Novelty Detection Using Sparse Online Gaussian Processes for Visual Object Recognition

Ruben Ramirez-Padron [a], Boris Mederos [b], Avelino J. Gonzalez [a]

[a] Intelligent Systems Lab, Department of Electrical Engineering and Computer Science, University of Central Florida
rramirez@knights.ucf.edu, avelino.gonzalez@ucf.edu

[b] Departamento de Física y Matemática, Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez
borismadrazo@gmail.com

Abstract

Gaussian processes (GPs) have been shown to be highly effective for novelty detection through the use of different membership scores. However, applications of GPs to novelty detection have been limited only to batch GP, which require all training data at once and have quadratic space complexity and cubic time complexity. This paper proposes the use of sparse online GP (SOGP) for novelty detection, overcoming these limitations. Our experiments show that SOGP-based novelty detection is capable of achieving performances similar to those from batch GP, even under strong sparseness constraints. Additionally, it is suggested here that membership scores that combine the posterior mean and the posterior variance of the GP might be better fitted to novelty detection than scores leveraging only one of the two posterior moments.

Introduction

The problem of novelty detection (one-class classification, anomaly detection, or outlier detection) consists of learning a model of a target class, such that observations that are not members of the target class can be effectively detected. Novelty detection has received much attention recently, because of its numerous applications in fields like cyber-security, data mining, bioinformatics, computer vision, finance, and signal processing (Chandola, Banerjee, and Kumar 2009). Novelty detection is particularly important when it is either very expensive or virtually impossible to collect a representative sample of objects that do not belong to the target class.

Kernel-based methods like Support Vector Data Description (SVDD) (Tax and Duin 2004) and one-class Support Vector Machines (Schölkopf et al. 2001) are among the most successful novelty detection methods, also having strong mathematical foundations. Gaussian processes (GPs) are kernel-based methods defined within a Bayesian framework. There are three learning approaches to GPs: batch GP, online GP and the sparse online GP (SOGP) (Csató and Opper 2002). GPs have been used primarily for solving regression and classification problems (Rasmussen and Williams 2006), and to estimate probability density functions (Adams, Murray, and MacKay 2009). However, it was only very recently that GPs were proposed to solve novelty detection problems (Kemmler, Rodner, and Denzler 2010). The work of Kemmler et al. showed that GPs can outperform the state-of-the-art SVDD method. However, the use of GPs for novelty detection has been limited to the batch learning approach. The main limitations of batch GPs are that training data have to be available before training the GP and it has to remain in memory for prediction purposes. Additionally, batch GP has a quadratic space complexity and a cubic time complexity. These are strong limitations for most modern problems, where data is not always available up front and there are huge datasets to learn from. We propose the use of SOGP for novelty detection, which overcomes those limitations. The main purpose of this work is to determine whether SOGP is capable of performing at a level comparable to batch GP, while taking advantage of its memory savings and its online learning capabilities. For the sake of completeness, the online GP (i.e. non-sparse) was also included in our experiments. We compare the four membership scores that were employed by (Kemmler, Rodner, and Denzler 2010).

The following section gives an introduction to GPs. Subsequently, we review the work by Kemmler et al., which proposed using batch GP for novelty detection. We describe the spatial pyramid matching (SPM) kernel (Lazebnik, Schmid and Ponce 2006), which is the kernel that we used. Finally, our experimental work shows that SOGP-based novelty detection is capable of performing similar to batch GP-based novelty detection, even under strong sparseness constraints.

Gaussian Processes

GPs are nonparametric kernel-based function estimation techniques that model a probability distribution over a space $\mathcal{F}$ of functions $f: \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is a continuous input space. GPs employ a Bayesian learning approach, providing an assessment of the uncertainty associated to predicting the estimated function *f* at any given point $\mathbf{x} \in \mathcal{X}$. The following definition of Gaussian processes is taken from (Rasmussen and Williams 2006):

Definition: *A Gaussian process is a collection of random variables* $\{f_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$, *such that any finite subcollection* $\boldsymbol{f} = \{f_{\mathbf{x}_1}, f_{\mathbf{x}_2}, \dots, f_{\mathbf{x}_m}\}$ *has a joint Gaussian distribution*.

