 25 using PCA. To create a rare class scenario, we subsample the full dataset to produce 13000 images in geometrically imbalanced training proportions, as is standard practice for evaluation of rare class discovery

4. http://yann.lecun.com/exdb/mnist/
5. http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp

	Area Under Classification Curve					
Data	G/G	S/S [12], [31]	S/GSmix [3]	S/GSonline [17]	S/GSadapt	GSsw/GSadapt
Ecoli	$59 \pm 2.8^{**}$	$55 \pm 2.4^{**}$	$60 \pm 2.1^{**}$	$60 \pm 2.2^{**}$	60 ± 2.3	$\mathbf{61} \pm 1.8$
Pageblocks	$56 \pm 7.0^{*}$	$44 \pm 6.7^{**}$	58 ± 3.1	$54 \pm 7.0^{**}$	58 ± 4.1	$\mathbf{59} \pm 4.6$
Glass	64 ± 1.1	$53 \pm 4.9^{**}$	$56 \pm 6.0^{**}$	$58 \pm 6.4^{**}$	57 ± 6.6	$\mathbf{65} \pm 3.6$
Covertype	$41 \pm 2.0^{**}$	$36 \pm 4.1^{**}$	$39 \pm 3.1^{**}$	$40 \pm 2.1^{**}$	43 ± 3.2	$\mathbf{46} \pm 2.6$
Shuttle	$38 \pm 4.2^{**}$	$36 \pm 9.2^{**}$	$39 \pm 1.3^{**}$	$41 \pm 2.2^{*}$	41 ± 1.9	$\mathbf{42} \pm 1.9$
Thyroid	$50 \pm 1.8^{**}$	$56 \pm 7.4^{*}$	$55 \pm 2.8^{**}$	$50 \pm 8.6^{**}$	55 ± 5.7	$\mathbf{59} \pm 4.3$
KDD99	$42 \pm 6.4^{**}$	$32 \pm 10^{**}$	$31 \pm 11^{**}$	$41 \pm 7.0^{**}$	49 ± 7.6	$\mathbf{59} \pm 5.5$

Table 2
UCI data classification performance: means and standard deviations of AUC. Superscripts * and ** indicate respectively statistically significant improvement in performance by GSsw/GSadapt at $p < 0.05$ and $p < 0.01$ or higher.

Dataset	N	d	N_c	S%	L%
MNIST digits	13000	25	10	.1%	50%
CASIA Gait	2353	25	9	3%	49%

Table 3
Vision dataset properties. (N) number of instances. (d) dimension of data. (N_c) number of classes. (S%/L%) proportions of smallest and largest classes.

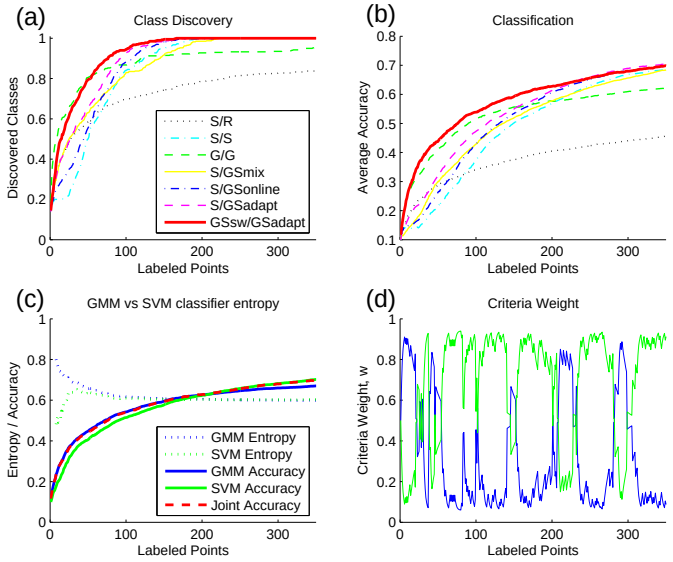


Figure 6. Digits dataset. (a) Discovery rate, (b) classification rate, (c) entropy based classifier switching, (d) selected query strategy.

