Probabilistic Active Learning with Structure-Sensitive Kernels

Dominik Lang[1], Daniel Kottke[2], Georg Krempl[1], and Bernhard Sick[2]

[1] KMD Lab, Faculty of Computer Science,
Otto-von-Guericke University, Magdeburg, Germany
{dominik.lang / georg.krempl}@ovgu.de
[2] IES Group, Faculty of Computer Science,
University of Kassel, Germany
{daniel.kottke / bsick}@uni-kassel.de

Abstract. This work proposes two approaches to improve the pool-based active learning strategy '*Multi-Class Probabilistic Active Learning*' (McPAL) by using two kernel functions based on Gaussian mixture models (GMMs). One uses the kernels for the instance selection of the McPAL strategy, the second employs them in the classification step. The results of the evaluation show that using a different classification model from the one that is used for selection, especially an SVM with one of the kernels, can improve the performance of the active learner in some cases.

Keywords: active learning, gaussian mixture, kernel function, support vector machine, McPAL

1 Introduction & Motivation

Active learning (AL) is a special case of semi-supervised machine learning, in which a learning algorithm has both labeled and unlabeled data available to it and is able to acquire the true labels of instances from an external source, in most cases one or multiple human agents. Since the number of labels that can be acquired is limited due to the cost that the acquisition entails, AL strategies aim to select instances that maximize the learners classification performance while being efficient with respect to the costs. A pool-based AL strategy named '*Multi-Class Probabilistic Active Learning*' (McPAL) [10] has shown to outperform competing strategies. This paper investigates the possibility of improving the performance of the method by including the information captured by a Gaussian mixture model (GMM) into the active learner. To achieve this, two kernel functions that are based on a GMM are used. These structure-sensitive kernel functions, based on the GMM [1] and RWM [17] distance measures, are leveraged by the active learner in two different ways: (1) by being included in the computation of the McPAL score, (2) by being used in the model that performs classification based on the sampled set of labeled instances. These approaches are compared to the original McPAL method as well as random sampling in a

series of experiments on one artificial data set and nine real-world data sets from the UCI machine learning repository [14].

2 Related Research

In active learning, various criteria have been proposed to determine which instances are most helpful to learn a classification model. One of the most common is the model's uncertainty regarding the classification of a sample. A strategy that solely relies on this criterion is known as uncertainty sampling (US) [13, 18]. In their application of US to SVMs, Tong and Koller [20] motivated that the goal is to approximately halve the version space through selecting instances that lie closest to the current decision boundary of the classifier. To extend this to multi-class problems, many methods have been proposed, for example, 'Best-versus-Second-Best' [8, 9, 11] (also referred to as 'Margin Sampling' [18]) or Entropy-based sampling [8, 18, 9]. Solely relying on this criterion to select instances has been shown to be prone to being 'locked in', ignoring possibly informative instances in favor of refining the current decision boundary [18, 12]. Hence, various approaches have been proposed that, in addition to uncertainty, also include other criteria. These include, for example, the diversity of the sampled instances [2, 5] or the density around a candidate instance [3, 4]. A promising AL strategy is Multi-class Probabilistic Active Learning (McPAL) [10], which has shown promising results compared to other approaches. It combines the density, the class posterior probability and the number of already sampled instances in the neighborhood of a candidate instance to estimate the potential gain of acquiring an instance's label. Instances that entail the highest potential gain are selected for labeling by the strategy. This acquisition is performed in a one-by-one fashion.

However, the selection does not have to be solely based on supervised models but can also use unsupervised approaches. Known clustering algorithms like k-medoid [15] or hierarchical clustering [6] as well as generative models like GMMs [16, 7] can be used to model the structure of the data and include it in the selection process.

The classification models, that are used in the process of instance selection, are often used for training the final classifier. Tomanek & Morik [19] investigated, to which degree the bias towards the learning algorithm used in the selection process affects, what they call *label reusability* - i.e., the training of a classifier, other than the one used for selection, on the acquired labeled data. They introduced the terms of *selector* and *consumer* classifiers to describe the model used for selecting instances, and performing classification based on them, respectively. Contrary to their initial assumptions, they concluded that *self-selection* (the selector and consumer classifier are the same) is in fact not in all cases the best choice.