This definition implies that a GP is completely determined by its mean and covariance functions:

$$m(f_{\mathbf{x}}) = \mathbb{E}[f_{\mathbf{x}}]$$
$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}\left[\left(f_{\mathbf{x}_i} - m(f_{\mathbf{x}_i})\right)\left(f_{\mathbf{x}_j} - m\left(f_{\mathbf{x}_j}\right)\right)\right]$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = cov(f_{\mathbf{x}_i}, f_{\mathbf{x}_j})$ is a positive definite kernel function.

In general, estimating the posterior GP implies obtaining expressions for its posterior mean and its posterior covariance function. In this section we briefly describe batch GP, online GP, and SOGP in the context of nonparametric regression, which is the GP formulation used in this paper.

Given a set of input-output observations $D = \{\mathbf{X}, \mathbf{y}\} = \{(\mathbf{x}_i, \mathrm{y}_i): \mathbf{x}_i \in \mathcal{X},\ \mathrm{y}_i \in \mathbb{R}, i = 1 \dots n\}$, it is required to estimate the mapping from the input space $\mathcal{X}$ to the output space $\mathbb{R}$. The classic approach assumes that $\mathrm{y}(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$, such that $\mathrm{y}|f_{\mathbf{x}} \sim N(m(f_{\mathbf{x}}), \sigma^2)$. In this context, the variables $f_{\mathbf{x}}$ are used as random latent variables. Given the set *D*, the posterior distribution of $f_{\mathbf{x}_*}$ for any given test point $\mathbf{x}_*$ is as follows:

$$f_{\mathbf{x}_*}|D, \mathbf{x}_* \sim N(\mu_*,\ \sigma_*^2)$$

where

$$\mu_* = \boldsymbol{k}_*^T(\boldsymbol{K} + \sigma^2\, I)^{-1}\mathbf{y}$$
$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) + \boldsymbol{k}_*^T(\boldsymbol{K} + \sigma^2\, I)^{-1}\, \boldsymbol{k}_*$$

Here $\boldsymbol{k}_* = \left(k(\mathbf{x}_*, \mathbf{x}_1), \dots, k(\mathbf{x}_*, \mathbf{x}_n)\right)^T$ and $\boldsymbol{K}$ denotes the covariance matrix $k(\boldsymbol{X}, \boldsymbol{X})$ that is obtained by evaluating the kernel in each pair of input points from *D*. The time complexity of GP regression is $\mathcal{O}(n^3)$, due to the calculation of an inverse matrix. The posterior moments μ_* and σ_*^2 can be written more compactly as follows:

$$\mu_* = \boldsymbol{k}_*^T\boldsymbol{\alpha}$$
$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) + \boldsymbol{k}_*^T\boldsymbol{C}\,\boldsymbol{k}_*$$

where

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \sigma^2\, I)^{-1}\mathbf{y}$$
$$\boldsymbol{C} = (\boldsymbol{K} + \sigma^2\, I)^{-1}$$

The online GP (Csató and Opper 2002) process the training data one observation at a time. The iterative formulations for the moments of the online GP are:

$$\mu_{t+1} = \boldsymbol{k}_{t+1}^T\boldsymbol{\alpha}_{t+1}$$
$$\sigma_{t+1}^2 = k(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) + \boldsymbol{k}_{t+1}^T\boldsymbol{C}_{t+1}\boldsymbol{k}_{t+1}$$

where $\boldsymbol{k}_{t+1} = \left(k(\mathbf{x}_{t+1}, \mathbf{x}_1), \dots, k(\mathbf{x}_{t+1}, \mathbf{x}_n)\right)^T$. The vector $\boldsymbol{\alpha}_{t+1}$ and the matrix $\boldsymbol{C}_{t+1}$ are updated whenever a new training observation $(\mathbf{x}_{t+1}, \mathrm{y}_{t+1})$ is processed, building upon their previous values at step *t*. Details about their recursive structures are given by (Csató and Opper 2002).