methods [1], [19]. Specifically, digit 0 gets 4096 examples and every subsequent digit gets half as many such that the "rarest" digit 9 gets only eight examples. **Gait View:** The gait dataset has 2353 video sequences of 128 subjects at 9 different angles of view from 18 to 162 degrees. We address the view angle recognition problem, so the goal is to learn to classify the viewpoint of an observation, and the different subjects provide intra-class variability. We extract a gait energy image (GEI) representation of each gait cycle in the video according to [32]; but we truncate the image to focus on most informative leg region. Again, we reduce the number of dimensions to 25 with PCA, and resample the data geometrically to create proportions of 200 images for the largest class (18 degrees) to 12 for the smallest class (164 degrees). The dataset properties are summarized in Table 3.

4.3.1 Handwritten Digits

GSsw/GSadapt outperforms all the others at rare digit discovery and classification (Figure 6(b), Table 4). The early class discovery rate is good (competitive with the best, purely likelihood based G/G) because it can exploit likelihood based sampling early on (Figure 6(a) and (d), iterations 1-40) to rapidly discover most of the classes; later it outperforms G/G because it can also refine decision boundaries with uncertainty based sampling (Figure 6(d), iterations 50-100). Finally, as the training data grows, the SVM begins to learn a better decision boundary than the GMM, and it switches classification strategy appropriately (iterations 150+). Figure 6(c) illustrates the actual performance of our GSsw/GSadapt tracking the better of the GMM or SVM performance.

To illustrate the learning process, Figure 7 shows some illustrative decisions made by our model during the first 20 iterations of a typical run. The likelihood expert (5) is selected first, and the most dissimilar example to the initial training 0 is new class instance, 2. This success reinforces the likelihood criteria (18), which is selected repeatedly. Digits 1 and 3 are then discovered as they appear very different to the two known classes so far. The uncertainty expert (3) is eventually selected, querying an understandably uncertain smudged 2, thereby increasing the generality of the model for 2s. The likelihood expert next selects an 8, which is again unlike any labeled examples it knows so far. The uncertainty expert next selects a stretched 0 and a slanted 1, further refining the distribution of these classes. Finally, the likelihood expert also queries a 2 that is very much unlike the others seen so far, illustrating the additional value of this strategy for discovering other modes or clusters of known classes.

4.3.2 Gait View

For gait view classification, GSsw/GSadapt is again the best model (Figure 8(b), Table 4). The data contains outliers, so the likelihood criteria (and hence G/G) are unusually weak at discovery. GSsw/GSadapt adapts well

	Area Under Classification Curve					
Data	G/G	S/S [12], [31]	S/GSmix [3]	S/GSonline [17]	S/GSadapt	GSsw/GSadapt
MNIST digits	$51 \pm 1.4^{**}$	$48 \pm 1.8^{**}$	$50 \pm 1.9^{**}$	$51 \pm 2.2^{**}$	54 ± 2.0	$\mathbf{57} \pm 1.1$
CASIA Gait	$40 \pm 2.8^{**}$	$40 \pm 5.7^{**}$	$35 \pm 4.9^{**}$	$47 \pm 3.7^{**}$	46 ± 4.5	$\mathbf{57} \pm 2.2$

Table 4

Vision data classification performance: means and standard deviations of AUC. Superscripts * and ** indicate respectively statistically significant improvement in performance by GSsw/GSadapt at $p < 0.05$ and $p < 0.01$ or higher.

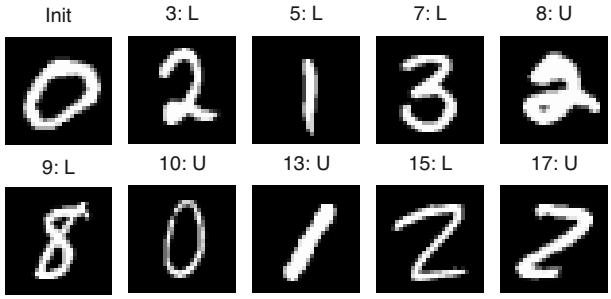


Figure 7. Discovering and learning digits. Labels indicate iteration number, and whether the instance was queried by the (l) likelihood or (u) uncertainty criterion.