3 Using GMM-based Kernels with the McPAL Strategy

Since the majority of data available in the scenario of AL is unlabeled and therefore carries no explicit information about the mapping of $f : x \mapsto y$, implicit information contained in the structure of the data becomes even more important. The GMM ([17], Eq. 6) and RWM ([17], Eq. 7) distance measures (denoted as Δ_{GMM} and Δ_{RWM}) are based on Gaussian mixture models (GMMs). A GMM models data with a number of J multivariate Gaussian distributions[3]. To speed up the training process, first k-means clustering is performed to find J clusters in the data. Based on the samples belonging to the clusters, the initial means and variances are computed to initialize the Gaussian distributions. Then the components are refined, either with the *Expectation Maximization* (EM) or *Variational Inference* (VI) training method [1], using only the feature vectors x of the samples. The GMMs used in this paper are trained with the VI method. The result of the training is a GMM with J components, component weights ϕ_j[4] that determine the influence of the component in the mixture, as well as the component covariance matrices Σ_j. Building on such a mixture model, the GMM [1] and RWM distance measures [17] consist of the Mahalanobis distance of two instances a and b with respect to the covariance matrices of the mixture model, weighted in two different ways: (1) the distance is weighted by the mixture coefficients of the model (GMM-distance, Eq. 1); (2) the distance is weighted by half the sum of the components responsibilities for the two instances (RWM-distance, Eq. 2). Both measures include the information captured by the GMM into the distance measure. The resulting distance is small, if both instances lie closest to the same GMM component, and large, if their closest GMM components differ. These distance measures are incorporated into kernel functions by substituting the Euclidean distance in the Gaussian RBF kernel with the GMM or RWM distance respectively [17]. The kernel functions thereby keep the parameter γ of the RBF kernel. These kernel functions can be used in kernel-based learning methods like SVMs or Parzen-Window kernel density estimation.

$$\Delta_{GMM}(a,b) = \sum_{j=1}^{J} \phi_j \Delta_{\Sigma_j}(a,b)) \tag{1}$$

$$\Delta_{RWM}(a,b) = \sum_{j=1}^{J} (\frac{1}{2}(p(j|a) + p(j|b))\Delta_{\Sigma_j}(a,b)) \tag{2}$$

The research question of this work is whether the inclusion of structural information by means of such kernels improves the performance of the McPAL approach. To examine that this work investigates two possible ways the McPAL strategy can employ such kernels and to which extent these benefit the strategy. The first approach of using these structure-sensitive kernel functions in combination with

[3] These distributions are referred to as 'components'

[4] As part of the VI training method, the weights of some components can be set close to zero, effectively 'pruning' them from the model

the McPAL strategy is incorporating them into the process of instance selection. To this end, two changes to the method are made that are described in the following.

First, the GMM/RWM kernels replace the Gaussian RBF kernel in the computation of the kernel frequency estimates (denoted as $\overrightarrow{k}$ in Eq. 4), which are required by the McPAL method, by means of the Parzen-Window method. These frequency estimates are computed in the same way as in the original McPAL approach [10], i.e. by computing the kernel density estimates with the Parzen-Window method, but leaving out the normalization by the number of samples.

$$k_{x,y} = \sum_{\{(x',y');y'=y\}} K_{GMM/RWM}(x,x') \tag{3}$$

$$\overrightarrow{k}_x = \{k_{x,y_1}, k_{x,y_2}, \cdots, k_{x,y_n}\} \tag{4}$$

Second, instead of using Parzen-Window estimation, the density estimates are directly taken from the GMM used by the GMM and RWM kernels. The Parzen-Window method places a kernel K with bandwidth h on each of the N samples in the data set, with each of them equally contributing to the resulting density estimate (s. Eq. 5). The GMM uses a fixed number of J multivariate Gaussians $\mathcal{N}$ to model the data, the contribution of each of these components being weighted by the mixture coefficient or component weight ϕ of the component (s. Eq. 6). These changes enable the McPAL strategy to use the information provided by the GMM and RWM kernels in the instance selection process. For the purpose of disambiguation, this modified version of the McPAL strategy is in the following referred to as *StrucPAL*.

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} K_h(x - x_i) = \frac{1}{Nh}\sum_{n=1}^{N} K(\frac{x - x_n}{h}) \tag{5}$$

$$p(x) = \sum_{j=1}^{J} \phi_j \mathcal{N}(x|\mu_j, \Sigma_j) \tag{6}$$

The second approach to use structure-sensitive kernels to improve the performance of the McPAL strategy is by using them in the consumer classifier. This is possible in two ways, either as *'self-selection'* or *'foreign-selection'*. Tomanek & Morik [19] use these terms to refer to, in the first case, the selector and consumer classifiers being the same, or in the second case, the selector and consumer classifiers being different. Therefore, two scenarios for using the GMM and RWM kernels in the classification process are possible. The first is the StrucPAL method being used with self-selection, so pwc_{rwm} or pwc_{gmm} act as both selector and consumer classifier respectively. The second is that the McPAL or StrucPAL strategy is employed for instance selection but classification is performed by a foreign classifier which uses the GMM or RWM kernels - i.e. Parzen-Window

classifier or a SVM.
This work aims to investigate two questions regarding the use of structural information by the McPAL method, in order to gain additional insight into what approaches are worth exploring in future research.
The first question is whether, and to what extent, the performance of the StrucPAL method differs from the original McPAL method. Due to the already mentioned inclusion of the information of the underlying mixture model into the instance selection process, a positive impact on the performance is expected.
The second question is if and to what extent the McPAL and StrucPAL learners benefit from foreign-selection. The first part of this question is to investigate, how the performance of McPAL and StrucPAL learners using self-selection compares to using a SVM with the same kernel as the selector as consumer classifier. The second part of this question is to investigate, how the original McPAL strategy can benefit from consumer classifiers that use the GMM or RWM kernels.

