I.J. Image, Graphics and Signal Processing, 2015, 9, 1-10
Published Online August 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2015.09.01

Texture Analysis of Remote Sensing Imagery with Clustering and Bayesian Inference

Jiang Li
Department of Computer Science and Information Technology Austin Peay State University, Clarksville, TN 37044, USA
Email: lij@apsu.edu

William Rich and Donald Buhl-Brown
Department of Computer Science and Information Technology Austin Peay State University, Clarksville, TN 37044, USA
Email: {wrich1, dbuhlbrown}@my.apsu.edu

Abstract**—**Texture is one of the most significant characteristics for retrieving visually similar patterns in remote sensing images. Traditional approaches for texture analysis are based on symbolic descriptions and statistical methods. This study proposes a new method to extract and classify texture patterns from multispectral Landsat TM satellite images using optimized clustering and probabilistic inference. After the images are preprocessed with Principal Component Analysis and decomposed into regions of interest, Gabor wavelets are computed for each region in the first component image to obtain texture feature vectors. An adapted *k*-means clustering algorithm with optimized number of clusters and initial starting centers generates training and testing data for Bayes Point Machine classifiers. The classifiers may run in the online mode for binary classification and the batch mode for multi-class classification. The experimental results show the effectiveness of the proposed classification method and its potentials in other image texture pattern recognition applications.

Index Terms**—**Texture analysis, clustering, Bayesian inference, remote sensing.

I. Introduction

Texture is an important element for pattern recognition and interpretation in the human visual system. In a digital image, it represents the visual impression of smoothness or coarseness produced by the uniformity or variability of image color or tone [1]. Without universally accepted mathematical definition of texture, traditional approaches use symbolic descriptions, such as coarse and fine, rough and smooth, linear and nonlinear, low and high density, and orientation and directionality, etc., to represent texture in images [2]. However, descriptive terms for texture patterns presented in remote sensing imagery, usually generated by multispectral or hyperspectral sensors with various resolution, are yet to be established.

Texture analysis tools have been developed to extract features from remote sensing images with different criteria. The most frequently used pattern recognition techniques are built on statistical methods, such as spatial co-occurrence matrix and variogram. For instance, texture features are identified with a moving window to show the connection between a pixel and its neighbor according to a gray-level co-occurrence matrix (GLCM) [3]. Variogram analysis has been reported to be superior to GLCM for distinguishing very similar texture patches [4]. Algorithms based on the frequency transformation, such as discrete cosine transforms [5] and wavelet transforms [6] have achieved good performance in multi-resolution texture analysis [7].

Texture features extracted from remote sensing images often offer complementary information for applications in which the spectral information of the images alone might not be sufficient for classifying spectrally heterogeneous ground cover and ground use classes [8]. Unsupervised classification techniques, such as morphology [9] and clustering [10], have been used to identify similar texture patterns in remote sensing images, with each pattern represented by a class that is labeled by domain experts in the post-clustering process. Another family of texture classification techniques have been developed from supervised classification methods, such as Bayesian classifier [11] and decision trees [12] when training data is available. Furthermore, Hybrid texture classification [13] involving multi-stage classification methods has demonstrated favorable results.

Extending this line of research, this study develops an application to extract texture features from remotely sensed images and classify the texture patterns with a new approach that integrates wavelet transformation, clustering, and probabilistic inference (See Fig. 1). To obtain the texture features, the Principal Component Analysis (PCA) [14] is performed on a multispectral Landsat TM image, with the first PCA component divided into regions of interest. The texture feature vector of each region is then represented by Gabor wavelets. The classification can be done in either online mode or batch mode. The online mode in which the feature vectors are directly fed into a binary Bayes Point Machine (BPM) [15] classifier allows the user to perform binary classification with dynamically created training data, while the batch mode supports multi-class classification, given the patterns of interest as well as training and testing data for the BPM classifier generated from an optimized *k*-means clustering method.

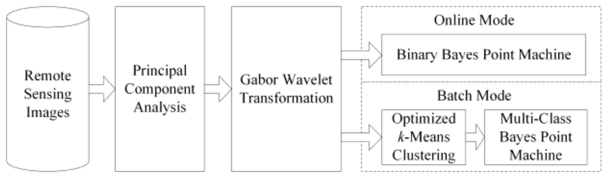

Remote Sensing Images
Principal Component Analysis
Gabor Wavelet Transformation
Online Mode
Binary Bayes Point Machine
Batch Mode
Optimized k-Means Clustering
Multi-Class Bayes Point Machine

Fig.1. Remote sensing image texture extraction and classification.

The rest of the paper is organized as follows. Previous work related to this study is listed in Section II. Section III introduces the process of texture feature extraction by applying PCA to multi-spectral remotely sensed images and computing Gabor wavelets. The classification process that incorporates an optimized *k*-means clustering algorithm and a BPM classifier is presented in section IV. Section V deals with the implementation and the experiments, Section VI discusses the results in details, and section VII concludes with proposals for future work.

II. Related Work

Ref [16] compares the performance of various methods for multi-class texture image classification, such as Naïve Bayes classifier, *k*-nearest neighbor classifier and Neural Network classifier, given features represented by Haar wavelet. The results indicate the efficiency of using wavelet in multi-class texture image classification. In addition, a new texture classification method based on texton features, which evaluate the relationship between the values of neighboring pixels, has achieved better performance than existing methods on various stone textures [17].

Among unsupervised clustering algorithms, *k*-means clustering is popular due to its implementation simplicity and local-minimum convergence [18]. However, *k*-means clustering suffers some intrinsic deficiencies besides its expensive computation that requires multiple data scans to achieve convergence. For instance, it is very sensitive to initial starting conditions, i.e., *k*-means clustering is fully deterministic given the randomly or arbitrarily chosen initial centers. In addition, *k*-means clustering requires a parameter of the number of clusters, which is difficulty to obtain when no prior knowledge about the data is available. Although a general solution does not exist, various approaches have been proposed as partial remedies. For example, embedding the data set in a multi-resolution *kd*-tree and storing sufficient statistics at its nodes may help to improve the speed of convergence [19]. A validity measure of clusters using intra-cluster and inter-cluster distance can be used to estimate the number of clusters [20]. Additionally, repeated sub-sampling and smoothing methods have been developed to refine the starting centers [21].

Recent advance in statistics learning with kernel methods has generated some powerful non-linear classification algorithms, such as Support Vector Machines (SVM) [22] and Bayes Point Machines [15]. SVM maximizes the margin between positive and negative training examples, the so-called support vectors, to find an optimal decision boundary, while BPM approximates the Bayes-optimal probabilities through the mass center of version space. These algorithms have performed well in the applications of content-based image retrieval [23], image classification [24, 25], and remote sensing image information mining [8].