The main limitation of the batch and online GPs is that both require keeping all training data in memory. This is a strong limitation for most practical problems. SOGP (Csató 2002) overcomes this issue by establishing a capacity parameter *m* that determines the maximum number of relevant observations to keep in memory throughout the online learning process. The set of relevant observations is called the basic vectors set (*BV* set). Given a new training data $(\mathbf{x}, y)$ at learning step *t*+1, the vector $\mathbf{x}$ is decomposed as the sum of its projection onto the span of the *BV* set plus a residual vector v_{res} that is orthogonal to the space spanned by the *BV* set. The value $\gamma = \|v_{res}\|$ measures how close the input vector $\mathbf{x}$ is to the space spanned by the *BV* set. If γ is less than some small tolerance ϵ, then the data point is considered redundant and the projection of $\mathbf{x}$ onto the space spanned by *BV* set is used to update $\boldsymbol{\alpha}_{t+1}$ and $\boldsymbol{C}_{t+1}$ without increasing their sizes and without changing the *BV* set. Otherwise, the vector $\mathbf{x}$ is included in the *BV* set, and the updated $\boldsymbol{\alpha}_{t+1}$ and $\boldsymbol{C}_{t+1}$ reflect the inclusion of the new vector by increasing its size accordingly. If the size of the *BV* set went over its capacity *m* after adding a new vector to it, then the *BV* vector contributing the least to the representation of the GP is removed from the *BV* set, carrying out a recomputation of $\boldsymbol{\alpha}_{t+1}$ and $\boldsymbol{C}_{t+1}$. SOGP has a better space complexity than GP ($\mathcal{O}(m^2)$ instead of $\mathcal{O}(n^2)$); and it achieves a time complexity that is linear with respect to data size: $\mathcal{O}(nm^2)$ (Csató 2002).

Novelty Detection with Gaussian Processes

The training data for the problem of novelty detection is denoted here as $D = \{\mathbf{X}, \mathbf{y}\} = \{(\mathbf{x}_i, \mathrm{y}_i): \mathbf{x}_i \in \mathcal{X},\ \mathrm{y}_i = 1, i = 1 \dots n\}$, i.e., it contains objects only from the target class. The main idea behind using GPs for novelty detection, as proposed by (Kemmler, Rodner, and Denzler 2010), is to start with a GP having a prior mean function equal to zero. After training the GP on *D*, if a test observation $\mathbf{x}_*$ is very near to points in *D* then the corresponding posterior mean μ_* will be close to 1. On the other hand, μ_* will be closer to zero for data points distant from training observations. For test observations that are increasingly distant from *D* the posterior variance of the GP (σ_*^2) will be higher than for observations that are close to *D*. Consequently, class membership scores based on the posterior mean, the posterior variance, or both combined can be used to detect novel observations. The lower the membership score of an input $\mathbf{x}$, the higher the likelihood of $\mathbf{x}$ being an outlier.

Table 1 lists the four scores considered by (Kemmler, Rodner, and Denzler 2010): Probability (*P*), Mean (*M*), Negative Variance (*V*) and Heuristic (*H*). The score *V* was previously proposed as part of a clustering technique (Kim and Lee 2006), and score *H* has been successfully applied to object categorization (Kapoor et al. 2010). The experiments reported by Kemmler et al. compared the performance of these four scores, calculated for GP regression and binary GP classification, to Support Vector Data Description (SVDD) (Tax and Duin 2004). Those experiments were run on all object categories (classes) of the Caltech 101 image database (Fei-Fei, Fergus and Perona 2004), where one class at a time was used as the target class. Each experiment repeatedly assessed the performances of SVDD and each GP-based scoring method by estimating the corresponding AUC values, i.e. the areas under the receiver operating characteristic (ROC) curves. Subsequently, an average AUC value was obtained by Kemmler et al. for each method by averaging its performance across all classes. No detailed analyses of performance on single image categories were offered as part of that work, and no variances or confidence intervals were reported for the AUC values. Two image-based kernel functions were employed: the pyramid of oriented gradients (PHoG) (Bosch, Zisserman and Munoz 2007) and the SPM kernel (Lazebnik, Schmid and Ponce 2006).