4 Experiments

In our experiments, eight data sets from the UCI machine learning repository [14] are used, namely australian, glass, haberman, heart, qsar-biodeg[5], steel-plates-fault[6], vehicle and vertebral. Furthermore, the phoneme data set from OpenML [21] is used. In addition to that, an artificial 2d data set referred to as blobs is used, consisting of three Gaussians that make up the classes.
The experiments include three AL strategies: *McPAL* (mp), *StrucPAL* (sp) and random sampling (rl). As classifiers, the Parzen-Window classifier (pwc) and support vector machine classifier (svm) are used. The PWC and SVM classifiers use either the Gaussian RBF, the RWM or the GMM kernels, as introduced earlier. The kernel that the classifier uses is denoted in subscript, for example pwc_{rbf}. An active learner in the experiments consists of three components: the AL strategy (al), the selector classifier (cl_{al}) and the consumer classifier (cl). In the case of self-selection, cl and cl_{al} are identical.
Based on a set of 10 seeds for randomization, for each seed, the data sets are split using five-fold stratified cross-validation. One fold per split is used as holdout set to test the trained consumer classifier, while the four remaining folds are used as training data. The initial labeled set $\mathcal{L}$ is initialized with one instance from two randomly picked classes in the training data. The random choice is based on the seed used in the current cross-validation split. Starting with $\mathcal{L}$ initialized with 2 instances and the rest of the training set comprising the unlabeled set $\mathcal{U}$, pool-based AL is performed. As part of this, the labels of 60 instances in total are acquired in a one-by-one fashion with both the selector and consumer classifier being updated after each acquisition. Then the consumer classifier is evaluated on the holdout set using the accuracy metric. This process is repeated until every fold has been used as test set once. The performance scores at every point in the AL process are averaged over all folds. After each of the 10 seeds

[5] in the following abbreviated as 'qsar'
[6] in the following abbreviated as 'steel'

has been used to seed the cross-validation split, the final results are gained by computing the average accuracy per step in the AL process and the standard deviation of the accuracy over all seeds.
The hyperparameters of the models for each data set are determined by performing an exhaustive search over a parameter grid on a subset of the data. This subset is a stratified, seed-based[7] random subsample consisting of 90 instances. The 90 instances are split using three-fold stratified cross-validation, with one fold being used to train a classifier with a given set of parameters, while the other two folds are used to evaluate the performance of this classifier. The small size of this tuning data set is founded in the fact, that in AL applications there is little labeled data available, therefore performing model selection with a large tuning set would be unrealistic. However, a review of the literature on active and semi-supervised learning did not provide a fitting way to determine the hyperparameters without using more labeled data than would be available in this scenario.

5 Results & Discussion

In the following, the results for the scenarios of self-selection and foreign-selection are presented in two ways.
First, the average accuracy scores and the corresponding standard deviation is tabulated for the different active learners on each data set. The highest accuracy score on a data set is printed in bold font. In the case of learners scoring equally, a lower standard deviation decides the winner. In case these are also identical, the first place is shared by these. For each learner, the difference in accuracy score to the highest score on each data set is computed and averaged for all data sets. This average difference in accuracy to the winners is shown in the column '*diff*' . Based on this difference, the learners are ranked, shown in column '*rank*'.
The second way of illustrating the results is so called *learning curve plots*. These show the performance of the learners on a given data set over the entire AL process, that is for each acquired label.

5.1 Self-Selection

First, the results of the experiments for the scenario of self-selection, shown in Tab. 1, will be considered. In the three moments in the learning process at 10, 20 and 30 acquired labels a good performance of the original McPAL method can be observed. It performs best on 6 of 10 data sets at 10 sampled instances, scoring the first rank in the comparison to the two StrucPAL variants and random selection learners with the Parzen-Window classifier with the RBF, GMM and RWM kernels. At 20 sampled instances, McPAL performs best on 5 data sets, scoring second rank and at 30 sampled instances it is best on 8 data

[7] The seed used for the model selection was not used in the splits for the experiments themselves.

sets, scoring first rank again. Based on these observations a solid performance can be attested to the McPAL strategy, although it does not manage to perform best on the steel and vehicle, where it is beaten by random selection with only one exception (vehicle, 10 sampled instances). The StrucPAL method only manages to perform better than McPAL on the blobs data set at 10 and 20 sampled instances as well as on heart at 20 sampled instances, although Fig. 1 shows an overall better performance of StrucPAL on heart. The gap between the scores of StrucPAL to the best performing learner on each data set varies in size, but when averaged leads to the two StrucPAL learners taking the last two ranks in the ranking.
Concluding the results of the self-selection scenario, the StrucPAL method did not provide better classification performance than the original method. Based on this observation it appears, that including the structural information from the Gaussian mixture model in the selection process did not improve the McPAL method.

Table 1. Mean accuracy scores and std (in brackets) after acquiring 10,20 and 30 labeled instances with Parzen Window Classifier (PWC), using self-selection or random-selection. Abbreviations are explained in Sec. 4.

