HYPERSPECTRAL IMAGE CLASSIFICATION USING GAUSSIAN PROCESS MODELS

Michael Ying Yang

Computer Vision Lab (CVLD)
TU Dresden, Germany

Wentong Liao, Bodo Rosenhahn, Zheng Zhang

TNT, Leibniz University Hannover, Germany
Chinese Academy of Sciences, China

ABSTRACT

Hyperspectral image processing has been a very dynamic area in remote sensing and other applications since last decades. Hyperspectral images provide abundant spectral information to identify and distinguish spectrally similar materials. Recent advances in kernel machines promote the novel use of Gaussian processes (GP) for classifying hyperspectral images. Many sophisticated kernel functions have been provided for kernel-based methods. However, different kernel functions has different performance in different applications. This paper introduces GP models with different kernel functions for classifying hyperspectral images. We first provided the mathematical formulation of GP models for classification. Then, several popular kernel functions and their hyperparaeters selection for GP models are introduced. The experiment are performed on three benchmark datasets to evaluate the performances of different kernel functions in terms of classification accuracy. Their performances are compared with each other and discussed in detailed.

Index Terms— Hyperspectral image classification, Gaussian processes, kernel function

1. INTRODUCTION

Kernel machines have received great attention in the remote sensing community since several decades ago. The kernel-based methods have following inherent virtues: 1) tackling high dimensional input spaces efficiently; 2) dealing with noisy samples in a robust way; 3) working with a relatively low number of labeled training samples. These characters make them well-suited to handle the classification problems of hyperspectral images, e.g., the well-known Huges phenomenon caused by a large number of spectral bands and a relatively small number of labeled training samples. In particular, Gaussian Process (GP) models [1, 2] have been proved as an excellent classifier for classifying hyperspectral images in terms of accuracy and robustness. In contrast to another popular kernel machines–SVM [3], GP models provide truly probabilistic outputs with an explicit degree of prediction uncertainty. The probabilistic methods have various advantages in practical recognition circumstances.

Table 1. Summary of several popular kernel functions. The covariances are written either as a function of $\mathbf{x}$ and $\mathbf{x}'$, or as a function of $r = |\mathbf{x} - \mathbf{x}'|$.

covariance function	expression
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$
RBF	$\sigma_0^2 exp(-\frac{r^2}{2l^2})$
ARD	$\sigma_0^2 exp(-\frac{r_b^2}{2l_b^2})$
rational quadratic	$\sigma_0^2 (1 + \frac{r^2}{2al^2})^{-\alpha}$
NN	$\sin^{-1}(\frac{2\tilde{x}^\top \sum \tilde{x}'}{\sqrt{(1+2\tilde{x}^\top \sum \tilde{x})(1+2\tilde{x}'^\top \sum \tilde{x}')}})$