Kemmler et al. reported that scores using GP regression were consistently similar or better than the corresponding scores based on binary GP classification. Consequently, the focus of their work was on comparing SVDD to the scores calculated using GP-regression (denoted by Reg-*P*, Reg-*M*, Reg-*V*, and Reg-*H*). Additionally, AUC values obtained through the SPM kernel were consistently higher across all methods than the corresponding AUC values obtained through the other kernel. That motivated Kemmler et al. to focus their conclusions on results coming only from the SPM kernel. Average AUC values from GP regression were better than those obtained from SVDD for all membership scores except when using Reg-*M.* The Reg-*M* score showed a great variation in performance across image categories. Finally, Kemmler et al. proposed Reg-*V* as the score of choice, given that detection based on it significantly outperformed all other methods when using the SPM kernel.

Membership score	Expression
Probability (*P*)	$p(y_* = 1 \mid \boldsymbol{X}, \mathbf{y}, \mathbf{x}_*)$
Mean (*M*)	$\mu_* = E(y_* \mid \boldsymbol{X}, \mathbf{y}, \mathbf{x}_*)$
Negative Variance (*V*)	$-\sigma_*^2 = -Var(y_* \mid \boldsymbol{X}, \mathbf{y}, \mathbf{x}_*)$
Heuristic (*H*)	$\mu_* \sigma_*^{-1}$

Table 1. Membership scores for novelty detection using GPs (Kemmler, Rodner, and Denzler 2010).

Spatial Pyramid Matching

The SPM kernel (Lazebnik, Schmid and Ponce 2006) works on descriptor vectors calculated from the images that are input to the kernel. It takes into account coarse spatial information about local features from those images. The SPM kernel makes use of the pyramid match kernel (Grauman and Darrell 2005). Consequently, it is important to understand pyramid matching in order to fully understand SPM.

Let us denote by X_1 and X_2 two sets of local features, e.g. SIFT descriptors (Lowe 1999); each set obtained from one image. Both feature sets must take values in the same *d*-dimensional feature space. The original pyramid matching kernel is applied as follows: A sequence of increasingly finer grids with resolutions 0, 1, … , L is placed over the feature space, so that at each resolution l the corresponding grid has a total of $D_l = 2^{dl}$ cells. Any two points, one from X_1 and the other from X_2, are a match at resolution l if they fall into the same cell at that resolution. At each resolution l, a histogram is built for each feature set, with each bin corresponding to a different grid cell. The histogram intersection function $I(.,.)$ (Swain and Ballard 1991) is used to calculate the total number of feature matches at a given resolution *l*:

$$I_l = I\left(H_{X_1}^l, H_{X_2}^l\right) = \sum_{i=1}^{D_l} \min\left(H_{X_1}^l(i), H_{X_2}^l(i)\right)$$

where $H_{X_1}^l$ and $H_{X_2}^l$ denotes the histograms at resolution l for X_1 and X_2 respectively. Finally, the pyramid match kernel κ_L is calculated as follows:

$$\kappa_L(\mathrm{X}_1, \mathrm{X}_2) = \frac{1}{2^L} I_0 + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} I_l$$

The SPM kernel uses spatial information by applying the pyramid match kernel in the two-dimensional image space. Before doing that, each feature set is quantized into *M* feature types, where *M* is a fixed number. SPM then applies the pyramid match kernel in the image space *M* times, each time constrained to the coordinates associated to the corresponding feature type. The SPM kernel is defined as the sum of pyramid match kernels on the image space over the *M* feature types:

$$k_L(\mathrm{X}_1, \mathrm{X}_2) = \sum_{m=1}^{M} \kappa_L\left(C_{X_1}^m, C_{X_2}^m\right)$$

where $C_{X_i}^m$ denotes the image coordinates associated to features from the feature set $\mathrm{X_i}$ that are of type *m*. A normalization of histograms by the total weight of all features in the image allows evaluating the kernel on images of different sizes.