10 labels sampled	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	.64(±0.08)	.81(±0.09)	**.66**(±0.07)	**.69**(±0.08)	**.77**(±0.06)	**.74**(±0.03)	.62(±0.09)	.55(±0.06)	**.41**(±0.03)	**.61**(±0.07)	.032	1
$pwc_{gmm} + sp$	.64(±0.09)	**.87**(±0.04)	.56(±0.07)	.68(±0.07)	.67(±0.1)	.49(±0.12)	.60(±0.11)	.57(±0.07)	.36(±0.05)	.54(±0.08)	.084	6
$pwc_{rwm} + sp$	.63(±0.11)	**.87**(±0.04)	.54(±0.07)	.68(±0.06)	.76(±0.08)	.62(±0.12)	.57(±0.1)	.55(±0.08)	.38(±0.03)	.53(±0.08)	.069	5
$pwc_{rbf} + rl$	**.69**(±0.08)	.78(±0.11)	.62(±0.08)	.65(±0.08)	.67(±0.1)	.69(±0.05)	**.65**(±0.07)	.68(±0.06)	.39(±0.05)	.60(±0.08)	.040	2
$pwc_{gmm} + rl$	.67(±0.09)	.77(±0.1)	.58(±0.1)	.64(±0.06)	.66(±0.09)	.68(±0.05)	.62(±0.06)	.70(±0.07)	.41(±0.06)	.58(±0.08)	.051	3
$pwc_{rwm} + rl$	.66(±0.1)	.77(±0.1)	.54(±0.09)	.64(±0.06)	.68(±0.11)	.71(±0.04)	.60(±0.07)	**.69**(±0.07)	.39(±0.05)	.60(±0.07)	.054	4
20 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	.68(±0.08)	.88(±0.05)	**.72**(±0.05)	**.73**(±0.05)	.75(±0.06)	**.74**(±0.03)	**.72**(±0.04)	.60(±0.05)	.46(±0.03)	**.67**(±0.06)	.033	2
$pwc_{gmm} + sp$	.66(±0.09)	**.90**(±0.02)	.58(±0.05)	.69(±0.04)	.74(±0.08)	.56(±0.11)	.54(±0.1)	.57(±0.07)	.43(±0.04)	.62(±0.06)	.099	6
$pwc_{rwm} + sp$	.64(±0.09)	**.90**(±0.01)	.54(±0.06)	.71(±0.04)	**.78**(±0.07)	.68(±0.09)	.51(±0.11)	.55(±0.08)	.43(±0.03)	.61(±0.07)	.093	5
$pwc_{rbf} + rl$	**.73**(±0.05)	.86(±0.04)	.70(±0.07)	.71(±0.04)	.73(±0.07)	.72(±0.03)	.69(±0.05)	.74(±0.04)	.48(±0.05)	.65(±0.06)	.027	1
$pwc_{gmm} + rl$	.72(±0.05)	.86(±0.04)	.65(±0.07)	.66(±0.06)	.70(±0.07)	.72(±0.04)	.64(±0.06)	**.78**(±0.04)	**.51**(±0.06)	.63(±0.06)	.041	3
$pwc_{rwm} + rl$	.71(±0.07)	.86(±0.04)	.63(±0.08)	.66(±0.06)	.73(±0.1)	.73(±0.03)	.63(±0.06)	.75(±0.04)	.45(±0.05)	.64(±0.06)	.049	4
30 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.76**(±0.04)	**.90**(±0.01)	**.76**(±0.04)	**.74**(±0.03)	**.79**(±0.04)	**.75**(±0.02)	**.74**(±0.04)	.69(±0.03)	.51(±0.02)	**.69**(±0.05)	.020	1
$pwc_{gmm} + sp$	.69(±0.07)	**.90**(±0.02)	.58(±0.05)	.70(±0.03)	.75(±0.08)	.63(±0.09)	.59(±0.08)	.57(±0.07)	.46(±0.03)	.66(±0.05)	.100	5
$pwc_{rwm} + sp$	.65(±0.08)	.89(±0.02)	.56(±0.07)	.70(±0.04)	.78(±0.07)	.70(±0.06)	.57(±0.09)	.55(±0.08)	.46(±0.03)	.65(±0.07)	.102	6
$pwc_{rbf} + rl$	.75(±0.04)	.88(±0.03)	.74(±0.06)	.72(±0.04)	.75(±0.06)	.74(±0.02)	.71(±0.05)	.77(±0.04)	.52(±0.05)	.68(±0.05)	.027	2
$pwc_{gmm} + rl$	.73(±0.06)	.88(±0.03)	.70(±0.07)	.67(±0.05)	.72(±0.08)	.73(±0.03)	.67(±0.05)	**.82**(±0.04)	**.57**(±0.06)	.67(±0.05)	.037	3
$pwc_{rwm} + rl$	.73(±0.06)	.88(±0.03)	.67(±0.07)	.67(±0.05)	.75(±0.09)	.73(±0.02)	.65(±0.06)	.78(±0.04)	.48(±0.06)	.67(±0.05)	.052	4