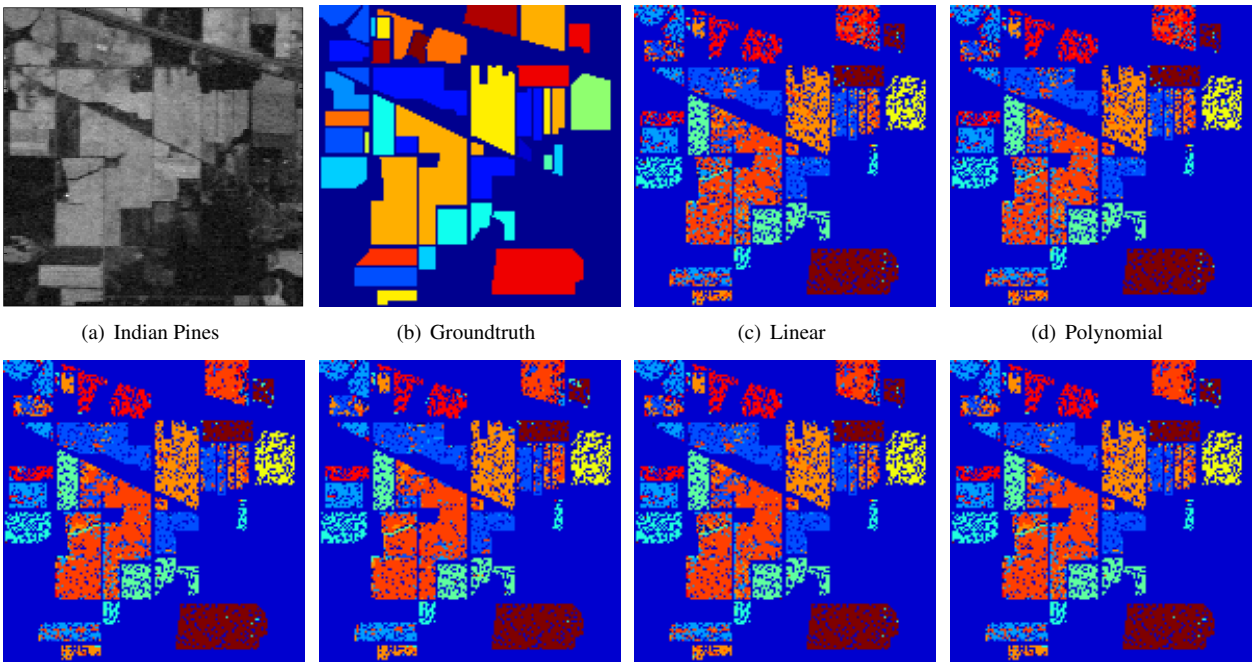
However, as a kernel-based method,the selection of the kernel will crucially affect the performance of GP models. Different kernel has different performances for different kinds of information [4]. In this paper, our goal is to evaluate the performance of GP models using several most popular kernels for classifying hyperspectral images in terms of accuracy.

2. GP MODELS

2.1. GP classification

Given a training set $(\mathbf{X}, \mathbf{y}) = \{\mathbf{X}_n, y_n\}_{n=1}^N$, where N is the number of labeled samples and y_n is the corresponding class label that indicates the land-cover type. Each vector $\mathbf{X}_n \in R^d$ represents the spectral d bands of a pixel in a HSI. We aim at labeling a new test sample set $\mathbf{x} = \{\mathbf{x}_m\}_{m=1}^M$, where M is the number of test samples, by computing the probability $P(y|\mathbf{X}, \mathbf{y}, \mathbf{x})$ belonging to a class. For simple illustrating the binary classification we consider here a target $y_i \in \{-1, +1\}$. The binary classification is easily extended to multiple classification by using the one-against-all or one-against-one strategy.

GP models generate a discrete label y_i for a data point $\mathbf{x}_i$ via a continuous latent variable f_i. A likelihood model $p(\mathbf{y}|\mathbf{f})$ characterizes the monotonic relationship between latent variable $\mathbf{f}$ and the probably observed annotation $\mathbf{y}$. The logis-



(a) Indian Pines (b) Groundtruth (c) Linear (d) Polynomial

(e) Radial basis function kernel (f) Automatic relevance determination (g) Neural network (h) Rational quadratic

Fig. 1. (a) Data of Indian Pines, (b) ground truth, and classification results of kernel (c) linear, (d) polynomial, (e) RBF, (f) ARD, (g) NN, (h) RQ.

tic and probit function are the most common choices. Their forms are:

$$\varphi_{logit}(z) = \frac{1}{1+e^{-z}}, \tag{1}$$

$$\varphi_{probit}(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} exp(-\frac{x^2}{2})dx. \tag{2}$$

Specially, from the equation we can see that the probit function is simply the standard normal cumulative distribution function. The likelihood term is written as:

$$p(y_i = +1|f_i) = \varphi(y_i f_i), \tag{3}$$

An integrating over the latent variable f to predict the probability for sample $\mathbf{x}_i$ is executed as follows:

$$p(y_i = +1|\mathbf{X}, \mathbf{y}, \mathbf{x}_i) = \int p(y_i|f_i)p(f_i|\mathbf{X}, \mathbf{y}, \mathbf{x}_i)df_i \tag{4}$$

where $p(f_i|\mathbf{X}, \mathbf{y}, \mathbf{x}_i)$ is the distribution of latent variable f_i corresponding to $\mathbf{x}_i$, which can be obtained by integrating over the latent variable $\mathbf{F} = (F_1, \ldots, F_n)$ corresponding to training set $(\mathbf{X}, \mathbf{y})$:

$$p(f_i|\mathbf{X}, \mathbf{y}, \mathbf{x}_i) = \int p(f_i|\mathbf{X}, \mathbf{y}, \mathbf{x}_i, \mathbf{F})p(\mathbf{F}|\mathbf{X}, \mathbf{y})d\mathbf{F} \tag{5}$$

where $p(\mathbf{F}|\mathbf{X}, \mathbf{y}) = p(\mathbf{F}|\mathbf{y})p(\mathbf{F}|\mathbf{X}) \;/\; p(\mathbf{y}|\mathbf{X})$ is the posterior over the latent variables. $p(\mathbf{y}|\mathbf{X})$ is the marginal likelihood (evidence) and $p(\mathbf{F}|\mathbf{X})$ is the GP prior over the latent function, which in GP model is a jointly zero mean Gaussian distribution and with the covariance given by the kernel K.

2.2. Kernel functions for GP models

The non-Gaussian likelihood $p(\mathbf{F}|\mathbf{X}, \mathbf{y})$ in Eq. (5) makes the integral analytically intractable. Similarly, Eq.4 might be also analytically intractable for certain sigmoid functions $p(y_i|f_i)$. To solve this problem, a number of approximations have been suggested. The *Monte Carlo Markov Chain* (MCMC) sampling is a standard procedure for posterior inference, but it is computation expensive. Under Gaussian process models, two analytic approximation approaches are commonly applied: *Laplace's* approximation method (LP) [5] and *expectation propagation* (EP) algorithm [6]. They both approximate the non-Gaussian joint posterior with a Gaussian one. In this thesis, the LP method will be adopted because of its less computation and easier inference. Interested readers are referred to [7] for more details on the two methods. Then, the posterior for latent f_i in Eq. (5) is approximated as a Gaussian

$$q(f_i|\mathbf{X}, \mathbf{y}, \mathbf{x}_i) = \mathcal{N}(\mu_i, \boldsymbol{\sigma^2}_i), \tag{6}$$

$$\mu_i = \mathbf{k}(\mathbf{x}_i)^T \mathbf{K}^{-} \widetilde{\mathbf{F}}, \tag{7}$$

$$\boldsymbol{\sigma^2}_i = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{k}(\mathbf{x}_i)^T (\mathbf{K} + \mathbf{W}^{-})\mathbf{k}(\mathbf{x}_i), \tag{8}$$

Table 2. Individual class, OA and AA accuracies in percentage of the Indian Pines data set with different classifiers.

Class	Linear	Poly	RBF	ARD	NN	RQ
Corn-notill	73.45	77.69	78.26	**79.72**	77.77	78.01
Corn-mintill	71.27	80.48	80.79	**90.26**	80.63	80.79
Grass-pasture	90.81	93.76	95.41	**96.48**	94.35	95.41
Grass-Trees	98.87	99.25	99.43	98.80	99.06	**99.62**
Hay-windrowed	98.20	99.64	**100**	**100**	98.92	99.64
Soybean-notill	81.09	83.46	87.31	**89.25**	86.14	87.56
Soybean-mintill	60.84	68.82	71.44	**78.46**	69.93	71.31
Soybean-clean till	90.08	93.13	93.63	**94.60**	93.64	93.38
Woods	98.22	97.84	**99.50**	99.16	98.31	97.84
OA	78.06	81.39	85.5	**87.06**	83.23	83.87
AA	84.76	88.56	89.53	**91.86**	88.75	89.29

where $\mathbf{W} \triangleq -\nabla\nabla \log p(\mathbf{Y}|\widetilde{\mathbf{F}})$ is diagonal. $\mathbf{K}$ denotes a $N - by - N$ covariance matrix between N training points. $\mathbf{k}(\mathbf{x}_i)$ is a covariance vector between N training points X and a test point $\mathbf{x}_i$ and $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_i)$ is covariance for test point $\mathbf{x}_i$ and $\widetilde{\mathbf{F}} = \arg\max_{\mathbf{F}} p(\mathbf{F}|\mathbf{X}, \mathbf{Y})$. Given the mean and variance of f_i, we can obtain the prediction probability in Eq. (4). Specially, if the probit function is chosen, the integration in Eq. (4) is much easier.