A pre-processing step is needed to create a dictionary of size *M*, which is used to quantize the feature set from each image. The elements of the dictionary were selected by Lazebnik et al. as the centroids of the *M* clusters obtained by applying *k*-means to features taken from all classes from multi-class classification problems. Quantization of

each feature vector was performed by choosing its nearest element from the dictionary. It was shown by Lazebnik et al. that support vector machines using the SPM kernel outperform other modern classifiers on three image datasets, including the Caltech 101 database (Fei-Fei, Fergus and Perona 2004).

Experimental Setup

We decided to use the Caltech 101 database, given the previously reported good performance of the SPM kernel on it. Additionally, using Caltech 101 allowed us to contrast our results to those of (Kemmler, Rodner, and Denzler 2010), because it was the database used in their work. Contrary to the work of Kemmler et al., which focused on average performance of novelty detectors across all image categories, our work focused on individual object categories from the Caltech 101 dataset. More importantly, Kemmler et al. were concerned exclusively with the application of batch GP to novelty detection. We compared the performances of the membership scores using online GP regression, SOGP regression, and batch GP regression. We performed a detailed analysis of these novelty detection methods on four of the object categories on which the SPM kernel achieved high classification performance (minaret, Windsor chair, Joshua tree, and okapi) and four of the categories for which its classification performance was poor (cougar body, beaver, crocodile, and ant), as reported by (Lazebnik, Schmid and Ponce 2006). These categories were chosen for our experiments because they are key examples from that work.

We used the Matlab implementation of the SPM kernel from (Lazebnik, Schmid and Ponce 2006), keeping their recommended values for the parameters: $M = 200$, $L = 2$. Similarly, we used their default SIFT descriptors of 16 by 16 pixel patches computed over a dense grid spaced at 8 pixels. However, we generated a dictionary for each target class, which is the only training data available for the corresponding recognition problem. The pyramid histograms for all images from the eight categories were built against the dictionary of the target class to be learned at each experiment.

We implemented the three variants of GP regression in Matlab. Our implementation of batch GP was validated against the NETLAB toolbox (Nabney 2004). The online GP and SOGP were validated against Dan Grollman's Online Gaussian Process C++ Library, available at http://lasa.epfl.ch/~dang/code/sogp-2.1.tar.gz.

For each of the eight target classes, we employed 10-fold stratified cross-validation (CV) (Kohavi 1995) to validate the novelty detectors based on each of the three GPs using each of the four membership scores. Stratified CV delivers data folds containing roughly the same class proportions as in the training data. Consequently, at each CV step the GPs were trained on approximately 90% of the target class. SOGP was validated multiple times, each time at a different capacity: 75%, 50%, 25%, and 10% of the number of training observations from the corresponding CV step, rounded to the nearest integer. The estimated GP capacity at each CV step is shown in table 2. The 10-fold CV was repeated 10 times for each target class. AUC values were obtained for each combination of score and GP type on each target class.

Class	Class Size	Capacity 75%	Capacity 50%	Capacity 25%	Capacity 10%
minaret	76	51	34	17	7
windsor chair	56	38	25	13	5
joshua tree	64	43	29	14	6
okapi	39	26	18	9	4
cougar body	47	32	21	11	4
beaver	46	31	21	10	4
crocodile	50	34	22	11	4
ant	42	28	19	10	4

Table 2. Estimated capacities for the SOGPs in our experiments.

