Improving Gaussian Process Classification with Outlier Detection, with Applications in Image Classification

Yan Gao and Yiqun Li

Institute for Infocomm Research
Agency for Science, Technology and Research (A*STAR), Singapore

Abstract. In many computer vision applications for recognition or classification, outlier detection plays an important role as it affects the accuracy and reliability of the result. We propose a novel approach for outlier detection using Gaussian process classification. With this approach, the outlier detection can be integrated to the classification process, instead of being treated separately. Experimental results on handwritten digit image recognition and vision based robot localization show that our approach performs better than other state of the art approaches.

1 Introduction

Outliers refer to the data which do not fall into any learned classes in a classification system. Outlier detection is the identification of the unknown data or signal that a classification system is not aware of during training [1]. It is also usually referred to as novelty detection or abnormality detection. It is a common issue encountered in many computer vision applications, such as in robot vision [2], face recognition [3], and other image classification applications [4, 5]. Machine learning is a popular methodology for image classification. Using machine learning methods, outlier detection is usually treated as a one-class learning problem. Treating the given training samples as the 'normal class', a pre-assumed model is used to describe the normal class. In the test phase, a sample is classified as 'normal' or 'abnormal' by comparing it to the model. To model the normal class, various approaches have been explored, including clustering [6], nearest neighbor [7], mixture models [8], neural networks [4], self organizing maps (SOM) [9], and one class support vector machines (SVM) [5, 10, 11].

As the above works focus on the 'one class' problem, i.e., only concern about whether a new sample is normal or abnormal, it cannot solve the multi-class classification problem directly. There are plenty of applications that require both classification of a test sample into the existing classes as well as detection of outliers. When the application requires a multi-class classification, it needs 2 classification processes. One is the classification of normal and abnormal, namely, the outlier detection. The other is the classification of different normal classes. The method proposed in this paper is able to solve the multi-class classification problem with outlier detection simultaneously in one classification process. This

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part IV, LNCS 6495, pp. 153–164, 2011.

is because the proposed outlier detection method is inherently part of the Gaussian process classification process. Besides the outlier detection, the classifier can also refrain from making a decision when the confidence level is low as indicated by the winning class's probability estimate. In other words, the classifier will reject the unreliable classification results so that the classification can be more reliable to reduce the potentially high cost of misclassifications.

Although a few papers have discussed the multi-class problem in outlier detection, the objectives and problem scope are quite different. Masud et. al. proposed an outlier class detector within a decision tree or k nearest neighbor classifier [12]. It is specifically targeted for classification of data stream with possible concept-drift. The outliers must have some degree of coherence in order to form a novel class. In [3], a multi-class classifier with outlier detection is formed by combining multiple one-class classifiers. Multiple thresholds can be tuned to each of the one-class classifiers to improve the performances. However, solving a multi-class classification problem using one-class models will decrease the discrimination capability because the between-class variations is not considered. Hempstalk and Frank use a multi-class classification approach to solve outlier detection problem by assuming that training samples from the new classes are available [13]. Different from the above, our proposed approach solves a generic multi-class classification problem with more reliable detection of outliers in the test data. It does not require sample data from the abnormal class for training. During testing, the classifier will classify a test sample into one of the training classes, or detect it as an outlier, or refrain from making a decision if not sure. Gaussian process classification (GPC) produces a probabilistic classifier which includes both prediction of probabilities of a sample belonging to the training classes, as well as a covariance matrix of the predicted probabilities. We make use of the covariance matrix for outlier detection. The proposed approach is evaluated on 2 benchmark datasets for handwritten digit recognition and robot localization and shows promising results.

2 Multi-class Gaussian Process Classification

We first give a brief introduction to multi-class Gaussian process classification. For more details, the readers are referred to [14]. For a multi-class problem, we are given at set of input vectors $X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)^T$ and a target vector $\mathbf{y} = (y_1^1, ..., y_n^1, y_2^2, ..., y_n^2, ..., y_1^C, ..., y_n^C)^T$, where n is the number of input vectors, C is the number of classes, $y_i^c = 1$ if the i^{th} input vector belongs to the c^{th} class, and it is all zero otherwise.

