Relevance-Based Feature Extraction for Hyperspectral Images

Michael J. Mendenhall, *Member, IEEE*, and Erzsébet Merényi, *Senior Member, IEEE*

***Abstract*—Hyperspectral imagery affords researchers all discriminating details needed for fine delineation of many material classes. This delineation is essential for scientific research ranging from geologic to environmental impact studies. In a data mining scenario, one cannot blindly discard information because it can destroy discovery potential. In a supervised classification scenario, however, the preselection of classes presents one with an opportunity to extract a reduced set of meaningful features without degrading classification performance. Given the complex correlations found in hyperspectral data and the potentially large number of classes, meaningful feature extraction is a difficult task. We turn to the recent neural paradigm of generalized relevance learning vector quantization (GRLVQ) [B. Hammer and T. Villmann, *Neural Networks*, vol. 15, pp. 1059–1068, 2002], which is based on, and substantially extends, learning vector quantization (LVQ) [T. Kohonen, *Self-Organizing Maps*, Berlin, Germany: Springer–Verlag, 2001] by learning relevant input dimensions while incorporating classification accuracy in the cost function. By addressing deficiencies in GRLVQ, we produce an improved version, GRLVQI, which is an effective analysis tool for high-dimensional data such as remotely sensed hyperspectral data. With an independent classifier, we show that the spectral features deemed relevant by our improved GRLVQI result in a better classification for a predefined set of surface materials than using all available spectral channels.**

***Index Terms*—Feature extraction, hyperspectral image compression, joint classification and compression, learning vector quantization (LVQ).**

I. Introduction

HYPERSPECTRAL images have hundreds of bands where a single remotely sensed scene can be greater than 100 MB in size. Due to complex correlations, the number of potential clusters, and the intricacy of cluster and subcluster relationships, compounded with the volume of data, classification of hyperspectral imagery is difficult. One way to potentially make the classification problem easier is to do meaningful feature extraction. Successful feature extraction will suppress superfluous signal content while preserving signal information important to maintain the classifier's ability to correctly identify material classes of interest. Processing with a reduced set of features also has the distinct advantage of reducing processing time, storage requirements, and transmission time and bandwidth.

A number of previous works addressed feature extraction versus classification performance for hyperspectral data. Benediktsson *et al.* extract 35 spectral features from 224 bands of a hyperspectral scene using a uniform feature design method coupled with decision boundary feature extraction (DBFE) [3]. Features are cascaded to the input of a backpropagation neural network [3] for classification. Classification accuracy with the 35 features is, in principle, as good as using all available features for the nine material classes they studied. However, they do not support this claim with a benchmark classification using all spectral channels. As a result, the true quality of the extracted features remains unknown. Moon and Merényi select the largest magnitude wavelet coefficients from a remotely sensed hyperspectral scene where the resulting spectral features are used in a hybrid neural network [4] for classification. Although using the largest magnitude wavelet coefficients is a common practice in signal compression, this method of feature selection does not show a clear trend between the number of retained features and the achieved classification accuracy for the 13-class problem they evaluated. This study is backed by an earlier benchmark classification by Merényi *et al.* using the same hybrid neural architecture on all available (158) spectral features [5]. Zhang *et al.* decompose soil spectra using wavelets where the subband energy is used as a feature set [6]. The extracted features are in turn used in a maximum-likelihood (ML) classifier. The degradation of classification accuracy from a three-class problem of major soil texture types to a more difficult 12-class soil texture problem (each major soil texture type having four subclasses) indicates that wavelet subband energy is not an adequate feature set. Each of the works described previously use a different method for feature extraction and different classifiers. They exhibit a common problem: feature extraction is not optimized for classification. Both functions are accomplished independently of each other.

Oehler and Gray [7] note that some joint compression and classification schemes optimize the compression of several different signals (e.g., in speech processing) where the compressor yielding the smallest distortion indicates the class to which the signal belongs. Others optimize for compression first, using the compressed output as the input to a classifier that minimizes the probability of error. Oehler and Gray present a true joint compression and classification system that uses a learning vector quantization (LVQ) [2] algorithm to minimize a distortion function which includes a *squared error* term and a *Bayes risk* term [7], where a parametric model for the posterior class probabilities is required. If posterior class probabilities are unavailable, one must be satisfied with suboptimal results where the distortion function reduces to the squared error term only.