Results and Analysis

We focused on the performance of the four scores for each image category using the batch GP, given that it provides a better fitting to the underlying regression problem than the other two GPs (which are approximations to batch GP). Figure 1 shows the 95% confidence intervals of AUC values for each image category and each score using batch GP. Reg-*M* showed a fluctuating performance, which was also reported by (Kemmler, Rodner, and Denzler 2010). Reg-*V*, which was the score of choice by Kemmler et al., performed always worse than Reg-*P* and Reg-*H* in our experiments. It is difficult to determine the reason behind these different results, given that the work by Kemmler et al. averaged AUC values across all classes in Caltech 101 and no confidence intervals or other dispersion indicators were offered. Furthermore, although their work also employed the SPM kernel, a different clustering technique was used to build their dictionaries. In any case, our results show that Reg-*V* is not necessarily the membership score of choice for visual object recognition. For the categories considered here, Reg-*H* and Reg-*P* were consistently better than the other two scores. This suggests that membership scores that use the posterior mean and the posterior variance combined are more appropriate for this problem. Interestingly, Reg-*P* and Re-*H* were the only two scores in the work by Kemmler et al. that significantly outperformed SVDD for the two kernels they used. That fact was not highlighted in their paper given the strong performance of Reg-*V* on their kernel of choice. Finally, Reg-*P* and Reg-*H* were also the best scores when we

repeated our experiments using the online GP and SOGP. Consequently, our subsequent comparison of the three GP variants focused on these two scores.

Our main goal was to study whether SOGP could be used for novelty detection instead of the other two GP variants without losing to much quality of prediction. For each of the two scores, we obtained the boxplots of AUC values for each type of GP on each image category. The plots from Reg-*H* showed no substantial differences to those obtained from Reg-*P*. Figure 2 shows some of the boxplots corresponding to the Reg-*H* score. Novelty detection based on (non-sparse) online GP consistently reported performances almost identical to those obtained through batch GP. This was expected, because online GP also keeps all training observations in memory. Regarding SOGP, the performances obtained when using capacities equal to 75% or 50% of the training data were always very similar and sometimes even slightly better than the performances obtained from the corresponding batch GP. Furthermore, for the categories on which the SPM kernel was previously linked to high classification performance (minaret, Windsor chair, Joshua tree, and okapi) the performances of SOGP with capacity equal to 25% were still similar to that from the batch GP. That is remarkable, given the complexity of the visual recognition problem and the fact that 25% in our experiments translated to capacities frequently under 15 observations. Finally, reducing the capacity of SOGP to 10% of the training data triggered a significant drop in performance in all cases, indicating that keeping so few observations was not enough to capture the complexity of the problem using the SPM kernel. We believe, however, that performance could be further improved at low capacities by using kernels better tailored to this particular problem. For example, the SPM kernel is not invariant to translations and/or rotations. Consequently, several images containing the same object rotated and/or at different locations may be considered worthy of inclusion in the *BV* set by SOGP. This might quickly fill up a *BV* set of a very limited capacity, therefore hindering the full inclusion of other relevant images. If the SPM kernel were modified to be invariant to those transformations then most likely fewer images would be required to be part of the *BV* set; letting more space in the *BV* set available to include further relevant observations.

Conclusions

To the best of our knowledge, this is the first work reporting experimental results from the application of SOGP to the problem of novelty detection. The specific problem chosen for our experiments (visual object recognition on image categories from the Caltech 101 database) was previously used by (Kemmler, Rodner, and Denzler 2010) as a benchmark to show the effectiveness of batch GP for novelty detection. Our work shows that SOGPs can be effectively used for novelty detection using the membership scores Reg-*P* and Reg-*H*. SOGP was capable of performances similar to those of batch GP even at capacities below 15 observations. That is remarkable, given the complexity of the object recognition problem. Moreover, the space and time complexities of SOGP are substantially better than those from batch GP.

The Reg-*P* and Reg-*H* scores consistently outperformed the Reg-*V* and Reg-*M* scores; suggesting that scores that combine the posterior mean and the posterior variance of the GP might be better fitted to GP-based novelty detection. We suggested that kernels that are invariant to affine transformations might improve the performance of SOGP under very limited capacities.