In order to make inference, a vector of latent function values $\mathbf{f}$ is introduced. $\mathbf{f} = (f_1^1, ..., f_n^1, f_2^2, ..., f_n^2, ..., f_1^C, ..., f_n^C)^T$. It is assumed that the C latent processes are uncorrelated. A prior over the latent function is specified. It follows a normal distribution with a mean of 0 and a covariance matrix K:

$$\mathbf{f} \sim \mathcal{N}(0, K) \tag{1}$$

The covariance matrix K is block diagonal with sub-matrices $K_1, ..., K_C$ on the diagonal. The covariance matrix for each of the C classes is defined by its own covariance function:

$$K_{c(i,j)} = k_c(\mathbf{x}_i, \mathbf{x}_j), \qquad i, j = 1, ..., n \tag{2}$$

The target vector $\mathbf{y}$ is related to the latent function vector $\mathbf{f}$ by:

$$p(y_i^c|\mathbf{f}_i) = \pi_i^c = \frac{\exp(f_i^c)}{\sum_{c'} \exp(f_i^{c'})} \tag{3}$$

where $\mathbf{f}_i = (f_i^1, ..., f_i^C)^T$.

The posterior $p(\mathbf{f}|X, \mathbf{y})$ is proportional to the joint probability $p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)$, based on the Baye's theorem. The log of the un-normalized posterior is shown to be:

$$\Phi(\mathbf{f}) = -\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} + \mathbf{y}^T\mathbf{f} - \sum_{i=1}^{n} \log(\sum_{c=1}^{C} \exp f_i^c) - \frac{1}{2}\log|K| - \frac{Cn}{2}\log 2\pi \tag{4}$$

As the posterior is not analytically tractable, the Laplace approximation is used to give a Gaussian approximation $q(\mathbf{f}|X, \mathbf{y})$ to the posterior $p(\mathbf{f}|X, \mathbf{y})$. To do this, a second order Taylor expansion of $\log p(\mathbf{f}|X, \mathbf{y})$ is needed around the maximum of the posterior. Denote the value that maximizes the posterior as $\hat{\mathbf{f}}$,

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \Phi(\mathbf{f}) \tag{5}$$

and it is found using the Newton's method.

To predict the label of a new input $\mathbf{x}_*$, the posterior distribution $q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*)$ is given by

$$q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*) = \int p(\mathbf{f}_*|X, \mathbf{x}_*, \mathbf{f})q(\mathbf{f}|X, \mathbf{y})d\mathbf{f} \tag{6}$$

Both $p(\mathbf{f}_*|X, \mathbf{x}_*, \mathbf{f})$ and $q(\mathbf{f}|X, \mathbf{y})d\mathbf{f}$ are Gaussian. Therefore $q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*)$ is also Gaussian. Its mean is given by

$$\mathbb{E}_q[\mathbf{f}(\mathbf{x}_*|X, \mathbf{y}, \mathbf{x}_*)] = Q_*^T K^{-1}\hat{\mathbf{f}} = Q_*^T(\mathbf{y} - \hat{\pi}) \tag{7}$$

where

$$Q_* = \begin{pmatrix} \mathbf{k}_1(\mathbf{x}_*) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_2(\mathbf{x}_*) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{k}_C(\mathbf{x}_*) \end{pmatrix} \tag{8}$$

where $\mathbf{k}_c(\mathbf{x}_*)$ is the vector of covariances between the test point and each of the training points, evaluated by class c's covariance function.