The kernel function is the crucial ingredient in GP predictor and its hyperparameters Θ crucially affect its performance. Table 1 gives a summary of several popular kernel functions. In particular, the Gaussian radial basis function (RBF) is one of the most widely used kernels because of its robustness for different types of data:

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_0^2 exp(-\frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{2l^2}) \qquad (9)$$

$\Theta = [\sigma_0^2, l]$ is the hyperparameter set for RBF, of which l in the function is the characteristic length-scale. The smaller l we choose, the more rapidly the function varies. Moreover, if l varies with input dimensions (i.e. input bands), e.g., $l = [l_1, \ldots, l_d]$, there is another kernel called the Automatic Relevance Determination (ARD) which is derived form RBF:

$$K_{ARD}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_0^2 exp(-\sum_{b=1\ldots d} \frac{\| x_i^b - x_j^b \|^2}{2l_b^2}) \qquad (10)$$

x_i^b indicates the bth band of the ith input point. The ARD has been proved to be an effective kernel successfully removing irrelevant information [7]. It provides a parametrization scheme for automatic feature reduction especially for the high-dimensional challenge such as HSI with more than one hundred bands.

Another interesting kernel function is Neural Network (NN) and its typical form is as follows [8]:

$$K_{NN} = \sigma_0^2 \sin^{-1}\left(\frac{2\tilde{x}^\top \sum \tilde{x}'}{\sqrt{(1 + 2\tilde{x}^\top \sum \tilde{x})(1 + 2\tilde{x}'^\top \sum \tilde{x}')}}\right) \qquad (11)$$

where $\tilde{\mathbf{x}} = [1, x_1, x_2, \ldots, x_d]^T$, and $\tilde{\sum}$ denotes a covariance matrix that may take structural parameterization. The importance of the NN kernel lies in that a GP model with this kernel is equivalent an infinite Bayesian NN with the probit transfer function.

3. EXPERIMENTAL RESULTS

In the experiments, three benchmark hyperspectral datasets-INDIAN PINES, UNIVERSITY OF PAVIA, and CENTER OF PAVIA will be exploited for the assessment of Gaussian Process classification (GPC) performance with different kernels. These datasets have been widely used as benchmark [3, 4, 1, 2] in the study of hyperspectral image classification. The INDIAN PINES dataset was acquired by the AVIRIS in 1992 and taken over a predominately agricultural region in NW Indiana, USA. The dataset is a 145×145 pixels scene and has

Table 3. Individual class percentage accuracies of the University of Pavia dataset with different classifiers.

Class	Linear	Poly	RBF	ARD	NN	RQ
Asphalt	78.37	82.04	82.09	**84.61**	81.59	82.54
Meadows	87.18	90.02	91.06	**92.39**	90.67	90.98
Gravel	79.04	82.52	84.89	84.41	84.89	**85.15**
Trees	93.51	96.51	96.68	96.02	**96.89**	96.65
Metal	98.95	99.21	99.30	99.21	**99.65**	99.30
Bare Soil	78.40	93.11	92.05	**93.21**	89.96	92.36
Bitumen	86.20	92.57	**95.75**	92.57	95.40	95.49
Bricks	78.37	94.96	87.05	85.12	85.90	87.16
Shadow	99.87	**100**	**100**	**100**	**100**	**100**
OA	84.61	89.44	90.06	**90.87**	83.23	90.14
AA	86.65	91.53	92.10	91.95	88.75	**92.18**

200 channels. Seven of the 16 different land-cover classes in the original ground-truth were removed, which can offer only a few training samples (this makes the experimental analysis more significant from the statistical viewpoint) [3]. The CENTER OF PAVIA image lies around the center of Pavia with 1096×492 pixels and remains 102 channels after removing some noisy bands. The ground-truth consists of 9 land-cover classes. The UNIVERSITY OF PAVIA dataset has 103 channels with 610×340 pixels and also 9 land-cover classes. For each dataset, 200 points of each class were randomly selected as training data and the rest points were test data. If the the amount of data points of any classes are less than 200, 50% of its points were selected as training data. All kernels listed in Table 1 were adopted in the GP models for comparison purpose and their hyperperameters were optimized by adopting *Conjugate Gradient* method [9] based on the *Laplace* approximation method. In order to simplify the classification and balance the samples problems, the one-against-one strategy was applied.