Manuscript received January 17, 2007; revised July 26, 2007; accepted August 23. 2007. The work of M. J. Mendenhall was supported by the Air Force Institute of Technology, Wright-Patterson AFB, OH. The work of E. Merényi was supported in part by the Applied Information Systems Research Program, NASA, Science Mission Directorate under Grant NNG05GA94G.

M. J. Mendenhall is with the Department of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433-7765 USA (e-mail: michael.mendenhall@afit.edu).

E. Merényi is with the Department of Electrical Engineering, Rice University, Houston, TX 77005-1827 USA (e-mail: erzsebet@rice.edu).

Digital Object Identifier 10.1109/TNN.2007.914156

Not all classifiers are well suited for classifying hyperspectral data. Accurate density models for remotely sensed hyperspectral images do not necessarily exist. We are unable to use methods such as that of Oehler and Gray [7] for this reason. Classifiers based on class covariance (e.g., ML) are not viable options because there is seldom enough training samples. This problem is especially severe when classifying rare classes where training samples are scarce. Often in cases where samples are plentiful, covariance methods still do not work well for hyperspectral data (see, e.g., [8] and [9]) for lack of prior probabilities.

Generalized relevance learning vector quantization (GRLVQ) is a recent supervised neural learning paradigm [1] extending the developments by Sato and Yamada [10] on generalized learning vector quantization (GLVQ). One may describe GRLVQ as a classification-driven feature extraction algorithm which discovers those features important for classification. Supervised neural learning paradigms, such as LVQs, are desirable because they are robust to noisy and incomplete data, do not require parametric models, and are optimal in the sense of minimizing the Bayes risk.

II. Generalized Relevance Learning Vector Quantization

LVQ and its variants [2] are supervised neural learning algorithms that are particularly powerful for classifying high-dimensional data sets with complicated class structure. Prototype vectors are the trained quantities in an LVQ that learn a given representation of the class to which they are assigned. During the learning process, an LVQ iteratively adjusts the prototype vectors in a fashion that defines class boundaries while minimizing the Bayes risk. Variants of LVQ based on LVQ2.1 [2], *differentially shift* the decision boundary by adjusting an in-class and an out-of-class prototype vector at each iteration. LVQs belong to a class of maximal-margin algorithms that maximize the hypothesis margin [11], and certain forms of LVQ (GLVQ and GRLVQ, for example) are *gradient–descent* algorithms with the following general weight update form [12]:

$$w(t+1) = w(t) - \epsilon(t)\nabla C(w(t)) \tag{1}$$

where $C(w(t))$ is the function we wish to minimize, $\epsilon(t)$ is the learn parameter, and $w(t)$ is the state of the prototype vector at time t.

GLVQ by Sato and Yamada [10] improves the class boundary approximation of LVQ2.1 by incorporating classification accuracy in the cost function C. Hammer and Villmann extend GLVQ by learning a weighting of the input dimensions for classification. Let us facilitate our discussion of GRLVQ by defining variables similarly as in [1].

- Define the training sample set as $\{x^m, y^m\}_{m=1}^{M} \in \{\Re^n \times \Re\}$. There are M samples x with n dimensions and class labels y.
- Define $\{W\}$ as the set of all prototype vectors and denote by $w^J \in \{W\}$ the best-matching in-class prototype vector with class label y^m, the same as that of the input sample x^m. The number of prototypes for class y^m is P and prototypes in class y^m are indexed by $p = \{1, \ldots, P\}$. Further define w^K as the best-matching out-of-class prototype vector with class label $y^r \neq y^m$.
- Define d^J and d^K as the squared Euclidean distance between the input sample x^m and prototype vectors w^J and w^K, respectively. The notation d^p is the squared Euclidean distance between prototype vector $p = \{1, \ldots, P\}$ with class label y^m, and the sample x^m.
- Define $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ as an n-dimensional vector of relevance factors and Λ as a diagonal matrix with $\Lambda_{ii} = \lambda_i$ where $i = \{1, \ldots, n\}$.
- Define the weighted squared Euclidean distance between the input sample x^m and prototype w^J and w^K as d_λ^J and d_λ^K, respectively, where λ indicates relevance factors used in the Euclidean distance calculation.
- Define $\mu(x^m)$ as the misclassification measure.
- Define $f(\mu(x^m))$ as the loss function.
- Define C as the cost function.