The covariance is given by

$$\begin{aligned} \mathrm{cov}_q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*) &= \Sigma + Q_*^T K^{-1}(K^{-1} + W)^{-1}K^{-1}Q_* \\ &= \mathrm{diag}(\mathbf{k}(\mathbf{x}_*, \mathbf{x}_*)) - Q_*^T(K + W^{-1})^{-1}Q_* \end{aligned} \tag{9}$$

where Σ is a diagonal $C \times C$ matrix with $\Sigma_{cc} = k_c(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_c^T(\mathbf{x}_*) K_c^{-1} \mathbf{k}_c(\mathbf{x}_*)$, and $\mathbf{k}(\mathbf{x}_*, \mathbf{x}_*)$ is a vector of covariances, whose c^{th} element is $k_c(\mathbf{x}_*, \mathbf{x}_*)$.

The marginal likelihood $\log p(\mathbf{y}|X, \theta)$ can be similarly approximated as

$$\begin{aligned} \log p(\mathbf{y}|X,\theta) &\simeq \log q(\mathbf{y}|X,\theta) \\ &= -\frac{1}{2}\hat{\mathbf{f}}^T K^{-1}\hat{\mathbf{f}} - \sum_{i=1}^{n} \log(\sum_{c=1}^{C} \exp \hat{f}_i^c) - \frac{1}{2}\log|I_{Cn} + W^{\frac{1}{2}} K W^{\frac{1}{2}}| \end{aligned} \tag{10}$$

The marginal likelihood can be used to tune the parameters of the covariance functions which are also known as the hyperparameters of the model.

3 Outlier Detection in Gaussian Process Classification

To detect outliers under the Gaussian process (GP) classification framework, the covariance in prediction plays an important role. Recall from the previous section that the prediction made by GP classification is characterized by a mean (Eq. (7)) and a covariance matrix (Eq. (9)). In Gaussian process, the variance in prediction is large when the new sample is out of the support of the training samples. (See e.g., illustrations in [15,16]) The total variance in the covariance matrix is an indicator of how familiar the classifier is about a particular test sample. We propose to use the determinant of the covariance matrix as the measure of novelty, and the rule is given by:

$$\begin{cases} l(\mathbf{x}_*) = -1 & \text{if } \det(\text{cov}_q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*)) > t \\ l(\mathbf{x}_*) = \arg\max_c p(y_*^c|\mathbf{f}_*) & \text{otherwise} \end{cases} \tag{11}$$

where $l(\mathbf{x}_*)$ refers to the label of the sample $\mathbf{x}_*$, and -1 is used as the label of the outliers. As in Eq. (3), $p(y_*^c|\mathbf{f}_*) = \frac{\exp(f_*^c)}{\sum_{c'} \exp(f_*^{c'})}$. Alternatively, we can sort the test data according to the novelty measure (which is the determinant of covariance matrix) in a descending order, and classify a certain amount of test samples with largest novelty measures as outliers, if such information is given by prior knowledge.

The choice of using the determinant of the covariance function as the novelty measure is not just heuristic. Recall that the determinant is equal to the sum of the eigenvalues of the covariance matrix. We know that the eigenvalues of the covariance matrix indicate the portions of variance that are explained by the principal components (see principal component analysis [17]). Therefore the sum of all eigenvalues reflects the total variance involved with the particular prediction. To illustrate the idea, a toy data is designed. It consists of three 2-d Gaussian clusters to form the three classes. A few mis-labeled samples are also simulated. We use the squared exponential covariance function,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\beta^2}) \tag{12}$$

The hyperparameters α and β are determined by optimizing the likelihood function in Eq. (10). In Fig. 1, we show a contour plot of the novelty measure