The original image and ground truth of Indian Pines dataset are shown in Fig. 1(a) and Fig. 1(b) respectively. From Fig. 1(c) to Fig. 1(h) are the classification results of GP model with different kernel functions. Table 2 shows the individual class accuracy, overall accuracy (OA) and average accuracy (AA) of GPC with different kernel functions from the Indian Pines dataset. The results show that the ARD kernel has the best accuracy. However, in order to optimize more parameters for ARD kernel, more input dimensions increase the training time rapidly. The classification accuracy of RBF kernel and rational quadratic (RQ) kernel are similar. From their expressions we know that the main difference between them lies in the exponents. After optimizing hyperparameter, γ of RQ varies nearby 2 with high probability in this scene. Hence their results are close. However, because γ can change according to different data, RQ is more flexible than RBF in real application. The NN kernel performs relatively well. It provides an effective solution for GPC. The performance of polynomial kernel is not very satisfactory, yet it has a much simpler expression and lower computation cost than the other four discussed above kernels. The linear kernel has the worst performance, because it cannot describe the similarity and difference between two input feature vectors. But its computation cost is the lowest. For some simpler datasets, it may give compatible performance with faster training. Table 3 and Table 4 list the classification accuracy of individual class on University of Pavia and Center of Pavia dataset, respectively. From the tables we can see, different kernel functions has different performance for different datasets and land-cover classes.

Table 4. Individual class percentage accuracies of the Center of Pavia dataset with different classifiers.

Class	Linear	Poly	RBF	ARD	NN	RQ
Water	99.82	99.79	99.79	99.77	**99.89**	99.79
Trees	94.25	94.66	94.83	92.69	95.517	94.86
Asphalt	95.45	**97.45**	97.34	96.08	96.97	97.34
Bricks	87.84	94.33	94.69	**96.03**	94.64	95.00
Bitumen	92.63	95.75	96.60	96.19	91.16	**96.65**
Tiles	97.35	**97.89**	97.66	97.51	97.66	97.77
Shadow	85.37	88.83	89.84	**91.72**	89.52	89.77
Meadows	99.45	99.62	**99.66**	99.56	99.62	**99.66**
Bare Soil	99.90	99.90	**99.95**	99.90	99.90	**99.95**
OA	97.48	98.15	98.27	98.21	**98.32**	98.29
AA	94.67	96.47	96.71	96.60	96.65	**96.75**

4. CONCLUSION

This paper discussed the properties of several most popular kernel functions in GP models for classifying hyperspectral images. Their performances are evaluated for classifying three benchmark hyperspectral image datasets in terms of accuracy. This work will be a reference for the kernel selection of GPs for classifying hyperspectral images.

5. REFERENCES

[1] K. Zhao, S. Popescu, and X. Zhang, "Bayesian learning with gaussian processes for supervised classification of hyperspectral data," *Photogrammetric Engineering & Remote Sensing*, vol. 74, no. 10, pp. 1223–1234, 2008.

[2] W. Liao, J. Tang, B. Rosenhahn, and M.Y. Yang, "Integration of gaussian process and mrf for hyperspectral image classification," in *Joint Urban Remote Sensing Event*, 2015.

[3] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[4] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[5] C. Williams and D. Barber, "Bayesian classification with gaussian processes," *IEEE Trans. PAMI*, vol. 20, no. 12, pp. 1342–1351, 1998.

[6] T.P. Minka, *A family of algorithms for approximate Bayesian inference*, Ph.D. thesis, MIT, 2001.

[7] C.E. Rasmussen, *Gaussian processes for machine learning*, The MIT Press, 2006.

[8] R. Neal, *Bayesian learning for neural networks*, Ph.D. thesis, University of Toronto, 1995.

[9] S. Wright and J. Nocedal, *Numerical optimization*, vol. 2, Springer, 1999.