III. Feature Extraction

A. Principal Component Analysis for Satellite Imagery

Principal Component Analysis (PCA) is a coordinate transformation frequently used to reduce the correlation contained within a data set [14]. It has been applied to remote sensing image analysis to take out the correlation contained within the multispectral imagery by creating a new set of components, which are usually more interpretable than the original images. Ref. [26] provided a review of the applications of PCA in remote sensing, such as correlation analysis, change detection, and pattern recognition with multi-temporal Landsat TM images.

Assume N is the number of bands in a multispectral (multiband) remote sensing image, a pixel vector whose components are the individual spectral responses at a pixel location in each band can be characterized as $\mathbf{x}_k = [x_1, x_2, \ldots x_N]^T$. PCA transformation computes the covariance matrix of the original data [26]

$$\mathrm{C_x} = \frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T \tag{1}$$

where K is the total number of pixels and $\mathbf{m} = (1/K)\sum_{k=1}^{K}\mathbf{x}_k$ is the mean pixel vector. Given an identity matrix $\mathbf{I}$, the eigenvalues λ_k are computed via $|\mathbf{C} - \lambda_k \mathbf{I}| = 0$ and the corresponding N-dimensional eigenvectors $\mathbf{e}_k$ that describe component axes are obtained from $(\mathbf{C} - \lambda_k\, \mathbf{I})\mathbf{e}_k = \mathbf{0}$. As a result, the principal components are calculated by $\mathbf{y}_k = \mathbf{G}^T\,\mathbf{x}_k$ where $\mathbf{G}^T$ is the $N \times N$ orthonormal transformation matrix [26]

$$\mathbf{G}^T = [\mathrm{e}_1,\ \mathrm{e}_2, \ldots,\ \mathrm{e}_N]^T = \begin{bmatrix} \mathrm{e}_{11} & \cdots & \mathrm{e}_{N1} \\ \vdots & & \vdots \\ \mathrm{e}_{1N} & \cdots & \mathrm{e}_{NN} \end{bmatrix}^T \tag{2}$$

Each column in the transformation matrix represents the weights applied to the pixels in the corresponding band to create a principal component. The covariance matrix $\mathbf{C}_y$ of the transformed data becomes a diagonal matrix of which the elements are composed of the eigenvalues, while the transformed data points are linear combinations of their original data values weighted by the eigenvectors [26]. PCA component images built this way are uncorrelated and ordered by decreasing variance. The first component image, which has the largest percentage of the total variance and the highest signal-to-noise ratio, is used for texture feature representation.

B. Gabor Wavelet Transformation

Ref. [27] proposed a method for texture feature extraction based on Gabor wavelets, which may achieve the best overall performance compared with other multiresolution texture features using the Brodatz texture database. In addition, their experimental results on large aerial photographs indicate that Gabor wavelets give good accuracy of pattern retrieval [7].

A Gabor function and its Fourier transform are defined as [6]

$$g(x,y)=\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \quad (3)$$

$$G(u,v)=\exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} \quad (4)$$

Gabor wavelets, which form a self-similar filter dictionary, can be obtained by rotation and dilation of the Gabor function [27]

$$\begin{aligned} g_{mn}(x,y)&=a^{-m}g(x',y'),\ a>1\\ x'&=a^{-m}(x\cos n\pi/K, y\sin n\pi/K)\\ y'&=a^{-m}(-x\sin n\pi/K, y\cos n\pi/K) \end{aligned} \quad (5)$$

where $m=0,1,\ldots,S-1$ and $n=0,1,\ldots,K-1$. S is the number of scales and K is the number of orientations in the multiresolution decomposition. The scale factor a^{-m} normalizes the filter responses so that the energy is independent of *m*. Gabor filters are considered as orientation and scale tunable edge and line detectors, and the statistics of the filtered outputs can, therefore, be used to characterize the underlying texture information.

Gabor wavelet transform of an image $I(x,y)$ is defined as [28]

$$W_{mn}(x,y)=\int I(x_1,y_1)g_{mn}^{*}(x-x_1,y-y_1)dx_1dy_1 \quad (6)$$

where * indicates the complex conjugate. Suppose the local texture regions are spatially homogeneous, the mean and the standard deviation of the magnitude of the transform coefficients may represent the region for retrieval and classification [8, 28].

$$\mu_{mn}(x,y)=\iint\left|W_{mn}(x,y)\right|dxdy \quad (7)$$

$$\sigma_{mn}(x,y)=\sqrt{\iint\left(\left|W_{mn}(x,y)\right|-\mu_{mn}\right)^2dxdy} \quad (8)$$

For example, using three scales *S* = 3 and four orientations *K* = 4, a feature vector can be represented as

$$\mathbf{f}=\left[\mu_{00},\sigma_{00},\mu_{01},\sigma_{01},\cdots\mu_{23},\sigma_{23}\right], \quad (9)$$

while the distance between images *i* and *j* in the feature space is [8]

$$d(i,j)=\sum_m\sum_n d_{mn}(i,j)=\sum_m\sum_n\left|\frac{\mu_{mn}^{(i)}-\mu_{mn}^{(j)}}{\alpha(\mu_{mn})}\right|+\left|\frac{\sigma_{mn}^{(i)}-\sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})}\right| \quad (10)$$

with the individual feature components normalized by $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$, the standard deviations of the respective features over the entire set of images.

IV. Classification Methods

A. Optimized k-means Clustering

An empirical comparison of initialization methods for *k*-means clustering was presented in Ref. [19]. This study adopts an optimized *k*-means clustering procedure [8] that combines the validity measure [20], which estimates the number of clusters, with initial centers refinement [21] based on repeated sub-sampling and smoothing.

Suppose *N* is the total number of pixels, *K* is the number of clusters, and **x** represents a texture feature vector, let $\mathbf{m}_i$ be the vector representing the center of cluster C_i, the validity of clusters is defined as M_{intra}/M_{inter}, where the intra-cluster distance M_{intra} is the average of the sum of distance between each vector and its cluster center, while the inter-cluster distance M_{inter} is the minimum distance between any two cluster centers [20].

$$M_{intra}=\frac{1}{N}\sum_{i=1}^{K}\sum_{x\in C_i}\left\|\mathbf{x}-\mathbf{m}_i\right\|^2 \quad (11)$$

$$M_{inter}=\min\left\|\mathbf{m}_i-\mathbf{m}_j\right\|^2, i=1,2,\ldots,K-1,\quad j=i+1,\ldots K \quad (12)$$

Assume $K_{\max}$ is the upper limit of the number of clusters presented in a data set, for each integer *k* where $2\le k\le K_{\max}$, the optimal number of clusters is found by clustering that yields clusters with the minimum validity.