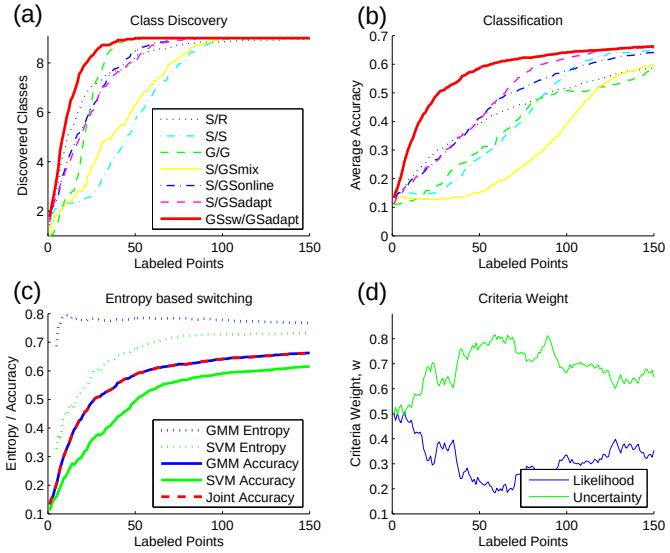


Figure 8. Gait view dataset. (a) discovery rate, (b) classification rate, (c) entropy based classifier switching (d) query criteria weights (average).

to this data by exploiting uncertainty sampling criteria extensively (Figure 8(d)). SVM classification is generally poor, especially in the first half of learning while data is very scarce. The classifier switching strategy for GSsw/GSadapt correctly tracks the better GMM classifier throughout the 150 iterations tested (Figure 8(c)).

Figure 9 shows the first five decisions made in a typical run. In this case uncertainty is actually good at discovering new classes; possibly because the classes here are actually a continuum, so new classes are often found "between" existing classes. The agglomerative process of the incrementally built GMM (Section 3.2)

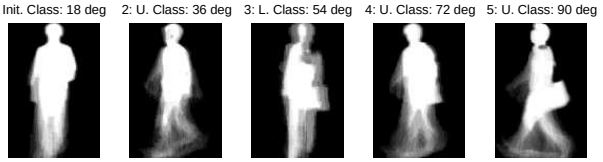


Figure 9. Discovering and learning view angles. Labels indicate iteration, angle and if the instance was queried by the (l) likelihood criteria or (u) uncertainty criterion.

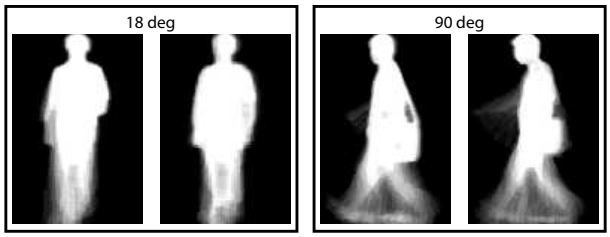


Figure 10. Similar gait images chosen for merging

is illustrated in Figure 10 by way of pairs of images selected for fusion into a single kernel (10).

4.4 Contrast to Sequential Methods

We have argued in favor of our adaptive approach to the joint discovery and classification problem compared to simpler sequential approaches. That is, approaches applying an active learning criteria suitable for discovery for a fixed number of iterations, followed by a criteria suitable for learning. Sequential approaches risk performing poorly due to different and unknown proportions of each criteria being ideal for each dataset. In this section, we verify that this is indeed a problem in practice. Specifically, we compare the performance of a series of sequential models (**S/GSseq**) using 25, 50, 100 and 150 discovery queries (likelihood criteria, (5)) before spending the remaining budget (of 150) on learning queries (uncertainty criteria, (3)).