5.2 Foreign-Selection

How does foreign-selection affect the result of the active learner? Tab. 2 shows the accuracy scores at the stages of 10, 20 and 30 sampled instances. For every AL strategy, self-selection is compared to the use of an SVM (with the same kernel as the selector) as consumer classifier.
As originally pointed out by Tomanek and Morik [19], it can be observed that foreign-selection can be indeed beneficial with regard to classification performance. However, the extent of this varies in the experiments, ranging from a difference in accuracy of 0.01 to 0.08 and is limited to some of the data sets. Based on the averaged difference in score to the best performing method, self-selection scores better than foreign selection in all three segments. This analysis, however, only included a consumer classifier (SVM), that uses the same kernel function as the selector. In order to investigate, how McPAL learners perform, if

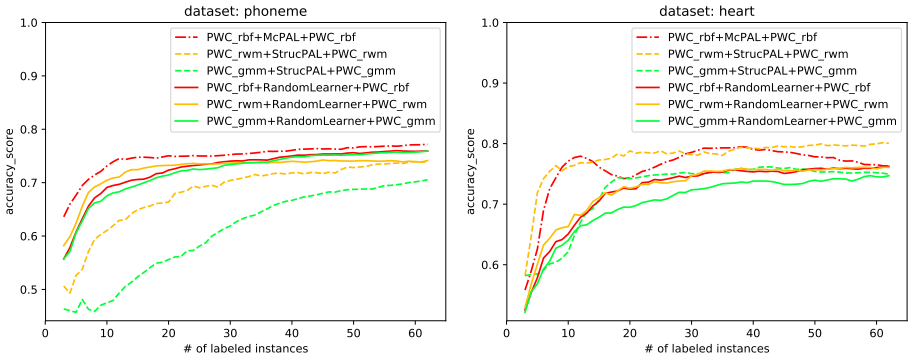

Fig. 1. Learning curves resulting from self-selection on the phoneme and heart data sets

paired with consumer classifiers using the GMM and RWM kernels, a separate tabulation is shown in Tab. 3.

Although the self-selection learner with McPAL+pwc_{rbf} scores the first rank in all three stages, it can be observed that McPAL can benefit from different consumer classifiers. At the stage of 10 sampled instances, an SVM with GMM kernel scores a higher accuracy on the steel (+0.1) and vehicle (+0.07) data sets, with minor gains being provided by an SVM with RBF kernel (+0.01 on qsar, +0.02 on glass) and a SVM with RWM kernel (+0.02 on haberman). However, these gains are accompanied by worse performance than McPAL on other data sets. The advantage provided by the foreign classifiers reduces in the stages of 20 and 30 sampled instances, with svm_{gmm} still showing good gains at 20 sampled instances (+0.05 on steel, +0.1 on vehicle).

Fig. 2 shows the learning curves on the vehicle and steel data sets. While on vehicle a solid advantage of svm_{gmm}, svm_{rbf} and pwc_{gmm} over McPAL in terms of accuracy can be observed, the development on steel is a different one. While the svm_{gmm} and svm_{rwm} learners perform well due to a stagnating but better performance in the early phase, they fail to exploit the additionally acquired labels in the fashion of the other learners, resulting in a slight but increasing advantage for the learners using GMM-based PWCs later, which are in the last phase of the learning process surpassed by svm_{rbf}.

Concluding the results of the foreign-selection scenario it can be summarized, that although self-selection McPAL has performed solidly in the experiments, the results indicate that the use of classifiers with GMM-based kernels in this scenario shows potential and the general use of foreign-selection motivates further research.

Table 2. Mean accuracy scores and std (in brackets) of McPAL and StrucPAL learners using either self-selection or a SVM with the same kernel as the selector for classification.