There are two aspects to the gradient–descent problem of (1). The first is to define a *misclassification measure* to represent a correct versus wrong classification. Sato and Yamada define the misclassification measure as

$$\mu(x^m) = \left(\frac{d^J - d^K}{d^J + d^K}\right). \tag{2}$$

In (2), $\mu(x^m)$ is a normalized distance bounded by -1 and 1. This definition has a nice numerical interpretation of how well the current sample was classified. A correct classification occurs if $\mu(x) < 0$ and the sample is classified perfectly if $\mu(x) = -1$. Similarly, a wrong decision is made if $\mu(x) \geq 0$.

The second aspect is to define a differentiable *loss function* that takes into account the misclassification measure $\mu(x)$. Sato and Yamada define the loss function as

$$\begin{aligned} f(\mu(x^m)) &= \frac{1}{1 + e^{-\mu(x^m)}} \\ &= \frac{1}{1 + e^{-\left(\frac{d^J - d^K}{d^J + d^K}\right)}}. \end{aligned} \tag{3}$$

Using the sigmoid function as the loss function [see (3)] has the distinct advantage of having a derivative that is a function of itself

$$\begin{aligned} f'(\mu(x^m)) &= f(\mu(x^m))[1 - f(\mu(x^m))] \\ &= \frac{e^{-\left(\frac{d^J - d^K}{d^J + d^K}\right)}}{\left(1 + e^{-\left(\frac{d^J - d^K}{d^J + d^K}\right)}\right)^2}. \end{aligned} \tag{4}$$

The *cost function* C minimized in (1) is a simple sum of the losses for each sample [10]

$$C = \sum_{m=1}^{M} f(\mu(x^m)). \tag{5}$$

Hammer and Villmann [1] use as the basis for their algorithm the misclassification measure, loss function, and cost function described previously. The behavior of the algorithm is changed if using a different loss function or if changing the misclassification measure (see, e.g., [13] and [14]).

A. Winner Selection in GRLVQ

The winning prototype vector c is the prototype vector selected as

$$c = \arg\min_q \left(\sum_{i=1}^{n} \lambda_i \left(x_i^m - w_i^q \right)^2 \right). \tag{6}$$

Winner selection varies depending on the form of LVQ. For all variants of LVQ, except GRLVQ, prototypes are chosen based on (6), where $\lambda_i = 1$ for $i = 1, \ldots, n$. That is, each dimension is equally weighted. In LVQ1, the single prototype globally minimizing the distortion in (6) is selected as the winner. For LVQ3, winner selection chooses the two prototypes with the smallest resulting distortion where update rules are formulated based on class membership. In LVQ2.1 variants, such as GLVQ, winner selection yields the in-class *and* out-of-class prototype vectors resulting in the smallest distortion for prototypes with label y^m and prototypes with label $y^r \neq y^m$. These are prototypes w^J and w^K, respectively. Winner selection in GRLVQ follows that of LVQ2.1 and GLVQ. For GRLVQ, the Euclidean distance is weighted using the relevance factors λ_i

$$d_\lambda^J = \sum_{i=1}^{n} \lambda_i \left(x_i^m - w_i^J \right)^2 \tag{7}$$

$$d_\lambda^K = \sum_{i=1}^{n} \lambda_i \left(x_i^m - w_i^K \right)^2. \tag{8}$$

B. Prototype Updates

Prototype vectors learn in an iterative fashion to define boundaries between neighboring classes. GRLVQ uses the differential shifting prototype update strategy of LVQ2.1 [2]. The best-matching in-class prototype (w^J) is moved toward the sample and the best-matching out-of-class prototype (w^K) is moved away from the sample, regardless of a correct or incorrect decision. Following the gradient–descent form of (1), updates for the in-class and out-of-class winning prototype vectors in [1] are

$$\Delta w^J = \frac{4\epsilon(t)^J f'|_{\mu(x^m)} d_\lambda^K}{\left(d_\lambda^J + d_\lambda^K \right)^2} \Lambda (x^m - w^J) \tag{9}$$

$$\Delta w^K = -\frac{4\epsilon(t)^K f'|_{\mu(x^m)} d_\lambda^J}{\left(d_\lambda^J + d_\lambda^K \right)^2} \Lambda (x^m - w^K) \tag{10}$$

where the loss function f is the sigmoid function of (3), f' is the derivative of the sigmoid function (4), $\epsilon(t)^J$ is the in-class learn rate, and $\epsilon(t)^K$ is the out-of-class learn rate.