The initial center refinement algorithm is briefly described as follows [8, 21]. Given randomly selected sub-samples S_i, *i* = 1, 2,…, *J*, *k*-means clustering may produce estimates of the true cluster centers $\mathbf{CM}_i$. A distortion value is computed as the sum of square distances of each data point to its nearest center. The centers with minimal distortion value over each sub-sample set are chosen as the refined initial centers. The algorithm also checks the solution at termination for empty clusters and sets the initial estimates of these empty cluster centers to data points that are farthest from their assigned cluster center.

Algorithm 1 Optimized *k*-means clustering

```
Input: $K$, $J$, a dataset $S = \{\mathbf{x} \in X\}$
Output: $K_{opt}$, final hypothesis $\mathbf{C}$
Begin
  for $i = 1 \dots J$ do
    Draw a random subset $S_i$ of $S$
    for $k = 2 \dots K$ do
      $K_{opt}^i = k$ with $\mathbf{min} \frac{M_{k,Intra}^i}{M_{k,Inter}^i}$
    Run $k$-means clustering on $S_i$ with $K_{opt}^i$
        to produce clusters $\mathbf{C}_i$ with centers $\mathbf{CM}_i$
  if $\mathbf{C}_i$ is empty
      $\mathbf{CM}_{opt} = \mathbf{x}$ with $\mathbf{\max}_{\mathbf{x} \in \mathbf{C}_j} \left\{ \left\| \mathbf{x} - \mathbf{CM}_j \right\|^2, j \neq i \right\}$
  else
      $\mathbf{CM}_{opt} = \mathbf{CM}_i$ with $\mathbf{min} \left\{ \sum_{\mathbf{x} \in \mathbf{C}_i} \left\| \mathbf{x} - \mathbf{CM}_i \right\|^2 \right\}$
    Run $k$-means clustering with $K_{opt} = \frac{1}{J} \sum_{i=1}^{J} K_{opt}^i$ and
        $\mathbf{CM}_{opt}$ on $S$ to output final hypothesis $\mathbf{C}$
End
```

The optimization procedure runs the cluster validation algorithm to find the optimal number of clusters K_{opt}^i for each sub-sample data set S_i. The initial center refinement algorithm then uses K_{opt}^i to select optimized starting centers. The final clusters are generated by *k*-means clustering with K_{opt} which is the average of K_{opt}^i [8].

B. Bayes Point Machine Classification

A Bayes Point Machine (BPM) uses a hypothesis function $h(\mathbf{x})$ to classify an input vector $\mathbf{x}$ by finding the inner product of $\mathbf{x}$ with a weight vector $\mathbf{w}$, yielding the output y as true if the inner product $\mathbf{w} \bullet \mathbf{x}$ is positive and false otherwise [15]. Given training data $\mathbf{z} = (\mathbf{x}, \mathbf{y}) = (\{x_1, y_1\} \dots \{x_m, y_m\})$ of size m, the hypothesis functions H form a version space $V(\mathbf{z})$, with each point in V representing a possible classifier.

$$V(z) := \{h \in H \mid \forall i \in \{1,\dots,m\} : h(x_i) = y_i\} \tag{13}$$

Suppose y' is the true output, a zero-one loss function may quantify the cost of predicting y

$$l_{0-1}(y, y') = \begin{cases} 0, & y = y' \\ 1, & y \neq y' \end{cases} \tag{14}$$

The Bayes classification of test data $h(x_i) = y_i$ aims to minimize the loss [29]

$$Bayes_z(x) := \arg\min_{y \in Y} \mathbf{E}_{\mathrm{H}|Z^m = z}\left[l(\mathrm{H}(x), y)\right] \tag{15}$$

where the posterior probability is

$$\mathbf{P}_{\mathrm{H}|Z^m=z}(h) = \begin{cases} \dfrac{\mathbf{P}_{\mathrm{H}}(h)}{\mathbf{P}_{\mathrm{H}}(V(z))}, & h \in V(z) \\ 0, & h \notin V(z) \end{cases} \tag{16}$$

given any prior belief $\mathbf{P}_{\mathrm{H}}(h)$. However, if the training data is limited, the optimal solution $Bayes_z(x)$ may not be any single classifier $h \in H$ [29]. An approximation named Bayes point can be obtained through

$$h_{bp} = Bayes_{bp}(z) := \arg\min_{h \in H} \mathbf{E}_{\mathrm{X}}\left[\mathbf{E}_{\mathrm{H}|Z^m=z}\left[l(h(x), \mathrm{H}(x))\right]\right] \tag{17}$$

which implies the classifier $h_{bp} \in H$ is the best approximation of the optimal Bayes classifier on average over randomly sampled testing data [15]. For linear classifier, a hypothesis $h(x) \in H$ is defined by its weight vector $\mathbf{w}$ and therefore, h_{bp} can be defined by $\mathbf{w}_{bp}$.

$$H := \{x \mapsto \mathrm{sign}(\langle \mathbf{x}, \mathbf{w} \rangle), \ \|\mathbf{w}\| = 1\} \tag{18}$$

When the input distribution is spherically Gaussian,

$$\mathrm{f}_{\mathbf{x}}(\mathbf{x}) = \frac{\exp\left(-\|\mathbf{x}\|^2\right)}{\sqrt{\pi^d}} \tag{19}$$

where d is the dimensionality of the feature vectors, the center-of-mass $\mathbf{w}_{\mathbf{cm}}$ in V can be a good approximation to the Bayes point [29]

$$\mathbf{w}_{\mathrm{cm}} = \frac{\mathbf{E}_{\mathbf{W}|Z^m=z}[\mathbf{W}]}{\left\|\mathbf{E}_{\mathbf{W}|Z^m=z}[\mathbf{W}]\right\|} \tag{20}$$

Compared with other kernel based classifiers, such as SVM which aims to find the center of the largest sphere embedded in the version space *V*, with radius of the sphere being the maximal margin between positive and negative support vectors, the BPM looks for classification at the optimal center, i.e., the Bayes point, of the entire *V*. Therefore, SVM is an approximation to and theoretically less accurate than BPM [15].

a. Binary BPM classification

The design of binary BPM classification is based on the Infer.NET [30], a framework by Microsoft Research for probabilistic programming and running Bayesian inference in graphical models. Besides supporting the popular C# language, Infer.NET class library can be

integrated with other programming languages on Microsoft .NET, such as F#, for parallel computing.
Given the training data that consists of m feature vectors and corresponding observed class labels $D = \{(\mathbf{x}_i, y_i), i = 1,\dots,m\}$, the BPM classification aims to find the predictive distribution $\{p(y = c|\mathbf{x}, D), c = 1,\dots, K\}$ over K classes, conditional on $\mathbf{x}$ and D [30]. Assume a parametric density of the form $p(y|\mathbf{x},\mathbf{w}) = p(y|s = \mathbf{w}^T\mathbf{x})$ where s is the instance score, Fig. 2 illustrates that, the training of a binary BPM classifier learns a posterior distribution over the weight **w**, which is used in the predictions on testing data [30].