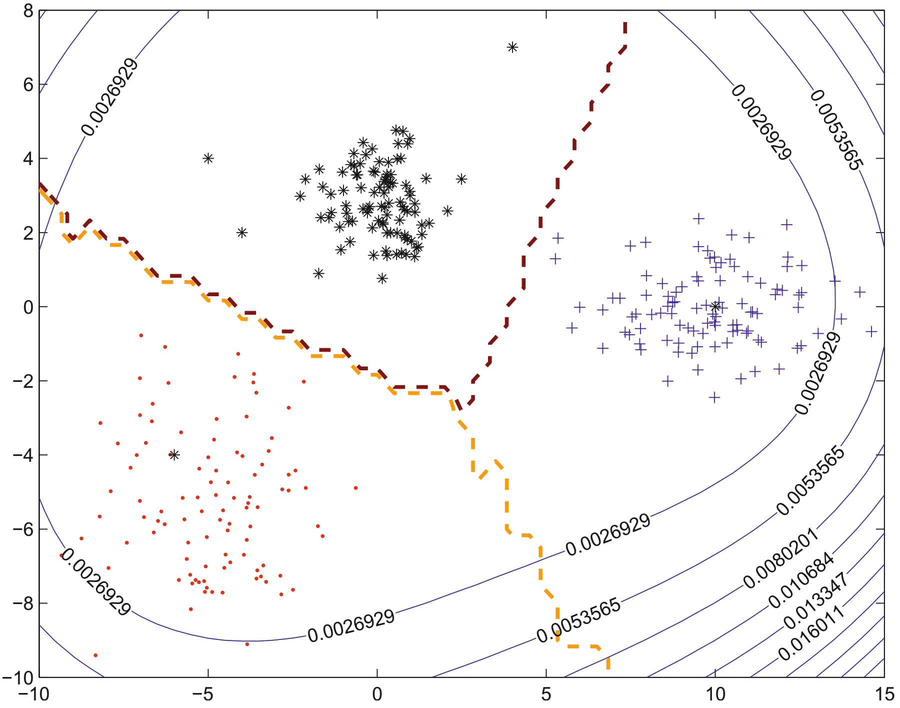


Fig. 1. Contour plot of the proposed novelty measure and partition of the input space into different classes

$\det(\mathrm{cov}_q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*))$ in the input space. It is observed that the proposed novelty measure gets larger when moving away from the cluster centers. In the meantime, the GP classifier also partitions the input space into the three training classes.

4 A Well-Rounded Classifier

With the proposed method for outlier detection, Gaussian process classification may offer advantages over other alternative classifiers to many real problems. Recall that GP classification produces a probabilistic prediction. The probability of a test sample belonging to all training classes are explicitly obtained by the classifier (Eq. (3)). As mentioned in [14], the probability of of the test sample belonging to the winning class can be used to reject unreliable predictions. If it is low,it shows that the classifier is not confident in classifying the test sample into a particular class. In this case, it might be advantageous to refrain from making a decision than making a wrong decision with high probability. This is known as the reject option in classification. To show how it works, we also plot the winning classes' probabilities for the three class classification problem in the previous section in Fig. 2. It is observed that the winning classes' probabilities