Table 5 summarizes the classification AUC for each dataset, with the best scoring model for each highlighted. Clearly, there is significant variety in the ideal number of discovery iterations across datasets. For example, thyroid, ecoli and gait datasets are all fairly quick to discover all the classes (see discovery curves in Figure 3(b) and (f) , Figure 8(a)), so performance is better with fewer discovery iterations. In contrast, there are many rare classes in dataset KDD, and discovery is fairly slow (Figure 3(g)) and more discovery iterations help. Datasets shuttle, pageblocks and digits are in the middle,

with all the classes being discovered around iteration 50 (Figure 3(a) and (d)) and 100 (Figure 6(a)). Only for the case of pageblocks with 50 discovery queries did any of the sequential models meet the performance of our proposed adaptive model GSsw/GSadapt. This diversity of results highlights the issue that for open ended data mining problems where the number of classes to be discovered is unknown, any particular chosen sequential method may perform poorly. In contrast, the consistently good performance of our adaptive model highlights the value of our contribution.

To provide some further context we also show the performance for sequential models based on 25 iterations[6] of two contemporary discovery methods [18] and [20] followed by SVM uncertainty sampling (Table 5). Note that this is not a fair comparison because [18], [20] exploit additional information not available to the other models or indeed for real data mining problems: the number of rare classes and their prior proportions. Despite this disadvantage, our model is best for 7 of 9 datasets, and it generally performs consistently well whereas the two sequential schemes perform very poorly for at least one dataset each.

4.5 Density Weighting

Our final experiment investigates how density weighting (Section 3.1) of query criteria affects performance. We compared our best GSsw/GSadapt model which adapts uncertainty (3) and likelihood criteria (5) against two variants which include a third criterion – density weighted uncertainty (DWU) (6) or density weighted likelihood (DWL) (7) sampling respectively. If density weighted criteria are effective for some or all of the learning process [25], the associated weights will be increased and the criteria exploited. In general, poor criteria will tend towards being ignored rather than seriously hindering performance. However, because of the need to explore each criteria (Section 3.3) occasionally in order to determine its value, adding each additional criteria does impose a cost to performance.

Figure 11 summarizes the effect of density weighting. Uncertainty weighting turns out to be consistently detrimental. The mean weight over all datasets assigned to DWL and DWU were 0.19 and .25 respectively, showing that the weighted criteria were indeed adapted down (compared to uniform weight of $1/3$). Likelihood weighting was generally better than uncertainty weighting (intuitively, as unweighted likelihood is more prone than uncertainty to query outliers). Nevertheless, it only improves performance compared to GSsw/GSadapt in a minority of cases, notably the gait dataset which we already observed contains outliers.

The poor performance of both DWL and DWU is understandable: since rare classes are by definition underrepresented, they tend to occur in low density regions of the input space, and so are not discovered by DWL or refined by DWU. This is an important result and insight, because it shows that although widely supported for general active learning [16], [24], [25], [26], density weighting is usually detrimental for rare-class problems.

6. Only 25 iterations are used because these algorithms terminate after all classes are discovered.

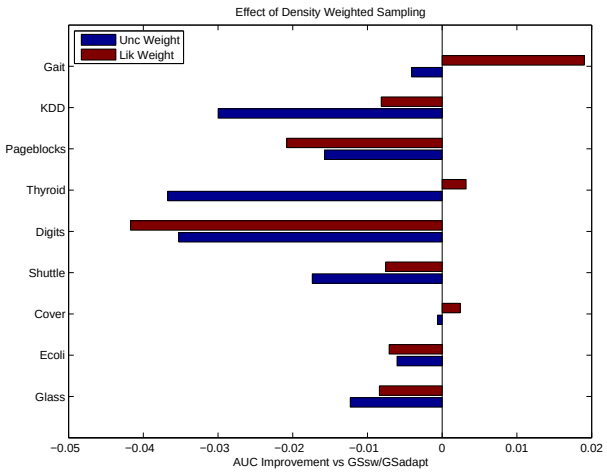


Figure 11. Effect of including density weighted likelihood or entropy criterion in GSsw/GSadapt framework.