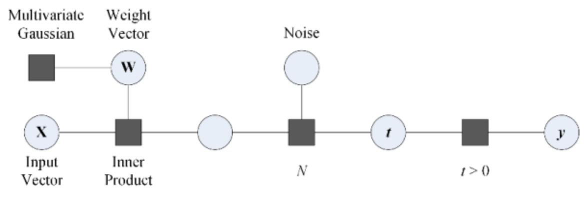


Fig.2. Factor graph of a binary Bayes Point Machine classifier (adapted from Ref [30] Minka, Winn, Guiver and Knowles).

Assume **w** is a vector with a multivariate Gaussian prior distribution, Gaussian noise is added to the score to allow for measurement and labelling error. Because if the training data is not linearly separable, which is common for kernel based classifiers, then no setting of **w** exists to classify the training data into two classes precisely. For each data sample during the training, variable t is computed by the inner product of **w** with the corresponding feature vector **x** in addition to the noise. The output variable y yields a value of true if $t > 0$ or false if $t < 0$.

b. *Multi-class BPM classification*

The multi-class BPM classification is similar to the binary BPM classification. However, instead of using a single linear discriminant function, it has K functions with their respective weight vectors $\mathbf{w}_c$ and scores s_c, one for each class c in {1,...,K} [30]. The maximum score decides the class y to which a feature vector belongs.

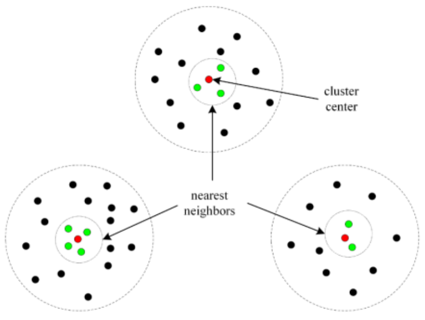


Fig.3. Training and testing data generated from nearest neighbors of the cluster centers.

The number of classes depends on the number of clusters generated from the optimized k-means clustering. Remote sensing domain experts may choose only the classes they are interested in. In addition, the training and testing texture feature vectors labeled by domain experts for multi-class BPM classification are built from the nearest neighbors of the cluster centers (See Fig. 3).

Assume M is the number of feature vectors in a cluster, the cluster center is computed as the average of all the feature vectors in that cluster.

$$\mathbf{F}_C = \frac{1}{M}\sum_{p=1}^{M}\mathbf{F}_p = \frac{1}{M}\sum_{p=1}^{M}\left[\mu_{00}^p\sigma_{00}^p\mu_{01}^p\sigma_{01}^p\cdots\mu_{IJ}^p\sigma_{IJ}^p\right] \tag{21}$$

The nearest neighbors of a cluster center are obtained by sorting the feature vectors according to their distance, e.g. the Euclidean distance, to the cluster center. The Euclidean distance between a feature vector $\mathbf{F}_q$ and the cluster center is defined as

$$d(q,C) = \left\|\mathbf{F}_q - \mathbf{F}_C\right\|^2 = \sqrt{\sum\nolimits_{i,j}[(\mu_{ij}^q - \mu_{ij}^C)^2 + (\sigma_{ij}^q - \sigma_{ij}^C)^2]} \tag{22}$$

Algorithm 2 Multi-class BPM Classification

```
Training
Input: A set of clusters
    $C = \{c_i, (\mathbf{x}_{i,j}, y_i) \in c_i, i = 1,\dots,K,\ j = 1,\dots,M_i\}$
Output: hypothesis $H = \{h_i, i = 1,\dots,K\}$
Begin
  for $i = 1 \dots K$ do
    for $j = 1 \dots M_i$ do
      $d_{i,j} = \left\|\mathbf{x}_{i,j} - \mathbf{cm}_i\right\|^2$
    Sort $\{d_{i,j}, \mathbf{x}_{i,j}, j = 1,\dots,M_i\}$
    Label $(\mathbf{X}_i, y_i) = \{(\mathbf{x}_{i,j}, y_i), j = 1,\dots,N_i\}$ where
        $N_i = \lceil\alpha \times M_i\rceil, \alpha \in [0,\ 1]$
   Train BPM with $(\mathbf{X}_i, y_i)$ to produce hypothesis
        $H = \{h_i, i = 1,\dots,K\}$
End

Testing
Input: A data set of
      $D = \{\mathbf{x}_{i,j} \in c_i, i = 1,\dots,K,\ j = N_i + 1,\dots,M_i\}$
      hypothesis $H = \{h_i, i = 1,\dots,K\}$
Output: Labeled data set
      $D_L = \{D_l : (\mathbf{x}_{l,s}, y_l), l = 1,\dots,K, s = 1,\dots,N_l\}$

Begin
 Ensemble $D = \{\mathbf{x}_{i,j} \in c_i, i = 1,\dots,K,\ j = N_i + 1,\dots,M_i\}$
 to produce $D_T = \{\mathbf{x}_t, t = 1,\dots,N\}$ where
        $N = \sum_{i=1}^{K}(M_i - N_i)$
 Run $H$ on $D_T$ to produce probabilities
    $P(\mathbf{X}) = \{p_l(\mathbf{x}_t), l = 1,\dots,K, t = 1,\dots,N\}$
 for $t = 1 \dots N$ do
```

```
    if $p_l(\mathbf{x}_t)$ = max$\{p_j(\mathbf{x}_t), j = 1,...,l,...K\}$ then
        $N_l = N_l + 1$,
        $s = N_l$,
        $\mathbf{x}_{l,s} = \mathbf{x}_t$, $y_l = l$
        Add $(\mathbf{x}_{l,s}, y_l)$ to class $D_l$
    $D_L = \{D_l : (\mathbf{x}_{l,s}, y_l), l = 1,..., K, s = 1,..., N_l\}$
End
```

V. Implementation and Experiments

A. Image Selection and Preprocessing

The experiments were conducted on a geometric rectified Landsat TM image, which has a 30 meter by 30 meter spatial resolution for six of its seven bands. The PCA did not use the thermal-infrared band 6 because of its much worse spatial resolution of 120 meter by 120 meter. Fig. 4 shows the first PCA component image that consists of 4096 × 4096 pixels, covering the scene of middle Tennessee, USA. This image was then divided into 32 × 32 = 1024 regions of 128 × 128 pixels each.