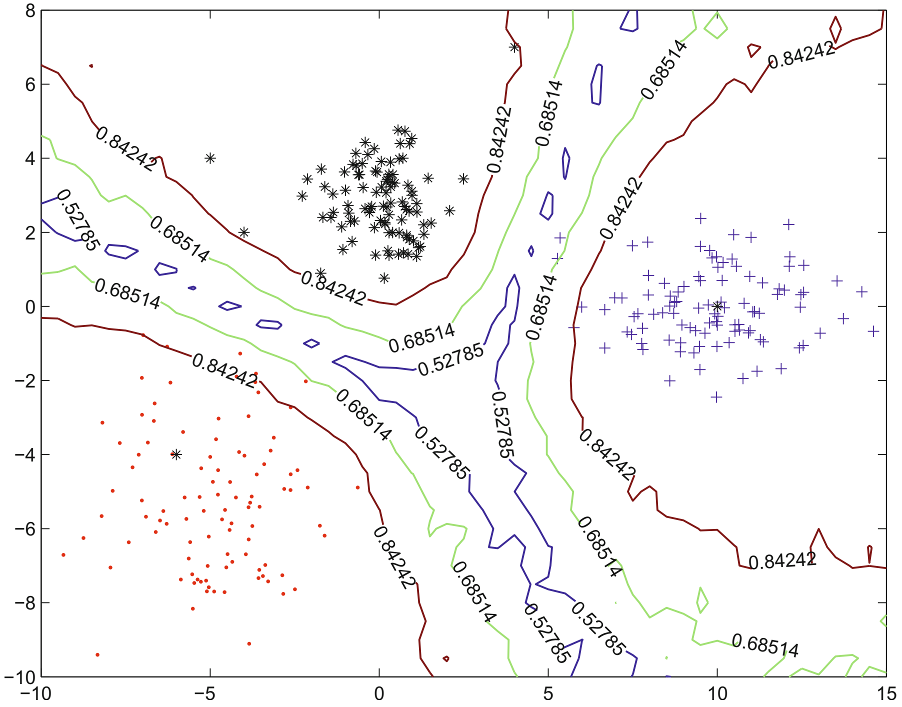


Fig. 2. Contour plot of the winning classes' probabilities

are smaller in between two training classes where it is most likely to make wrong predictions from a Bayesian point of view.

With the capabilities of detecting outliers and rejecting unreliable predictions, Gaussian process classification is well suited for some applications such as the robot localization application discussed in section 5.2.

5 Experiments

We implement the multi-class GP classification using Laplace approximation as outlined in section 2 and the outlier detection method in section 3 in Matlab. We compare the proposed outlier detection scheme with one class support vector machines which have been shown to be a state of the art for outlier detection, and have been popularly used in various applications [5,11,18,19,20]. The basic idea of one-class SVM is to find an enclosing boundary for the normal samples in the kernel space. The classification performance is compared to multi-class support vector machines (SVM). In SVM, the Gaussian radial basis function (RBF) is used as the kernel. The Gaussian width is set to the mean of pairwise distances among training samples. For one class SVM, the parameter ν that is used to control the percentage of training data that is allowed outside the enclosing boundary is set to 5%. For multi-class SVM, the cost parameter for controlling tradeoff between complexity and training accuracy is set to 100 [21].

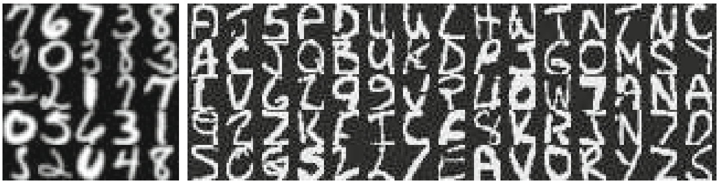


Fig. 3. Sample images from the USPS handwritten digit image dataset (left) and the alphabet and digit (AlphaDigs) image dataset (right)

5.1 Handwritten Digits Recognition

We first experiment on handwritten digit recognition and consider alphabet images as outliers. The USPS handwritten digit dataset is used. It consists of 4649 training images and 4649 test images of 10 digit classes[1]. The raw image intensity is used as the image feature. For Gaussian process classification, the square exponential covariance function (Eq. (12)) is used.The hyperparameters are tuned by maximizing the marginal likelihood and we simply adopt the values according to that in [14]. On the test data, an overall accuracy of 96.5% is achieved, which is consistent with that reported in [14]. To evaluate the proposed outlier detection scheme, we test the classifier trained on the USPS data on a completely different alphabet and digit (AlphaDigs) dataset[2]. It consists of 39 images for each of the 10 digit classes and 26 alphabet classes. Sample images from both the USPS dataset and the AlphaDigs dataset are shown in Fig. 3.