C. Relevance Updates—Emphasizing Relevant Input Features

Relevance factors hold the potential for dimensionality reduction by learning input dimensions important for classification. They indicate the importance of each dimension by assigning a weight. Larger weights indicate that the corresponding input dimensions are more important than those input dimensions with smaller weights. We can select the most important input features to achieve a good classification by ordering features based on relevance.

The relevance factor updates are found via gradient descent in [1], similar to the prototype vector updates, giving the following update rule:

$$\Delta\lambda_i = \max \left\{ -\frac{2\epsilon(t)^\lambda f'|_{\mu(x_i^m)} d_\lambda^K \left(x_i^m - w_i^J \right)}{\left(d_\lambda^J + d_\lambda^K \right)^2} + \frac{2\epsilon(t)^\lambda f'|_{\mu(x_i^m)} d_\lambda^J \left(x_i^m - w_i^K \right)}{\left(d_\lambda^J + d_\lambda^K \right)^2}, 0 \right\}. \tag{11}$$

Relevance factors are scaled such that $||\lambda||_1 = 1$ in order to avoid numerical instabilities [1]. Although it does not matter which norm is used, we choose the l_1-norm because it may have a convenient interpretation as a probability.

III. Improvements to GRLVQ for High Dimensions

In our analysis of GRLVQ, three issues emerge. By addressing them, increased convergence speed and classification accuracy result. First, we discover a flaw with the prototype update rule when a sample is classified correctly. This flaw, discussed in Section III-B, can cause prototypes to diverge. Our improved learning rule solves the divergence problem and decreases training time. Second, we find that GRLVQ suffers from poor prototype utilization where many of the prototypes never win the competitive selection process. In Section III-C, we offer an adaptation of DeSieno's *conscience mechanism* [15] to the competition resulting in a significant increase in classification accuracy. In the closing remarks of [1], Hammer and Villmann suggested the possibility that a maximum entropy approach could bring improvements to GRLVQ. Finally, we discuss the setup and initialization of LVQ classifiers. We suggest that the proposed maximum entropy prototype update strategy (i.e., DeSieno's conscience mechanism) can alleviate the need for time consuming prototype vector initialization schemes. Throughout our discussion, we will refer to our improvements to GRLVQ as GRLVQ-improved (or GRLVQI for short). We will use GRLVQ when referring to the work by Hammer and Villmann.

A. Analysis of LVQ2.1 and GRLVQ Update Windows

GRLVQ uses the differential shifting prototype update strategy of LVQ2.1 discussed in Section II-B. In Kohonen's treatment of LVQ2.1, a window is used to promote development of the decision boundary. Both in-class and out-of-class prototypes are updated if and only if the following condition holds [2]:

$$\min \left(\sqrt{\frac{d^J}{d^K}}, \sqrt{\frac{d^K}{d^J}} \right) > \frac{1 - D}{1 + D} \tag{12}$$

where D is the width of a window centered about the midpoint of the best-matching in-class and best-matching out-of-class prototype vectors (w^J and w^K, respectively). The value D is a percentage of the Euclidean distance between w^J and w^K. As an example, a window of width $D = 0.5$ occupies 50% of the distance between w^J and w^K centered at $(w^J + w^K)/2$. Although the window of LVQ2.1 is often viewed as a step function about the midpoint of w^J and w^K [2], [13], what is described by (12) is different.