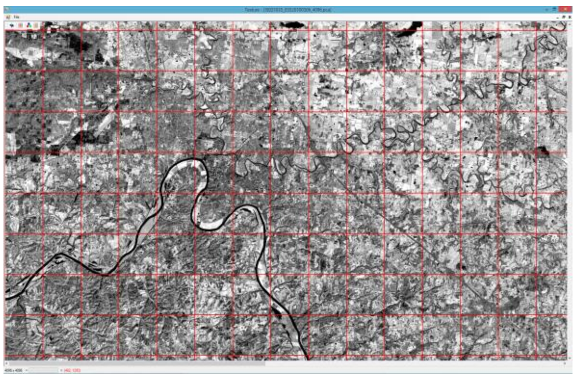


Fig.4. A Landsat TM image covering the scene of middle Tennessee (first PCA component).

For Landsat TM images, smaller region size may not cover sufficient spatial/texture information to characterize land use types, while a large region size may involve too much information from other types.

B. Feature Vector Construction and Clustering

The Gabor wavelet feature vectors were computed using a C++ Dynamic Link Library (DLL) and imported to the optimized *k*-means clustering application developed with C# .NET. The filter parameters for Gabor wavelet were *S* = 3, and *K* = 4. The indices of the regions range from [0, 0] to [31, 31] for a total of 1024 regions. The results of the optimized *k*-means clustering indicated the optimal number of clusters in the given image is 6. Fig. 5 shows the corresponding texture patterns identified by the clustering and labeled by a remote sensing domain expert.

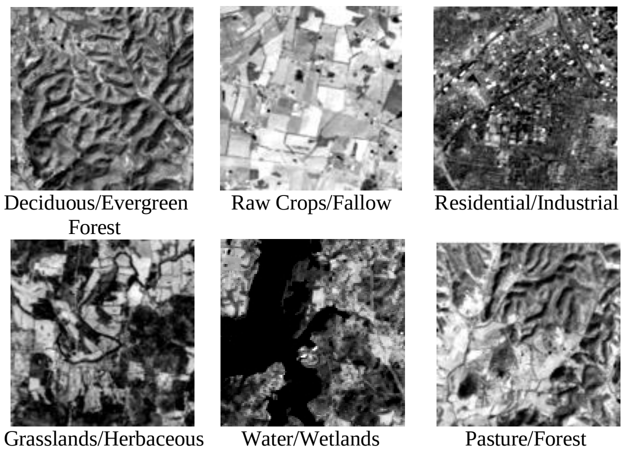


Fig.5. Texture patterns identified by the optimized *k*-means clustering.

C. Binary Classification in the Online Mode

Based on the Image Classifier example of the Infer.NET [30], a binary BPM classifier running in the online mode was implemented to categorize only one of these texture patterns at a time. The experimental task for the binary classification was to find regions containing texture pattern of Water/Wetlands [31].

Fig. 6 demonstrates the classification process that involves user actions of dragging and dropping a region to the box for the corresponding class, which triggers an event handler to activate the training model of the binary BPM classifier. The rest of regions are categorized into two classes by the testing model of the BPM classifier. This classification process is animated by automatically moving a region to the side of the class it belongs to. How far to the left or right the region appears is a measure of the degree to which the binary BPM classifier believes it belongs to that class. It can be observed that more regions with Water/Wetlands are aligned to the left given more training images, and in the end, the majority of Water/Wetlands regions are correctly classified.

The binary BPM classification application has indicated the potential of the BPM classifier via an interactive visual presentation. However, quantitative measurements are desired according to the characteristics and size of the training and testing image set, as well as the classification accuracy [31], which is reported in the next section based on the experiments of the multi-class BPM classification.

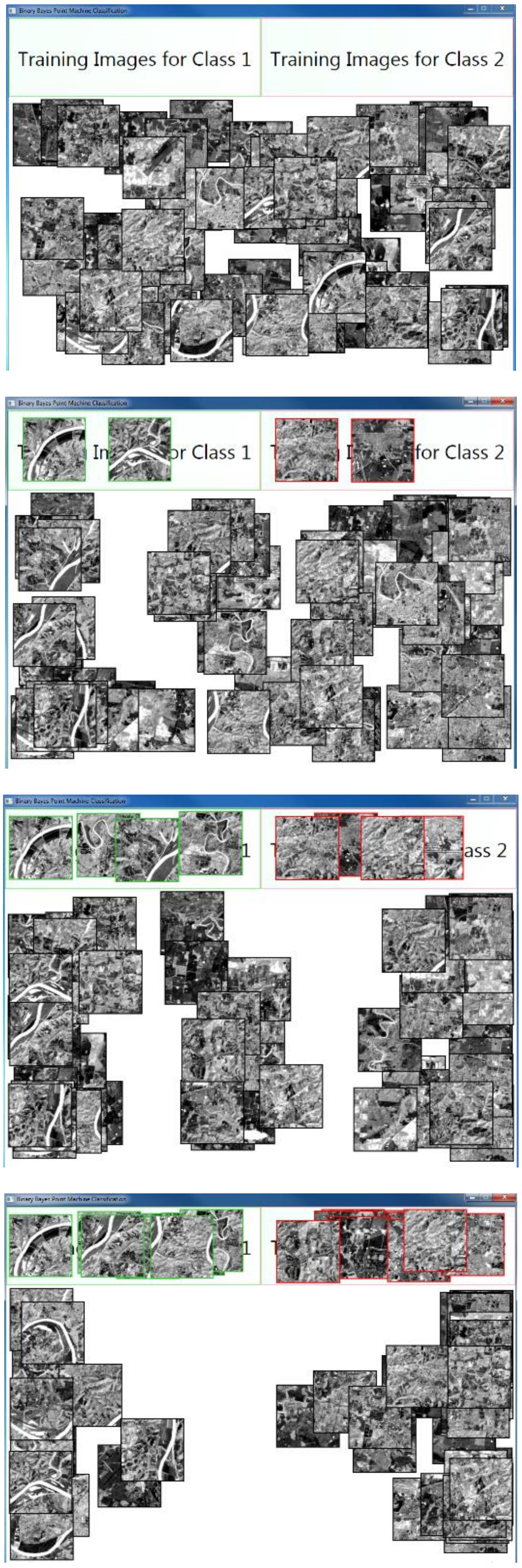

Binary Bayes Point Machine Classification
Training Images for Class 1
Training Images for Class 2

Fig.6. An example of training and testing of the binary BPM classifier.