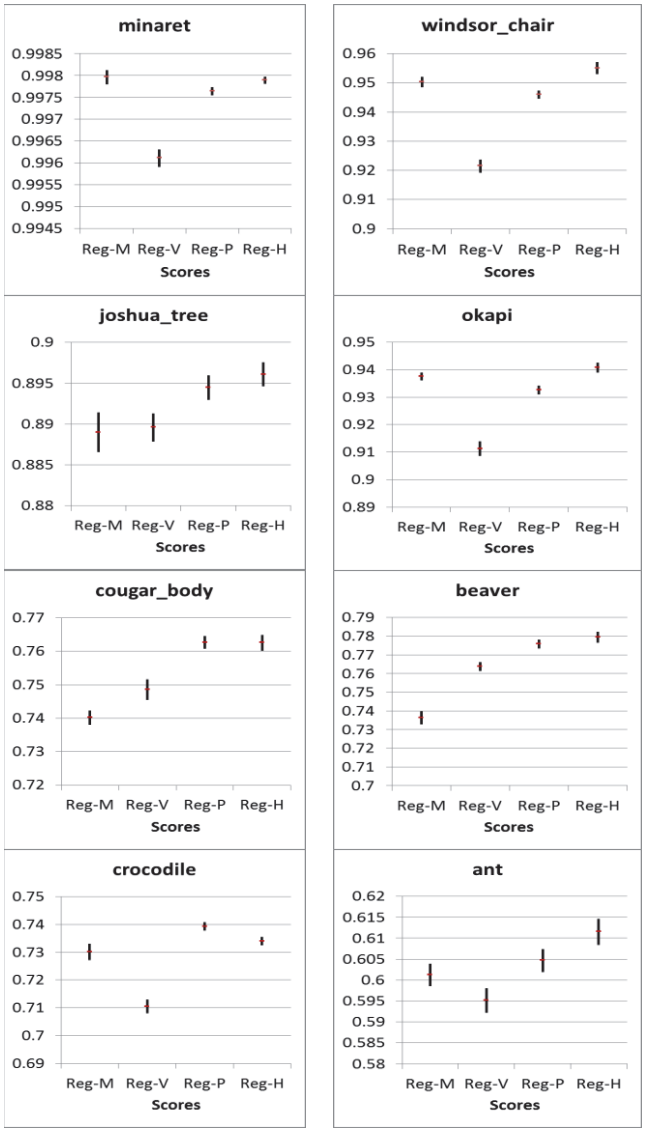

Figure 1. The 95% confidence intervals for AUC values for each membership score using batch GP.