5 Conclusion

5.1 Summary

We have proposed an algorithm for active learning to classify a priori undiscovered classes based on adapting two query criteria and choosing classifiers. To switch generative and discriminative classifiers we used a multi-class generalization of unsupervised classification entropy. Classifier learning in the presence of undiscovered classes was achieved by formulating a new model driven by an adaptive mixture of new class seeking and multi-class entropy maximization.

In our evaluation on nine datasets of widely varying domain, size and dimension, our model was consistently able to adapt query criteria and classifier online as more data was obtained, thereby outperforming other contemporary approaches making less efficient use of their active query budget (notably non-adaptively iterating over criteria [3], or sequentially applying discovery and then learning criteria). We therefore expect our approach to be of great practical value for many problems. Our active learning approach is also cheap compared to alternative active learning criteria, e.g., expected generalization error which requires $\mathcal{O}(n^3)$ per iteration [14] vs. our $\mathcal{O}(n)$. Our approach is also compatible with sub-sampling techniques for pool based active learning such as the "59 trick", which defines a constant time approximation to the full algorithm [31].

5.2 Discussion

In this work, we have constructed generative and discriminative models in parallel, exploited their properties

	S/GSseq25	S/GSseq50	S/GSseq100	S/GSseq150	NNDM25 [18]	RADAR25 [20]	GSsw/GSadapt
Ecoli	$\mathbf{60} \pm 2.2$	60 ± 3.1	59 ± 4.1	56 ± 5.1	55 ± 3.5	57 ± 2.2	61 ± 1.8
P.Blocks	58 ± 3.0	$\mathbf{59} \pm 4.5$	56 ± 3.8	54 ± 3.9	47 ± 3.5	44 ± 4.7	59 ± 4.6
Glass	55 ± 5.8	55 ± 5.5	55 ± 5.0	$\mathbf{55} \pm 5.0$	63 ± 4.2	63 ± 4.2	65 ± 3.6
C.Type	39 ± 1.5	40 ± 2.2	40 ± 2.5	$\mathbf{41} \pm 2.5$	39 ± 2.3	36 ± 2.3	46 ± 2.6
Shuttle	39 ± 1.0	$\mathbf{40} \pm 1.1$	39 ± 1.1	38 ± 1.1	44 ± 2.0	45 ± 2.5	42 ± 1.9
Thyroid	$\mathbf{57} \pm 2.1$	54 ± 1.5	49 ± 1.6	47 ± 1.5	59 ± 2.3	47 ± 8.2	59 ± 4.3
KDD99	29 ± 11	27 ± 13	32 ± 12	$\mathbf{34} \pm 8.2$	32 ± 5.8	17 ± 8.2	59 ± 5.5
Digits	52 ± 1.8	52 ± 1.7	$\mathbf{53} \pm 1.3$	48 ± 1.5	47 ± 3.1	43 ± 3.2	57 ± 1.1
Gait	$\mathbf{39} \pm 3.5$	38 ± 1.3	31 ± 0.9	28 ± 0.8	60 ± 2.5	51 ± 3.0	57 ± 2.2

Table 5
Classification performance for sequential active discovery followed by active learning: means and standard deviations of AUC. The digits in each column title indicate the number of discovery iterations used for the sequential models.

synergistically for active discovery and learning, and switched them adaptively for classification. We note that the improvement we were able to obtain by classifier switching supports the GMM vs SVM contrast made in [22]; but exploits it in a more automated way than [22], which required manually weighting the two models. A related body of work has tried to construct more closely integrated model pairs, or single hybrid models to obtain some benefits of each model type in various combinations. One approach is to optimize generative model parameters to maximize discriminative classification performance rather than data likelihood [33], [34], which allows some generative models to perform beyond their normal asymptotic performance. This is different to our problem because we are particularly interested in low data performance and also wish to maintain accurate models of each type for their role as query criteria. Another approach is to use byproducts of generative model learning – such as Fisher information, as features to train discriminative models [34], [35]. This is a promising avenue, but not straightforwardly applicable to our setting as we have a variable number of parameters per class in our generative model.