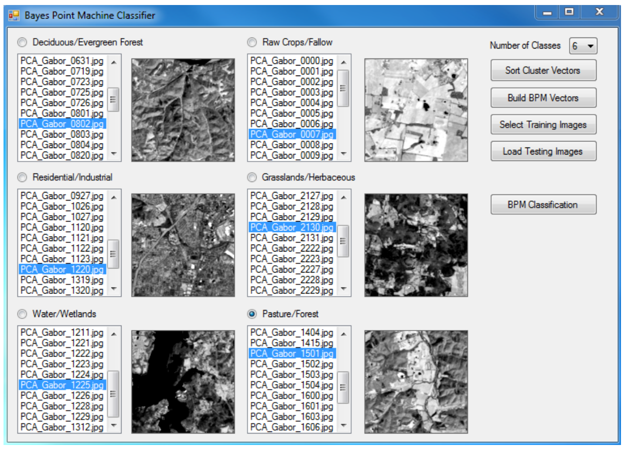
D. *Multi-class Classification in the Batch Mode*

The multi-class classification was designed to run in the batch mode which takes a set of training images for the BPM classifier. Table 1 shows the total number of images labeled for each class based on the nearest neighbors of the cluster centers.

Table 1. Number of Images Labeled for Each Texture Class

Texture Class	Number of Images labeled
Deciduous/Evergreen Forest (D/EF)	150
Raw Crops/Fallow (RC/F)	110
Residential/Industrial (R/I)	130
Grasslands/Herbaceous (G/H)	110
Water/Wetlands (W/W)	80
Pasture/Forest (P/F)	120

Training image selection was done through an interactive user interface that allows the remote sensing domain experts to build a list of feature vectors with the associated regions (See Fig. 7). In addition, it offers a preview picture box to display the region when the user clicks on the file name in the list box. The testing results using 5-fold cross validation and 10-fold cross validation are presented in the following section.



Number of Classes 6

Deciduous/Evergreen Forest: PCA_Gabor_0631.jpg, PCA_Gabor_0719.jpg, PCA_Gabor_0723.jpg, PCA_Gabor_0725.jpg, PCA_Gabor_0726.jpg, PCA_Gabor_0801.jpg, PCA_Gabor_0802.jpg, PCA_Gabor_0803.jpg, PCA_Gabor_0804.jpg, PCA_Gabor_0820.jpg

Raw Crops/Fallow: PCA_Gabor_0000.jpg, PCA_Gabor_0001.jpg, PCA_Gabor_0002.jpg, PCA_Gabor_0003.jpg, PCA_Gabor_0004.jpg, PCA_Gabor_0005.jpg, PCA_Gabor_0006.jpg, PCA_Gabor_0007.jpg, PCA_Gabor_0008.jpg, PCA_Gabor_0009.jpg

Residential/Industrial: PCA_Gabor_0927.jpg, PCA_Gabor_1026.jpg, PCA_Gabor_1027.jpg, PCA_Gabor_1120.jpg, PCA_Gabor_1121.jpg, PCA_Gabor_1122.jpg, PCA_Gabor_1123.jpg, PCA_Gabor_1220.jpg, PCA_Gabor_1319.jpg, PCA_Gabor_1320.jpg

Grasslands/Herbaceous: PCA_Gabor_2127.jpg, PCA_Gabor_2128.jpg, PCA_Gabor_2129.jpg, PCA_Gabor_2130.jpg, PCA_Gabor_2131.jpg, PCA_Gabor_2222.jpg, PCA_Gabor_2223.jpg, PCA_Gabor_2227.jpg, PCA_Gabor_2228.jpg, PCA_Gabor_2229.jpg

Water/Wetlands: PCA_Gabor_1211.jpg, PCA_Gabor_1221.jpg, PCA_Gabor_1222.jpg, PCA_Gabor_1223.jpg, PCA_Gabor_1224.jpg, PCA_Gabor_1225.jpg, PCA_Gabor_1226.jpg, PCA_Gabor_1228.jpg, PCA_Gabor_1229.jpg, PCA_Gabor_1312.jpg

Pasture/Forest: PCA_Gabor_1404.jpg, PCA_Gabor_1415.jpg, PCA_Gabor_1501.jpg, PCA_Gabor_1502.jpg, PCA_Gabor_1503.jpg, PCA_Gabor_1504.jpg, PCA_Gabor_1600.jpg, PCA_Gabor_1601.jpg, PCA_Gabor_1603.jpg, PCA_Gabor_1606.jpg

Fig.7. Training image selection for multi-class BPM classification.

VI. Results

A. *Classification Accuracy*

Fig. 8 (a) demonstrates classification accuracy of each partition using 5-fold cross validation while Fig. 8 (b) shows classification accuracy of each partition using 10-fold cross validation. It can be observed that the accuracy of Grasslands/Herbaceous (G/H) texture class is most consistent in each partition, while the accuracy of Water/Wetlands (W/W) texture class varies most significantly.

Meanwhile, the consistency of average classification accuracy of each texture class over all the partitions using 5-fold cross validation and 10-fold cross validation (see Fig. 9) indicates the effectiveness and robustness of the BPM classifier in multi-class classification.

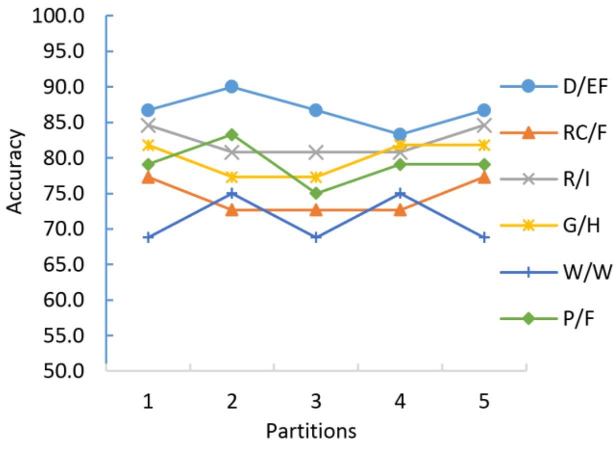


Fig.8. (a) Classification accuracy using 5-fold cross validation.