10 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.64**(±0.08)	**.81**(±0.09)	.66(±0.07)	.69(±0.08)	**.77**(±0.06)	**.74**(±0.03)	.62(±0.09)	.55(±0.06)	.41(±0.03)	**.61**(±0.07)	.006	1
$svm_{rbf} + mp + pwc_{rbf}$	.60(±0.08)	.80(±0.09)	**.68**(±0.07)	**.69**(±0.07)	.62(±0.09)	.71(±0.04)	**.63**(±0.09)	**.56**(±0.06)	**.43**(±0.04)	.57(±0.12)	.027	2
$pwc_{gmm} + sp$	**.64**(±0.09)	**.87**(±0.04)	**.56**(±0.07)	**.68**(±0.07)	**.67**(±0.1)	.49(±0.12)	**.60**(±0.11)	.57(±0.07)	**.36**(±0.05)	**.54**(±0.08)	.009	1
$svm_{gmm} + sp + pwc_{gmm}$	.59(±0.06)	.85(±0.05)	.54(±0.08)	.63(±0.06)	.61(±0.08)	**.51**(±0.12)	.57(±0.13)	**.64**(±0.01)	**.36**(±0.05)	.43(±0.12)	.034	2
$pwc_{rwm} + sp$	.63(±0.11)	**.87**(±0.04)	**.54**(±0.07)	.68(±0.06)	**.76**(±0.08)	**.62**(±0.12)	.57(±0.1)	.55(±0.08)	**.38**(±0.03)	**.53**(±0.08)	.016	1
$svm_{rwm} + sp + pwc_{rwm}$	.61(±0.11)	.84(±0.05)	.53(±0.08)	**.73**(±0.02)	.75(±0.07)	.58(±0.11)	**.60**(±0.11)	**.63**(±0.05)	.37(±0.05)	.49(±0.13)	.016	1
20 labels sampled	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.68**(±0.08)	**.88**(±0.05)	.72(±0.05)	**.73**(±0.05)	**.75**(±0.06)	**.74**(±0.03)	**.72**(±0.04)	**.60**(±0.05)	.46(±0.03)	**.67**(±0.06)	.007	1
$svm_{rbf} + mp + pwc_{rbf}$	.66(±0.09)	.87(±0.04)	**.74**(±0.05)	.72(±0.04)	.66(±0.1)	.72(±0.03)	.72(±0.06)	**.60**(±0.05)	**.51**(±0.04)	.64(±0.07)	.018	2
$pwc_{gmm} + sp$	**.66**(±0.09)	**.90**(±0.02)	**.58**(±0.05)	**.69**(±0.04)	**.74**(±0.08)	**.56**(±0.11)	**.54**(±0.1)	.57(±0.07)	.43(±0.04)	**.62**(±0.06)	.009	1
$svm_{gmm} + sp + pwc_{gmm}$	.63(±0.08)	.88(±0.02)	.57(±0.06)	.67(±0.04)	.70(±0.09)	**.56**(±0.11)	.45(±0.13)	**.64**(±0.01)	**.45**(±0.05)	.59(±0.09)	.024	2
$pwc_{rwm} + sp$	**.64**(±0.09)	**.90**(±0.01)	**.54**(±0.06)	.71(±0.04)	**.78**(±0.07)	**.68**(±0.09)	**.51**(±0.11)	.55(±0.08)	**.43**(±0.03)	.61(±0.07)	.011	1
$svm_{rwm} + sp + pwc_{rwm}$	.62(±0.1)	.88(±0.02)	.54(±0.08)	**.73**(±0.02)	.75(±0.09)	.62(±0.09)	.50(±0.13)	**.63**(±0.05)	.41(±0.05)	**.62**(±0.07)	.016	2
30 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.76**(±0.04)	**.90**(±0.01)	**.76**(±0.04)	**.74**(±0.03)	**.79**(±0.04)	**.75**(±0.02)	.74(±0.04)	**.69**(±0.03)	.51(±0.02)	**.69**(±0.05)	.008	1
$svm_{rbf} + mp + pwc_{rbf}$	.74(±0.07)	**.90**(±0.01)	.76(±0.05)	.73(±0.04)	.79(±0.06)	.72(±0.02)	**.75**(±0.05)	.67(±0.04)	**.58**(±0.04)	.65(±0.04)	.012	2
$pwc_{gmm} + sp$	**.69**(±0.07)	**.90**(±0.02)	**.58**(±0.05)	**.70**(±0.03)	**.75**(±0.08)	**.63**(±0.09)	**.59**(±0.08)	.57(±0.07)	.46(±0.03)	**.66**(±0.05)	.01	1
$svm_{gmm} + sp + pwc_{gmm}$	.66(±0.06)	**.90**(±0.02)	.57(±0.06)	.69(±0.03)	.72(±0.09)	.61(±0.09)	.49(±0.15)	**.65**(±0.02)	**.48**(±0.04)	.63(±0.05)	.023	2
pwc_{rwm}+sp	**.65**(±0.08)	**.89**(±0.02)	**.56**(±0.07)	.70(±0.04)	**.78**(±0.07)	**.70**(±0.06)	**.57**(±0.09)	.55(±0.08)	**.46**(±0.03)	.65(±0.07)	.012	1
$svm_{rwm} + sp + pwc_{rwm}$	.63(±0.08)	**.89**(±0.02)	.54(±0.08)	**.73**(±0.03)	.77(±0.07)	.63(±0.09)	.53(±0.13)	**.63**(±0.05)	.40(±0.05)	**.66**(±0.06)	.022	2

Table 3. Results with the McPAL strategy with PWC and SVM consumer classifiers with different kernels