Another related area of research to this study is that of learning from imbalanced data [36] which aims to learn classifiers for classes with very imbalanced distributions, while avoiding the pitfall of simply classifying everything as the majority class. One strategy to achieve this is uncertainty based active learning [31], which works because the distribution around the class boundaries is less imbalanced than the whole dataset. Our problem is also an imbalanced learning problem, but more general in that the rare classes must also be discovered, so we therefore effectively generalize [31].

Although our approach lacks the theoretical bounds of the fusion method in [17], we find it more compelling for various reasons: it jointly optimizes searching for new classes and refining their decision boundaries, and it adapts based on the current state of the learning process, typically (but not always) class discovery via likelihood early on, and boundary refinement via uncertainty later. In contrast [17] solely optimizes classification accuracy and is not directly applicable to class discovery.

Some other adaptive methods [24], [25], [26] address the fusion of uncertainty and density criteria (to avoid outliers) for classifier learning. [24], [26] sample from a fixed weighted sum of density and uncertainty criteria. This is less powerful than our approach because it does not adapt the weighting online based on the performance of each criteria. Most importantly they all prefer high density points, which we have shown in this study to be the wrong intuition for rare class problems which require low likelihood points instead.

Relative to other active rare class discovery work [1], [18], [19], [20] our framework solves a more general problem of joint discovery and classification by adapting two criteria. A different active discovery intuition is exploited in [18]: using local gradient to detect non-separable rare classes. We derived an analogous query criterion based on GMM local gradient and integrated it into our framework. However, it was generally weaker than likelihood-based discovery (and was hence adapted downward in our framework) for most datasets, so we do not report on it here. Finally, unlike our work here, many related studies including [1], [18], [20], [25] rely on the strong assumption that the user specifies the number and prior proportion of classes in advance. This is a fatal assumption for the open ended data mining problem considered here, where one does not know the classes in advance as they may correspond to previously unknown types of fraud or intrusions etc.

5.3 Future Work

There are various interesting questions for future research including: further theoretical analysis and grounding of the joint discovery-classification problem and algorithms introduced here; how well our fusion methods generalize to other generative-discriminative pairs and query criteria; and how to create tighter coupling between the generative and discriminative classifiers [22]. A final key goal is to generalize some of the contributions we have discussed in this paper to the domain of online – rather than pool-based – active learning, which is a more natural setting for some practical problems [37] where online real-time classification is required and new classes may appear over time.

REFERENCES

[1] D. Pelleg and A. Moore, "Active learning for anomaly and rare-category detection," in *Neural Information Processing Systems*, 2004.

[2] S. Bay, K. Kumaraswamy, M. G. Anderle, R. Kumar, and D. M. Steier, "Large scale detection of irregularities in accounting data," in *International Conference on Data Mining*, 2006, pp. 75–86.

[3] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman, "Aladin: Active learning of anomalies to detect intrusions," MSR, Tech. Rep. 2008-24, 2008.

[4] T. Hospedales, S. Gong, and T. Xiang, "A markov clustering topic model for behaviour mining in video," in *IEEE International Conference on Computer Vision*, 2009.

[5] T. Hospedales, J. Li, S. Gong, and T. Xiang, "Identifying rare and subtle behaviours: A weakly supervised joint topic model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[6] B. Settles, "Active learning literature survey," University of wisconsin–Madison, Tech. Rep. 1648, 2009.

[7] M. Wang and X.-S. Hua, "Active learning in multimedia annotation and retrieval: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 10:1–10:21, 2011.

[8] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Neural Information Processing Systems*, 2001.

[9] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85–126, 2004.

[10] R. Sillito and R. Fisher, "Incremental one-class learning with bounded computational complexity," in *International Conference on Artificial Neural Networks*, 2007.

[11] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 762–769.