For outlier detection, the images in the AlphaDigs dataset are sorted according to the novelty measure that is used to determine if a sample is an outlier. In the proposed method, it is the determinant of the covariance matrix in a descending order. For one-class SVM, it is the distance to the enclosing hyperplane (with distance outside the hyperplane being positive) in a descending order. A certain amount of test samples with largest novelty measures are then classified as the outliers. We evaluate the outlier detection performance using the receiver operating characteristic (ROC) curve. The ROC curve is a plot of sensitivity (true positive rate) against specificity (false positive rate). The true positive rate is equal to the number of alphabet images that are correctly detected as outliers divided by the total number of alphabet images. The false positive rate is the number of digit images that are wrongly detected as outliers divided by the total number of digit images. The threshold is set at various values in order to obtain a set of points to plot the curves. From Fig. 4, the proposed outlier detection scheme clearly out-performs that of one-class SVM.

It is also worth mentioning the relative classification performance of Gaussian process classification and multi-class SVM. The results are shown in Table 1. It is observed that while Gaussian process classification and SVM give comparable results on the test data from USPS dataset, the former gives a much better result

[1] Available at `http://www.gaussianprocess.org/gpml/data/`

[2] Available at `http://cs.nyu.edu/~roweis/data.html`

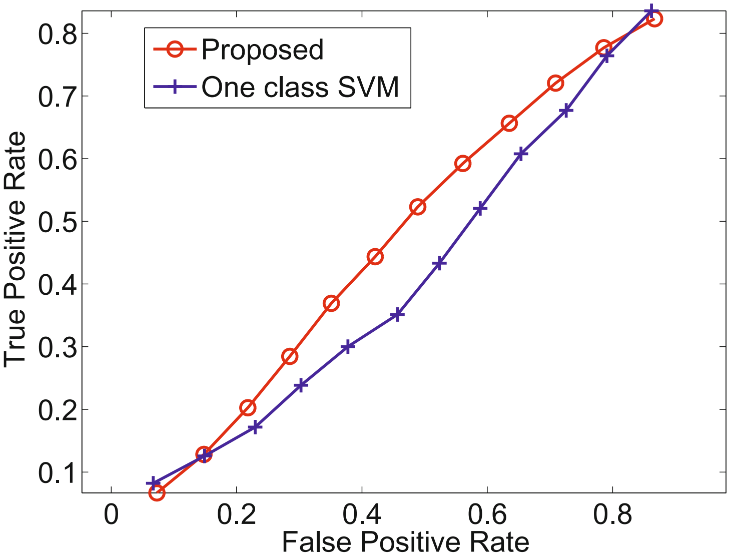


Fig. 4. ROC curves for outlier detection using the proposed method and one class SVM

on the digit images from the AlphaDigs dataset. Since the AlphaDigs dataset is independently collected, and the classifiers are trained on the USPS data, it is considered a more difficult dataset compared with the test set from the USPS data. This shows that Gaussian process classification has a better generalization capability on a different dataset as compared with that of SVM. Note that on the AlphaDigs dataset, only the digit images are used to evaluated the classification performance. This is because the AlpahDigs dataset is dominated by outliers (alphabet images), thus including them will make the evaluation of classification accuracy heavily biased towards detecting more outliers.

Table 1. Classification accuracy comparison using multi-class Gaussian process classification (mcGPC) and multi-class support vector machine (mcSVM)

	mcGPC	mcSVM
On USPS test dataset	96.5%	97.2%
On digit images from the AlphaDigs dataset	72.31%	62.31%