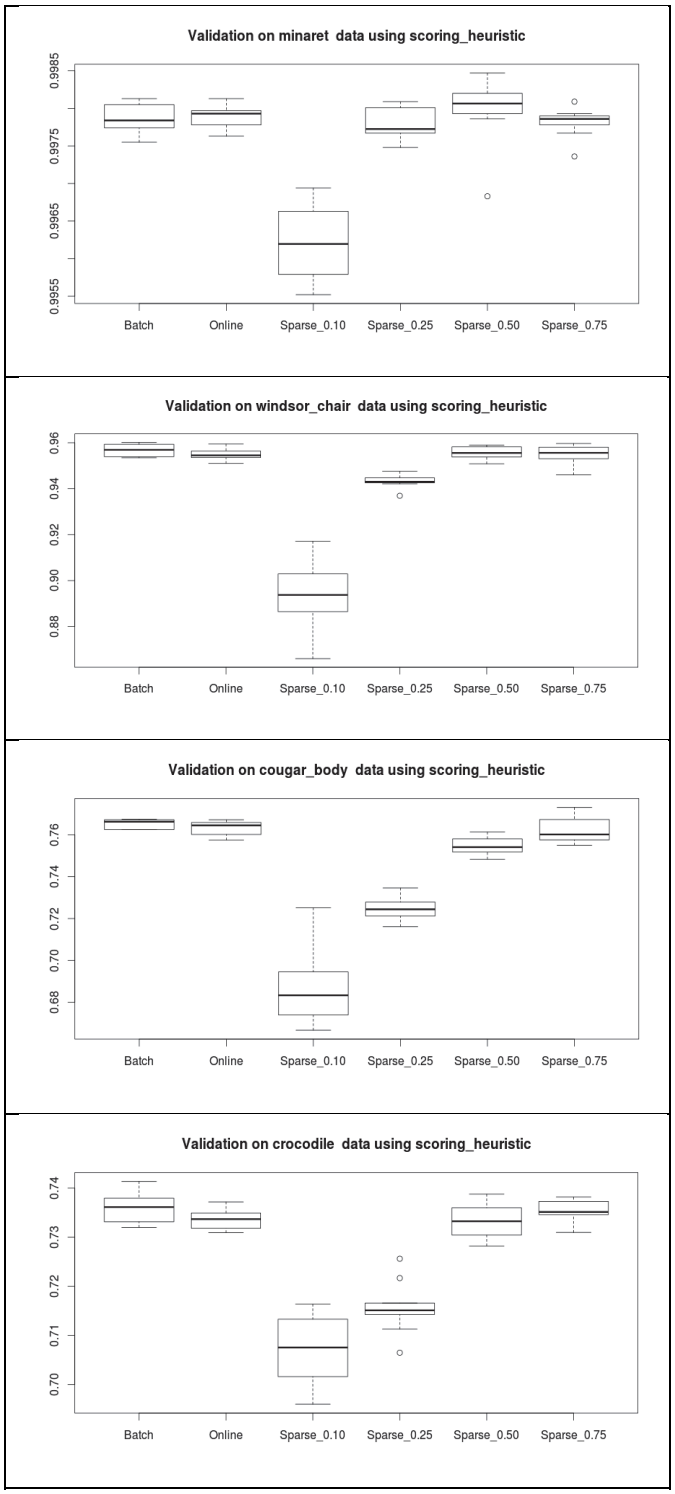

Figure 2. Performances of novelty detection using Reg-H score for different GPs. Results were essentially the same for Reg-P.

Acknowledgments

The second author acknowledges the support of the Mexican PROMEP Program.

References

Adams, R.; Murray, I.; and MacKay, D. 2009. The Gaussian Process Density Sampler. Advances in Neural Information Processing Systems (NIPS 2009), 21:9 – 16.

Bosch, A.; Zisserman, A.; and Munoz, X. 2007. Representing shape with a spatial pyramid kernel. In Proceedings of the 6th ACM International Conference on Image and Video Retrieval, 401 – 408.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys* 41:15:1–15:58.

Csató, L.; and Opper, M. 2002. Sparse On-line Gaussian Processes. *Neural Computation* 14(3):641 – 668.

Csató, L. 2002. Gaussian Processes – Iterative Sparse Approximations. PhD thesis. Birmingham, UK: Aston University.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE CVPR Workshop on Generative-Model Based Vision. http://www.vision.caltech.edu/Image_Datasets/Caltech101/

Grauman, K.; and Darrell, T. 2005. Pyramid match kernels: Discriminative classification with sets of image features. In Proceedings of the International Conference on Computer Vision.

Kapoor, A.; Grauman, K.; Urtasun, R.; and Darrell, T. 2010. Gaussian Processes for Object Categorization. *International Journal of Computer Vision* 88(2): 169 – 188.

Kemmler, M.; Rodner, E.; and Denzler, J. 2010. One-class Classification with Gaussian Processes. In *Proceedings of the Asian Conference on Computer Vision*. Lecture Notes in Computer Science. 6493:489 – 500. Springer.

Kim, H. C.; Lee, J. 2006. Pseudo-density estimation for clustering with Gaussian processes. Advances in Neural Networks, 3971:1238 – 1243.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 1137–1143.

Lazebnik, S,; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition, 2169 – 2178.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In Proceedings of the IEEE International Conference on Computer Vision 2:1150–1157.

Nabney, I. T. 2004. *NETLAB: Algorithms for Pattern Recognition*. Springer.

Rasmussen, C.; and Williams, C. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation* 13(7):1443-1471.

Swain M.; and Ballard D. 1991. Color indexing. *International Journal of Computer Vision* 7(1):11 – 32.

Tax, D. M.; and Duin, R. P. 2004. Support Vector Data Description. *Machine Learning* 54(1):45 – 66.