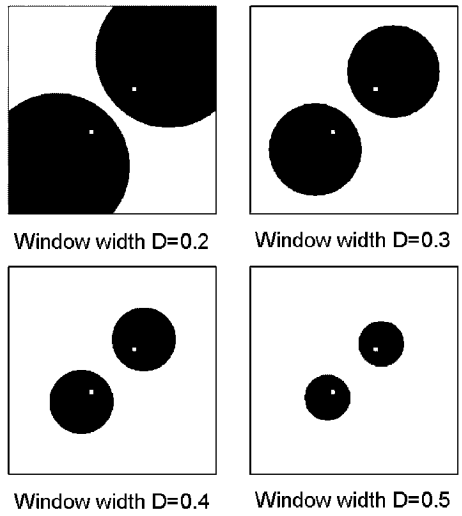

Fig. 1. Kohonen's window for 2-D data. For inputs in the black regions, prototype vectors *are not* updated; white regions are those areas where prototype vectors *are* updated. Prototypes are indicated by white dots located in the black regions.

We show the solution of (12) that results in updates to the prototype vectors w^J and w^K by deriving a closed formula. In doing so, it is necessary to describe two boundaries, one for the winning in-class prototype vector and one for the winning out-of-class prototype vector.

For the in-class prototype w^J, we define the boundary by equating the left-hand side of (12) to the right-hand side. For simplicity, we rewrite $(1-D)/(1+D)$ as $(a)/(b)$ where $a = (1-D)$ and $b = (1+D)$. If the sample is correctly classified, then $(d^J)/(d^K)$ of (12) is the smallest

$$\sqrt{\frac{d^J}{d^K}} = \frac{1-D}{1+D} = \frac{a}{b}.$$

Squaring both sides gives

$$\frac{d^J}{d^K} = \frac{a^2}{b^2}.$$

In two dimensions, we can rewrite the previous equation as

$$\frac{\left(w_1^J - x_1\right)^2 + \left(w_2^J - x_2\right)^2}{\left(w_1^K - x_1\right)^2 + \left(w_2^K - x_2\right)^2} = \frac{a^2}{b^2}.$$

After cross multiplying, collecting terms, and completing the square, we have the equation for a circle that describes the update boundary (about w^J). The circular boundary is not centered at w^J; rather, it is centered at

$$\left(-\frac{a^2 w_1^K - b^2 w_1^J}{b^2 - a^2}, -\frac{a^2 w_2^K - b^2 w_2^J}{b^2 - a^2}\right)$$

with radius

$$\left(\frac{a^2 w_1^K - b^2 w_1^J}{b^2 - a^2}\right)^2 + \left(\frac{a^2 w_2^K - b^2 w_2^J}{b^2 - a^2}\right)^2 - \frac{b^2\left(w_1^J + w_2^J\right) - a^2\left(w_1^K + w_2^K\right)}{b^2 - a^2}.$$

Similarly, one can find the boundary for the out-of-class prototype vector by equating $\sqrt{d^K/d^J}$ to $(a)/(b)$ and rearranging in the form of $(x_1 - c)^2 + (x_2 - d)^2 = e$, where the center is described by c and d and the radius by e.

For the two-class case, with a single prototype vector for each class, (12) describes two spherical regions about each prototype vector. Updates *do not* occur if the sample falls *inside* the spherical regions. All samples lying outside the spherical regions result in updates to the prototypes. Prototypes are not located at the sphere centers; they are eccentrically located in equal but opposite directions (Fig. 1). For multiple classes with multiple prototype vectors per class, the results are similar but are further constrained by the Voronoi cells.

We rewrite the LVQ2.1 prototype updates to include indicator functions that capture Kohonen's update rule. That is, $g_J(\cdot)$ and $g_K(\cdot)$ in (13) and (14) evaluate to 1 if the condition in (12) holds

$$\Delta w^J = \epsilon(t) g_J(x^m, w^J, w^K)(x^m - w^J) \tag{13}$$

$$\Delta w^K = -\epsilon(t) g_K(x^m, w^J, w^K)(x^m - w^K). \tag{14}$$

In general, $g_J(x^m, w^J, w^K)$ and $g_K(x^m, w^J, w^K)$ are composed of all the information in update rules (9) and (10) except the learn rate and the difference between the prototype and the input sample. A 2-D example of the Kohonen window is presented in Fig. 1 for varying window widths. Black regions of Fig. 1 equate to $g_J(\cdot) = g_K(\cdot) = 0$, while white regions equate to $g_J(\cdot) = g_K(\cdot) = 1$. Prototypes are indicated as white dots in the black regions. Fig. 1 clearly shows that updates to w^J and w^K not only occur when the sample falls within a window about their midpoint, but also they are updated for a wide range of input samples. It is unclear what effect the Kohonen window has on the development of the decision boundary for samples lying outside the midpoint between w^J and w^K. Perhaps this contributes to the divergence problem LVQ2.1 exhibits.