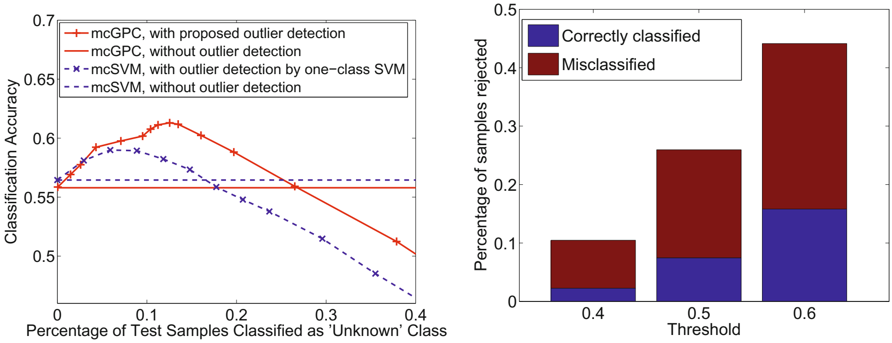
5.2 Localization of Mobile Robots

In this experiment we show the capabilities of Gaussian process classification in terms of both outlier detection and rejection of unreliable predictions. In a robot localization problem, a set of training sequences of various locations are acquired for training a classifier. The classifier answers the question“where am I” when presented with a test sequence. The test sequence may contain locations that were not imaged in the training sequences. These locations should be classified as the ‘unknown’ class. In addition, the classifier may refrain from making a decision when it is not confident about a particular prediction. Therefore, there are two types of uncertainties faced by the classifier when trying to classify a test sample into the training classes. One is that the new sample is not similar to any of the training classes and therefore it is likely to come from a new class. The second type is that the new sample is equally similar to two or more training classes and therefore cannot be classified with strong confidence.

The training and validation data are from the IDOL2 database [22]. The image sequences in the database are acquired using the MobileRobots PowerBot robot platform. The training sequence consists of 1034 image frames of 5 classes according to the robot’s topological location, namely, one-person office(BO), corridor(CR), two-persons office(EO), kitchen(KT), and printer area(PA). The test sequence consists of 1690 image frames classified into 6 classes, 5 of which are the same as those of the training sequence, and one additional unknown(UK) class corresponding to the additional rooms that are not imaged previously. The test sequence is acquired 20 months after the training sequence. For more details please refer to [23].

Gradient based features are chosen as the robot is in indoor environment with strong edge characteristics. Each training image in the training sequence is described by normalized Gaussian derivatives on the L component of the LAB color space. 5 partial derivatives $(L_x, L_y, L_{xx}, L_{yy}, L_{xy})$ are computed and quantized into 32 bins built by k-means. A three-tier spatial pyramid of histograms is then obtained on each image. Each image is represented by a 672 dimensional feature vector.

Using Gaussian process classification, an overall classification accuracy of 55.8% is obtained (the unknown class is treated equally as the training classes in calculation of classification accuracy). Note that the test data include about 20% outliers which are from locations that were not imaged in the training sequence. Without outlier detection, these 20% outliers will be classified into one of the training classes and this explains the low overall classification accuracy. Fig. 5 (a) shows the improvement in classification accuracy if a certain amount of samples are classified as outliers based on the proposed novelty measure. If the prior knowledge that about 20% outliers are present, the classification accuracy is improved to about 59%. We compare the performance with that of multi-class support vector classification with outlier detection by one class SVM. As addressed earlier, in this case, the classification process is independent from the outlier detection. Without outlier detection, the classification accuracy is 56.45%, which is slightly better than that of Gaussian process classification. With outlier detection using one class SVM, the classification accuracy also improves with a certain amount of samples detected as outliers. But the improvements are not as much as that of the proposed method, and we observe a narrower window before the classification accuracy drops below the baseline (Fig. 5(a)).



(a) (b)

Fig. 5. (a) Classification accuracy using multi-class GPC and multi-class SVM, with or without outlier detection. (b) Rejection of unreliable predictions.

Further, if we make use of the rule proposed in section 4 to reject unreliable predictions at a threshold of winning classes's probability exceeding 0.4, 0.5, and 0.6, the classification accuracy further improves to 62.19%, 66.51%, and 71.61%, respectively. Figure 5(b) shows the relative proportions of correctly classified and misclassified in the rejected samples. It is observed that it is dominated by misclassified samples, showing that the reject rule is useful in reject unreliable predictions. If rejection of a sample has a lower cost than misclassifying a sample, the reject rule could help reduce the overall cost of the classification. For example, if the cost of correctly classifying a sample is 0, wrongly classifying a sample is 1, and making no decision about a sample is 0.5, the savings in cost by rejecting at the three thresholds are 50.5, 93.5, and 106, respectively.