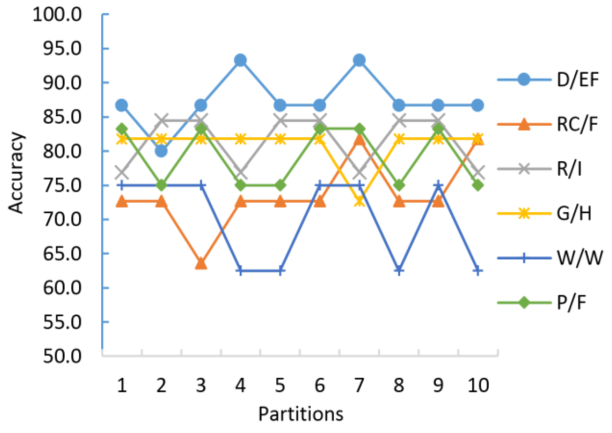


Fig.8. (b) Classification accuracy using 10-fold cross validation.

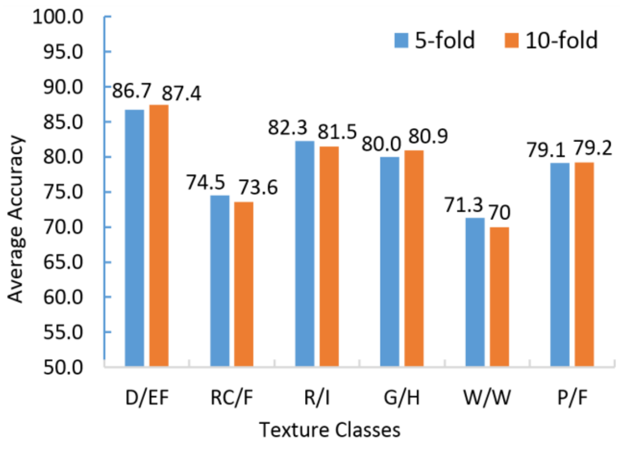


Fig.9. Average accuracy using 5-fold and 10-fold cross validation.

B. *Comparing the BPM classifier with the SVM classifier*

Additionally, using the same set of training and testing data, Fig. 10 (a) and (b) indicate the average accuracy of the BPM classifier compared with that of the SVM classifier implemented using LIBSVM [32] with 5-fold and 10-fold cross validation respectively. Overall, the BPM classifier achieved better results than the SVM classifier for all the classes except for Water/Wetlands, which is in line with the study that SVM is an approximation to and theoretically less accurate than BPM [15, 29].

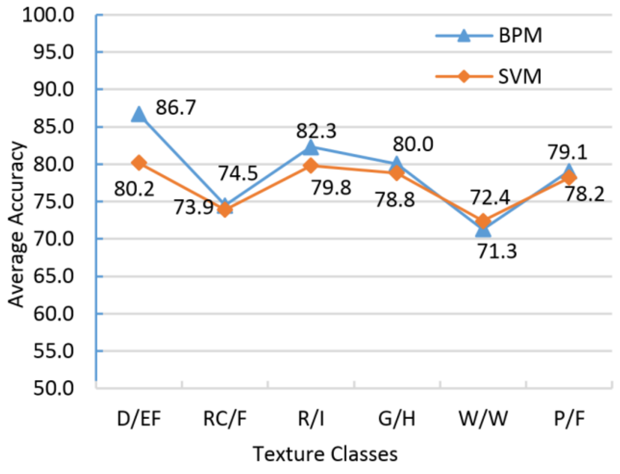


Fig.10. (a) BPM vs. SVM using 5-fold cross validation.

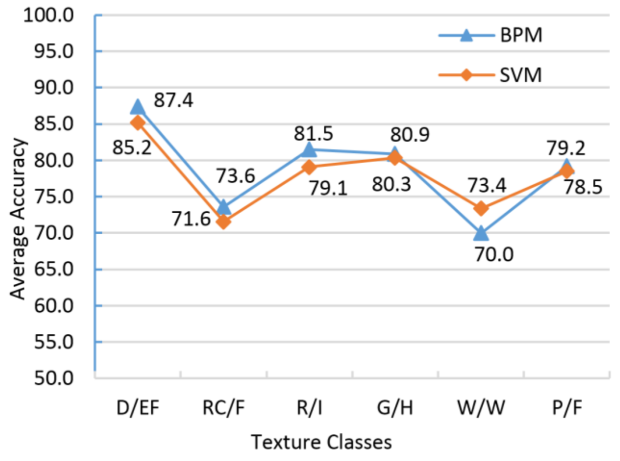


Fig.10. (b) BPM vs. SVM using 10-fold cross validation.

The limited data size of the Water/Wetlands texture pattern may have caused the relatively low classification accuracy and led to this inconsistent performance. Meanwhile, the parameters of BPM and SVM classifiers were chosen based on the successful experience of previous studies [8, 31], which can be further adjusted and optimized.

VII. Conclusion

This study proposes a new approach for representing and extracting the texture features in remotely sensed images using PCA and Gabor Wavelets, and categorizing the texture patterns based on the BPM classifier and the adapted *k*-means clustering with estimated number of clusters and optimized initial starting centers. In particular, the training and testing data for the multi-class BPM classifier are obtained from the nearest neighbors of the cluster centers generated by the optimized *k*-means clustering, a unique process that may help the domain experts to select the texture patterns of interest and the corresponding training and testing regions in a large collection of images. The BPM classifier, when running in the online mode, provides the user an interactive visualization tool for binary classification. The experimental results of the multi-class BPM classifier running in the batch mode indicate overall good performance compared with the SVM classifier.

Our future work includes (1) create a database for better support of image selection and organization, (2) apply the proposed texture feature classification method to other types of satellite images, and (3) implement the algorithms in a parallel computing environment using F# programming language so that it scales well on larger set of images.

ACKNOWLEDGMENT

The authors thank Dr. B. S. Manjunath from University of California - Santa Barbara for sharing the C++ source code for the Gabor wavelet representation. This work is supported by the Scholarly and Creative Fellowship Program and the Presidential Research Scholars Program at Austin Peay State University.

REFERENCES

[1] S. W. Myint, "A robust texture analysis and classification approach for urban land-use and land-cover feature discrimination," *Geocarto Int.*, vol. 16, pp. 27–38, 2001.

[2] P. Maillard, "Comparing texture analysis methods through classification," *Photogrammetric Eng. & Rem. Sens.*, vol. 69, pp. 357–367, 2003.

[3] S. E. Franklin, R. J. Hall, L. M. Moskal, A. J. Maudie and M. B. Lavigne, "Incorporating texture into classification of forest species composition from airborne multispectral images," *Int. J. Rem. Sens.*, vol. 21, pp. 61–79, 2000.