Sato and Yamada's generalization of LVQ2.1 [13] alleviates the need of the windowing function as it is replaced by a Gaussian-like window as part of the new update formulation. In order to evaluate the effect of the GRLVQ update window, we define the two windowing functions $g_J(\cdot)$ and $g_K(\cdot)$ of (13) and (14) as

$$g_J(x^m, w^J, w^K) = \frac{4 d_\lambda^K f'|_{u(x^m)} \Lambda}{\left(d_\lambda^J + d_\lambda^K\right)^2} \tag{15}$$

$$g_K(x^m, w^J, w^K) = \frac{4 d_\lambda^J f'|_{u(x^m)} \Lambda}{\left(d_\lambda^J + d_\lambda^K\right)^2}. \tag{16}$$

To provide some illustration, we consider a two-class problem in one dimension. The in-class prototype (w^J) and the out-of-class prototype (w^K) are fixed at locations -5 and 5 (indicated by dots in Fig. 2). The values of the windowing functions $g_J(\cdot)$ and $g_K(\cdot)$ are plotted on the y-axis for different values of the input sample x^m along the x-axis. Function values evaluated to the left of the line defined by the input sample $x^m = 0$ are for a correct classification whereas the function values evaluated to the right are for a wrong decision. For the 1-D case, $\Lambda = [\lambda_{11}] = \mathbf{1}$.

For a correct decision, Fig. 2 shows that w^J is moved much closer to the sample than w^K is moved away. For the extreme

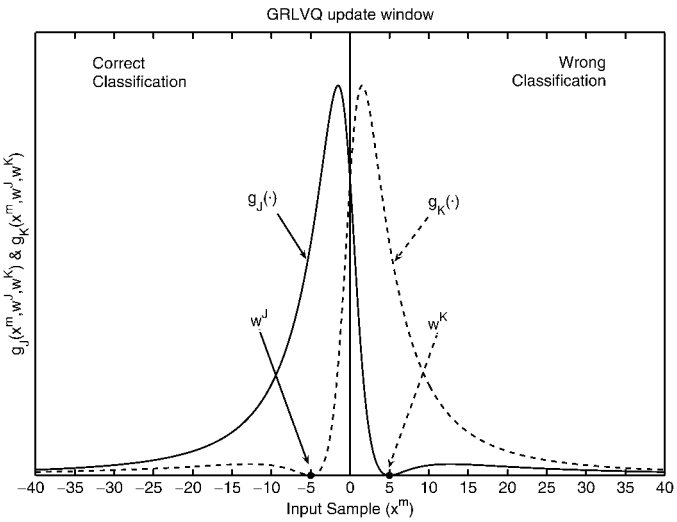

Fig. 2. Effective update window for GRLVQ for 1-D data. The values of $g_J(\cdot)$ and $g_K(\cdot)$ along the y-axis are evaluated for fixed prototype locations at $-5(w)$ and $5(w)$, where the input sample x^m varies along the x-axis. Solid curves reflect updates to the in-class prototype vector (w^J) and dashed curves reflect updates to the out-of-class prototype vector (w^K).