10 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.64**(±0.08)	**.81**(±0.09)	.66(±0.07)	.69(±0.08)	**.77**(±0.06)	**.74**(±0.03)	.62(±0.09)	.55(±0.06)	.41(±0.03)	**.61**(±0.07)	.021	1
$pwc_{gmm} + mp$	.63(±0.09)	.79(±0.1)	.60(±0.07)	.68(±0.07)	.75(±0.07)	.73(±0.04)	.58(±0.08)	.57(±0.07)	.44(±0.04)	**.61**(±0.07)	.033	2
$pwc_{rwm} + mp$	.63(±0.09)	.79(±0.1)	.59(±0.08)	.67(±0.08)	.74(±0.09)	.73(±0.03)	.55(±0.09)	.56(±0.08)	.40(±0.03)	**.61**(±0.07)	.044	4
$svm_{rbf} + mp$	.60(±0.08)	.80(±0.09)	**.68**(±0.07)	.69(±0.07)	.62(±0.09)	.71(±0.04)	**.63**(±0.09)	.56(±0.06)	.43(±0.04)	.57(±0.12)	.042	3
$svm_{gmm} + mp$	.60(±0.08)	.79(±0.1)	.61(±0.09)	.66(±0.07)	.69(±0.08)	.69(±0.04)	.53(±0.13)	**.65**(±0.0)	**.47**(±0.05)	.54(±0.12)	.048	5
$svm_{rwm} + mp$	.62(±0.11)	.78(±0.1)	.61(±0.09)	**.71**(±0.07)	.69(±0.09)	.71(±0.04)	.52(±0.13)	.63(±0.05)	.38(±0.05)	.57(±0.09)	.049	6
20 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.68**(±0.08)	**.88**(±0.05)	.72(±0.05)	**.73**(±0.05)	**.75**(±0.06)	**.74**(±0.03)	**.72**(±0.04)	.60(±0.05)	.46(±0.03)	**.67**(±0.06)	.017	1
$pwc_{gmm} + mp$	.68(±0.09)	.87(±0.05)	.68(±0.07)	.70(±0.05)	.73(±0.07)	**.74**(±0.03)	.66(±0.06)	.61(±0.08)	.50(±0.04)	.66(±0.06)	.029	3
$pwc_{rwm} + mp$	.68(±0.09)	.87(±0.05)	.65(±0.07)	.70(±0.06)	.72(±0.09)	.73(±0.02)	.64(±0.07)	.63(±0.05)	.45(±0.03)	.66(±0.06)	.039	5
$svm_{rbf} + mp$	.66(±0.09)	.87(±0.04)	**.74**(±0.05)	.72(±0.04)	.66(±0.1)	.72(±0.03)	.72(±0.06)	.60(±0.05)	.51(±0.04)	.64(±0.07)	.028	2
$svm_{gmm} + mp$	.65(±0.1)	.87(±0.04)	.69(±0.08)	.69(±0.06)	.71(±0.08)	.71(±0.04)	.63(±0.09)	**.65**(±0.0)	**.56**(±0.04)	.61(±0.07)	.035	4
$svm_{rwm} + mp$	.68(±0.09)	.87(±0.04)	.65(±0.06)	.71(±0.04)	.71(±0.09)	.72(±0.03)	.62(±0.09)	**.65**(±0.0)	.38(±0.05)	.65(±0.06)	.048	6
30 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.76**(±0.04)	**.90**(±0.01)	**.76**(±0.04)	**.74**(±0.03)	**.79**(±0.04)	**.75**(±0.02)	.74(±0.04)	**.69**(±0.03)	.51(±0.02)	**.69**(±0.05)	.010	1
$pwc_{gmm} + mp$	.75(±0.04)	**.90**(±0.01)	.74(±0.06)	.71(±0.05)	.75(±0.06)	**.75**(±0.02)	.67(±0.06)	.69(±0.04)	.55(±0.03)	.68(±0.05)	.024	3
$pwc_{rwm} + mp$	.75(±0.04)	.89(±0.01)	.69(±0.07)	.70(±0.05)	.75(±0.08)	.74(±0.02)	.66(±0.07)	.69(±0.04)	.49(±0.03)	.68(±0.04)	.039	4
$svm_{rbf} + mp$	.74(±0.07)	**.90**(±0.01)	.76(±0.05)	.73(±0.04)	**.79**(±0.06)	.72(±0.02)	**.75**(±0.05)	.67(±0.04)	.58(±0.04)	.65(±0.04)	.014	2
$svm_{gmm} + mp$	.72(±0.07)	**.90**(±0.01)	.74(±0.07)	.70(±0.05)	.75(±0.07)	.71(±0.05)	.63(±0.08)	.65(±0.0)	**.6**(±0.04)	.63(±0.05)	.040	5
$svm_{rwm} + mp$	.74(±0.06)	.89(±0.02)	.64(±0.07)	.71(±0.04)	.75(±0.08)	.72(±0.03)	.61(±0.09)	.65(±0.0)	.39(±0.05)	.67(±0.05)	.066	6

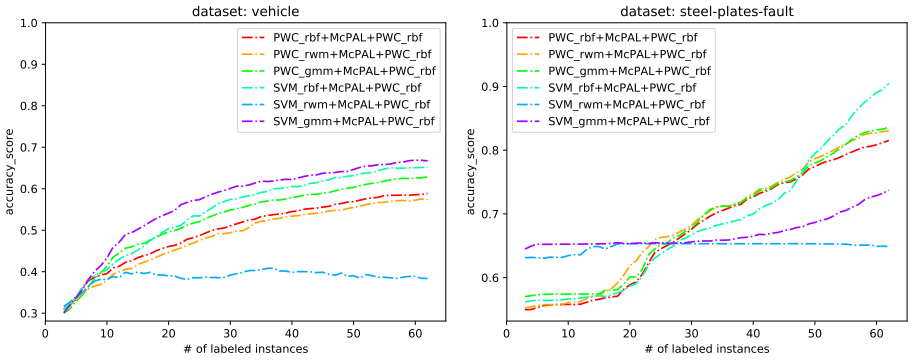

Fig. 2. Learning curves of McPAL learners using different consumer classifiers on the vehicle and steel data sets