6 Conclusion

In this paper, we explore the outlier detection capability of a Gaussian process classifier. It is shown that the determinant of the covariance matrix from the output of the Gaussian process classifier is a good measure of how novel a test sample is compared to the training samples. With this discovery, Gaussian process classifier, as a probabilistic classifier, is able to handle both outlier detection and rejection of unreliable predictions. Experiments on two practical applications show the advantages of the Gaussian process classification with outlier detection.

References

1. Markou, M., Singh, S.: Novelty detection: a review-part 1: statistical approaches. Signal Processing 83, 2481–2497 (2003)

2. Automatic Outlier Detection: A Bayesian Approach. In: 2007 IEEE International Conference on Robotics and Automation (2007)
3. Tax, D.M.J., Duin, R.P.W.: Growing a multi-class classifier with a reject option. Pattern Recogn. Lett. 29, 1565–1570 (2008)
4. Singh, S., Markou, M.: An approach to novelty detection applied to the classification of image regions. IEEE Trans. on Knowl. and Data Eng. 16, 396–407 (2004)
5. Lukashevich, H., Nowak, S., Dunker, P.: Using one-class svm outliers detection for verification of collaboratively tagged image training sets. In: ICME 2009, pp. 682–685 (2009)
6. Loureiro, A., Torgo, L., Soares, C.: Outlier detection using clustering methods: a data cleaning application. In: Proceedings of the Data Mining for Business Workshop (2004)
7. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. SIGMOD Rec. 29, 427–438 (2000)
8. Lauer, M.: A mixture approach to novelty detection using training data with outliers. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 300–311. Springer, Heidelberg (2001)
9. Xing, H., Wang, X., Zhu, R., Wang, D.: Application of kernel learning vector quantization to novelty detection, pp. 439–443 (2008)
10. Muñoz, A., Moguerza, J.M.: One-class support vector machines and density estimation: The precise relation. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 216–223. Springer, Heidelberg (2004)
11. Chen, Y., Zhou, X.S., Huang, T.: One-class Svm for Learning in Image Retrieval, vol. 1, pp. 34–37 (2001)
12. Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: Integrating novel class detection with classification for concept-drifting data streams. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5782, pp. 79–94. Springer, Heidelberg (2009)
13. Hempstalk, K., Frank, E.: Discriminating against new classes: One-class versus multi-class classification. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 325–336. Springer, Heidelberg (2008)
14. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2006)
15. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. In: ACM Special Interest Group on Graphics and Interactive Techniques Conference (SIGGRAPH), pp. 522–531 (2004)
16. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: Advances in Neural Information Processing Systems, vol. 16 (2004)
17. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2001)
18. Clifton, L.A., Yin, H., Clifton, D.A., Zhang, Y.: Combined support vector novelty detection for multi-channel combustion data. In: ICNSC, pp. 495–500 (2007)
19. Tax, D.M.J., Ypma, A., Duin, R.P.W.: Support Vector Data Description Applied to Machine Vibration Analysis (1999)
20. Heller, K.A., Svore, K.M., Keromytis, A.D., Stolfo, S.J.: One class support vector machines for detecting anomalous windows registry accesses. In: Proc. of the Workshop on Data Mining for Computer Security (2003)

21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines Software (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
22. Luo, J., Pronobis, A., Caputo, B., Jensfelt, P.: Incremental learning for place recognition in dynamic environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA (2007)
23. Caputo, B., Pronobis, A., Jensfelt, P.: Overview of the clef 2009 robot vision track. In: CLEF working notes 2009, Corfu, Greece (2009)