[4] Q. Chen and G. Peng, "Automatic variogram parameter extraction for textural classification of the panchromatic IKONOS imagery," *IEEE Trans. Geo. and Rem. Sens.*, vol. 42, pp. 1106–1115, 2004.

[5] T. R. Reed and J. M. H. Buf, "Review of recent texture segmentation and feature extraction techniques," *Comp. Vision, Image Proc. Graphics*, vol. 57, pp. 359–372, 1993.

[6] M. R. Turner, "Texture transformation by Gabor function," *Bio. Cyber.*, vol. 55, pp. 71–82, 1986.

[7] W. Y. Ma and B. S. Manjunath, "A texture thesaurus for browsing large aerial photographs," *J. Amer. Soc. Info. Sci.*, vol. 49, pp. 633–648, 1998.

[8] J. Li and R. M. Narayanan, "Integrated spectral and spatial information mining in remote sensing imagery," *IEEE Trans. Geo. Rem. Sens.*, vol. 42, 673–685, 2004.

[9] M. Pesaresi and J.A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geo. Rem. Sens.*, vol. 39, pp. 309–320, 2001.

[10] A. Weisberg, M. Najarian, B. Borowski, J. Lisowski and B. Miller, "Spectral angle automatic cluster routine (SAALT): an unsupervised multispectral clustering algorithm," in *Proc. IEEE Aerospace* 1999 (Aspen, CO, 1999), pp. 307–317.

[11] B. Gorte and A. Stein, "Bayesian classification and class area estimation of satellite images using stratification," *IEEE Trans. on Geo. Rem. Sens.*, vol. 36, pp. 803–812, 1998.

[12] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[13] A. Pitiot, A. W. Toga, N. Ayache and P. Thompson. "Texture based MRI segmentation with a two-stage hybrid neural classifier," in *Proc. 2002 Int. Joint Conf. Neural Networks* (Honolulu, HI, 2002), vol. 3, pp. 2053–2058.

[14] S. Wold, E. Kim and G. Paul, "Principal component analysis," *Chemometrics Intell. Lab. Sys.*, vol. 2, 37–52, 1987.

[15] R. Herbrich, T. Graepel and C. Campbell, "Bayes point machines," *J. Mach. Learning Res.*, vol. 1, pp. 245–279, 2001.

[16] D. Choudhary, A. K. Singh, S. Tiwari and V. P. Shukla, "Performance analysis of texture image classification using wavelet feature," *Int. J. Image, Graphics & Sig. Processing* (IJIGSP), vol. 5, no. 1, pp.58–63, 2013. DOI: 10.5815/ijigsp.2013.01.08.

[17] U. Babu, V. V. Kumar and B Sujatha, "Texture classification based on texton features," *Int. J. Image, Graphics & Sig. Processing* (IJIGSP), vol. 4, no. 8, pp.36–42, 2012. DOI: 10.5815/ijigsp.2012.08.05.

[18] J. A. Hartigan and M. A. Wong. "Algorithm AS 136: A *k*-means clustering algorithm," *Appl. Stat.*, pp. 100–108, 1979.

[19] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Y. Wu, "An efficient *k*-means clustering algorithm: analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 881–892, 2002.

[20] S. Ray and R. H. Turi, "Determination of number of clusters in *k*-means clustering and application in color image segmentation," in *Proc. Int. Conf. Advances in Pattern Recogn. Digital Tech.* 1999 (Calcutta, India, 1999), pp. 137–143.

[21] P. S. Bradley and U. M. Fayyad, "Refining initial points for *k*-means clustering," in *Proc. Int. Conf. Mach. Learning*, J. Shavlik, Ed., San Francisco, CA, 1998, pp. 91–99.

[22] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: The Cambridge University Press, 2000.

[23] E. Chang, K. Goh, G. Sychay and G. Wu, "CBSA: Content-based soft annotation for multimodal image retrieval using Bayes point machines," *IEEE Trans. Cir. and Sys. Video Tech.*, vol. 13, pp. 26–38, 2003.

[24] P. Mantero, G. Moser and S. B. Serpico, "Partially supervised classification of remote sensing images through SVM-based probability density estimation," *IEEE Trans. Geo. Rem. Sens.*, vol. 43, pp. 559–570, 2005.

[25] T. Celik and T. Tjahjadi, "Bayesian texture classification and retrieval based on multiscale feature vector," *Pattern Recogn. Let.*, vol. 32, pp.159–167, 2011.

[26] C. Rodarmel and J. Shan, "Principal component analysis for hyperspectral image classification," *Surveying Land Info. Sci.*, vol. 62, pp. 115–122, 2002.

[27] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 837–842, 1996.

[28] B. S. Manjunath, P. Wu, S. Newsam and H. D. Shin, "A texture descriptor for browsing and similarity retrieval," *J. Sig. Proc.: Image Comm.*, vol. 16, pp. 33–43, 2000.

[29] R. Herbrich, T. Graepel and C. Campbell, "Bayes point machines: Estimating the Bayes point in kernel space," In *Proc. IJCAI 1999 Workshop on Support Vector Machines* (Stockholm, Sweden, 1999), pp. 23–27.

[30] T. Minka, J. Winn, J. Guiver and D. Knowles, *Infer.NET 2.6*, Microsoft Research Cambridge, 2014. http://research.microsoft.com/infernet.

[31] J. Li, "Texture classification of Landsat TM imagery using Bayes Point Machine," In *Proc. ACMSE* 2013 (Savannah, GA, 2013), pp. 16.1–16.6.

[32] C-C. Chang and C-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intel. Sys. Tech.*, vol. 2, pp. 27.1–27.27, 2011.

Authors' Profiles

Jiang Li received his Ph.D. in Electrical Engineering from the University of Nebraska – Lincoln, USA in 2003. He also holds B.S. and M.S. degrees in Electronics Engineering from Beijing Institute of Technology, China.

He is a Professor in Computer Science and Information Technology at Austin Peay State University, USA. His research interest is in image processing, machine learning, data mining, software engineering, and distributed computing. He has published over 20 refereed articles in international journals and conference proceedings. Dr. Li is a member of ACM and he currently serves on the ODBMS.ORG's Panel of Experts.

William Rich received the B.S. degree in Computer Science from Austin Peay State University, USA in 2015.

Donald Buhl-Brown received the B.S. degree in Computer Science from Austin Peay State University, USA in 2015.

How to cite this paper: Jiang Li, William Rich, Donald Buhl-Brown,"Texture Analysis of Remote Sensing Imagery with Clustering and Bayesian Inference", IJIGSP, vol.7, no.9, pp.1-10, 2015.DOI: 10.5815/ijigsp.2015.09.01