6 Conclusion

The experiments explored two possible approaches to incorporate the information of a GMM into the McPAL method. The first approach, using two GMM-based kernel functions in the instance selection process, has shown to not provide an advantage regarding the performance compared to the original method. In total, the original McPAL selection strategy with pwc_{rbf} both as selector and consumer classifier, has shown to perform better than the StrucPAL learners, with random sampling performing better than both methods in few cases. Especially data sets like australian, glass, vehicle and vertebral proofed harder for the StrucPAL learners. One possible explanation for this is that the assumption of the GMM, i.e. that the subpopulations in the data representing the different classes fit a multivariate Gaussian distribution, does not hold in these cases.
The second approach, using the GMM-based kernel functions in the consumer classifiers of a foreign-selection scenario, showed potential gains regarding classification accuracy. Using a svm_{gmm} as consumer classifier for the original McPAL learner has shown to improve the classification performance on the steel and vehicle data sets while performing slightly worse on others. The fact, that learners using the StrucPAL method and PWC with the GMM or RWM kernel generally did not benefit from using an SVM with these kernels proves interesting. It appears that either the use of the same kernel function did not mitigate the adverse effect foreign-selection seems to entail in this case, or that the labeled set sampled by StrucPAL is simply less fit for classification with svm_{gmm} or svm_{rwm}.
Regarding the performance of the SVM classifiers used in the experiments, it has to be considered that the model selection procedure employed in the experiments is admittedly weak. Therefore, it is possible that the hyperparameters used in the experiments, not only for the SVMs but also the other classifiers, are suboptimal. Considering the restrictive nature of the AL setting regarding the availability of labeled data, this circumstance is acceptable, since using more data for model selection would be even more unrealistic in this setting.
It appears that the McPAL strategy already performs very well at selecting the most useful instances and including the information of the GMM does not add to this, in some cases even hindering a good selection. Based on these results it appears that work on the McPAL strategy in the future should focus on improving the method regarding other aspects, for example imbalanced data.
However, using other classifiers to exploit the labeled set sampled with the McPAL strategy has shown to be of possible gain, in order to improve the overall classification performance of the active learner. The use of SVMs as consumer classifiers showed to have potential, although determining fitting hyperparameters in the setting of active learning still poses a problem, that should be investigated further.

Bibliography

[1] C Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York*, 2007.

[2] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *ICML*, volume 3, pages 59–66, 2003.

[3] Nicolas Cebron and Michael R Berthold. Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*, 18(2):283–299, 2009.

[4] Gang Chen, Tian-jiang Wang, Li-yu Gong, and Perfecto Herrera. Multi-class support vector machine active learning for music annotation. *International Journal of Innovative Computing, Information and Control*, 6(3):921–930, 2010.

[5] Charlie K Dagli, Shyamsundar Rajaram, and Thomas S Huang. Utilizing information theoretic diversity for svm active learn. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 506–511. IEEE, 2006.

[6] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[7] Dezhi Hong, Hongning Wang, and Kamin Whitehouse. Clustering-based active learning on sensor type classification in buildings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 363–372. ACM, 2015.

[8] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2372–2379. IEEE, 2009.

[9] Ajay J Joshi, Fatih Porikli, and Nikolaos P Papanikolopoulos. Scalable active learning for multiclass image classification. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2259–2273, 2012.

[10] Daniel Kottke, Georg Krempl, Dominik Lang, Johannes Teschner, and Myra Spiliopoulou. *Multi-Class Probabilistic Active Learning*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 586 – 594. IOS Press, 2016.

[11] Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

[12] Dominik Lang, Daniel Kottke, Georg Krempl, and Myra Spiliopoulou. Investigating exploratory capabilities of uncertainty sampling using svms in active learning. In *Active Learning: Applications, Foundations and Emerging Trends*, 2016.

[13] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[14] M. Lichman. UCI machine learning repository, 2013.

[15] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[16] Tobias Reitmaier and Bernhard Sick. Active classifier training with the 3ds strategy. In *Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on*, pages 88–95. IEEE, 2011.

[17] Tobias Reitmaier and Bernhard Sick. The responsibility weighted mahalanobis kernel for semi-supervised training of support vector machines for classification. *Information Sciences*, 323:179–198, 2015.

[18] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[19] Katrin Tomanek and Katharina Morik. Inspecting sample reusability for active learning. In Isabelle Guyon, Gavin C. Cawley, Gideon Dror, Vincent Lemaire, and Alexander R. Statnikov, editors, *AISTATS workshop on Active Learning and Experimental Design*, volume 16 of *JMLR Proceedings*, pages 169–181. JMLR.org, 2011.

[20] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[21] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.