[12] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *International Conference on Machine Learning*, 2000.

[13] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *International Conference on Learning Theory*, 1992.

[14] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *International Conference on Machine Learning*, 2001, pp. 441–448.

[15] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, pp. 129–145, 1996.

[16] M. Wang, X.-S. Hua, Y. Song, J. Tang, and L.-R. Dai, "Multi-concept multi-modality active learning for interactive video annotation," in *Proc. Int. Conf. Semantic Computing ICSC*, 2007.

[17] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 255–291, 2004.

[18] J. He and J. Carbonell, "Nearest-neighbor-based active learning for rare category detection," in *Neural Information Processing Systems*, 2007.

[19] P. Vatturi and W.-K. Wong, "Category detection using hierarchical mean shift," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 847–856.

[20] H. Huang, Q. He, J. He, and L. Ma, "Radar: Rare category detection via computation of boundary degree," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2011.

[21] J.-H. Xue and D. M. Titterington, "Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes"," *Neural Process. Lett.*, vol. 28, no. 3, pp. 169–187, 2008.

[22] T. Deselaers, G. Heigold, and H. Ney, "Object classification by fusing svms and gaussian mixtures," *Pattern Recognition*, vol. 43, no. 7, pp. 2476–2484, 2010.

[23] C. Campbell, N. Cristianini, and A. Smola, "Query learning with large margin classifiers," in *International Conference on Machine Learning*, 2000.

[24] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan, "Multi-criteria-based active learning for named entity recognition," in *Association for Computational Linguistics*, 2004.

[25] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *European Conference on Machine Learning*, 2007.

[26] N. Cebron and M. R. Berthold, "Active learning for object classification: from exploration to exploitation," *Data Min. Knowl. Discov.*, vol. 18, no. 2, pp. 283–299, 2009.

[27] J. Goldberger and S. Roweis, "Hierarchical clustering of a mixture model," in *Neural Information Processing Systems*, 2004.

[28] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[29] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[30] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/ml/

[31] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the border: active learning in imbalanced data classification," in *ACM conference on Conference on information and knowledge management*, 2007.

[32] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[33] R. Raina, Y. Shen, A. Ng, and A. McCallum, "Classification with hybrid generative/discriminative models," in *Neural Information Processing Systems*, 2003.

[34] A. D. Holub, M. Welling, and P. Perona, "Combining generative models and fisher kernels for object recognition," in *IEEE International Conference on Computer Vision*, 2005, pp. 136–143.

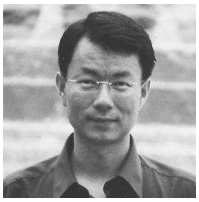
[35] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Neural Information Processing Systems*, 1998, pp. 487–493.

[36] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Transactions on Data and Knowledge Engineering*, vol. 21, no. 9, pp. 1263 –1284, 2009.

[37] C. C. Loy, T. Xiang, and S. Gong, "Stream based active anomaly detection," in *Asian Conference on Computer Vision*, 2010.

Timothy Hospedales received the Ph.D degree in Neuroinformatics from University of Edinburgh in 2008. He is currently a postdoctoral researcher with the School of Electronic Engineering and Computer Science, Queen Mary University of London. His research interests include probabilistic modeling and machine learning applied variously to computer vision, data mining, sensor fusion, human-computer interfaces and neuroscience.

Shaogang Gong is Professor of Visual Computation at Queen Mary University of London, a Fellow of the IEE and of the BCS. He received his D.Phil in computer vision from Oxford University in 1989. He has published over 200 papers in computer vision and machine learning. His work focuses on video analysis; object detection, tracking and recognition; face and expression recognition; gesture and action recognition; visual behaviour profiling and recognition.



Tao Xiang received the Ph.D degree in electrical and computer engineering from the National University of Singapore in 2002. He is currently a lecturer in the School of Electronic Engineering and Computer Science, Queen Mary University of London. His research interests include computer vision, statistical learning, video processing, and machine learning, with focus on interpreting and understanding human behaviour.