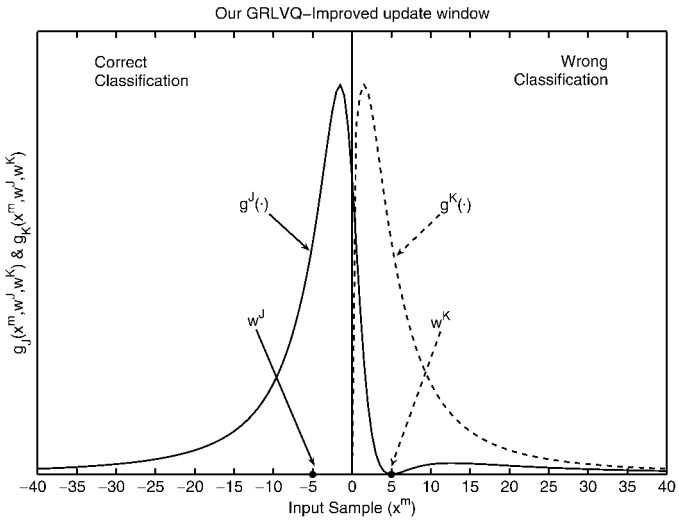

Fig. 3. Effective update window for our improved GRLVQI. The values of $g_J(\cdot)$ and $g_K(\cdot)$ along the y-axis are evaluated for fixed prototype locations at $-5(w)$ and $5(w)$, where the input sample x^m varies along the x-axis. Solid curves reflect updates to the in-class prototype vector (w) and dashed curves reflect updates to the out-of-class prototype vector (w).

case when $x^m = w^J$, no update occurs to w^K. For a wrong decision, w^J is moved slightly closer to the sample whereas w^K is moved much further away. Prototype vectors are updated most when the sample falls in the areas slightly skewed from their midpoint. It is clear from the graphical representations of the LVQ2.1 (Fig. 1) and GRLVQ (Fig. 2) windows that the latter can be fine-tuned to better promote the development of the decision boundary.

B. New Update Strategy

Our experimentation with GRLVQ reveals a flawed update rule which can lead to the divergence of prototype vectors. We attribute this divergence problem to a prototype being updated as a winning out-of-class prototype more than it is updated as a winning in-class prototype. This problem can occur if a class with a large number of samples (class *C1*) shares a boundary with a class with much fewer samples (class *C2*). In this example, *C1* will have as its nearest out-of-class prototype(s), any in-class prototype(s) of *C2*. Because *C1* has more samples than *C2*, *C2*'s prototypes will be moved further and further away from the boundary. Class *C2* having few samples will unsuccessfully attempt to reposition its in-class prototypes to redefine its boundary between itself and *C1*. We see that given the aforementioned scenario, divergence can occur. Even if divergence does not occur, it is easy to see that the current GRLVQ update strategy can have a negative effect on the rate of convergence because prototypes properly positioned may be moved unnecessarily.

Our in-class conditional update addresses the divergence problem by changing the update strategy for correctly classified samples. We argue that it is only necessary to adjust the out-of-class prototype if a sample is incorrectly classified. The new update rule is then to move the in-class prototype towards the sample and the out-of-class prototype away from the sample only if a sample is misclassified. If the sample is classified correctly, we adjust the in-class prototype towards the sample and leave the out-of-class prototype unchanged. We can view the effect of our update rule in Fig. 3.

One can additionally employ a time decay in the exponent of the sigmoid function to emphasize the development of the decision boundary. Our two windowing functions $g_J(\cdot)$ and $g_K(\cdot)$ with a time decay term (τ) are

$$g_J(x^m, w^J, w^K, \tau) = \frac{4d^K f'|_{u(x^m),\tau}\Lambda}{(d^J + d^K)^2} \tag{17}$$

$$g_K(x^m, w^J, w^K, \tau) = \frac{4d^J f'|_{u(x^m),\tau}\Lambda}{(d^J + d^K)^2} \tag{18}$$

where $f'|_{u(x^m),\tau}$ is defined as

$$\begin{aligned} f'(\mu(x^m),\tau) &= f(\mu(x^m),\tau)[1 - f(\mu(x^m),\tau)] \\ &= \frac{e^{-\left(\tau\frac{d^J-d^K}{d^J+d^K}\right)}}{\left(1+e^{-\left(\tau\frac{d^J-d^K}{d^J+d^K}\right)}\right)^2}. \end{aligned} \tag{19}$$

Increasing τ will cause the update window to converge from a bimodal to a unimodal window about the midpoint of w^J and w^K (Fig. 4). As $\tau \to \infty$, the functions $g_J(\cdot)$ and $g_K(\cdot)$ converge to functions similar to a degenerative Gaussian (i.e., a Gaussian with infinitely small variance). In this extreme case, the update window would consist of a single point positioned at $(w^J + w^K)/2$.

Sato and Yamada [13] suggest using a constant learn rate, starting with large windows and shrinking with increasing training time. While this would affect the width of the window and perhaps enhance the development of the classification boundary, decreasing the learning rate is a good idea to control how far a prototype can be adjusted once the boundary is well defined.

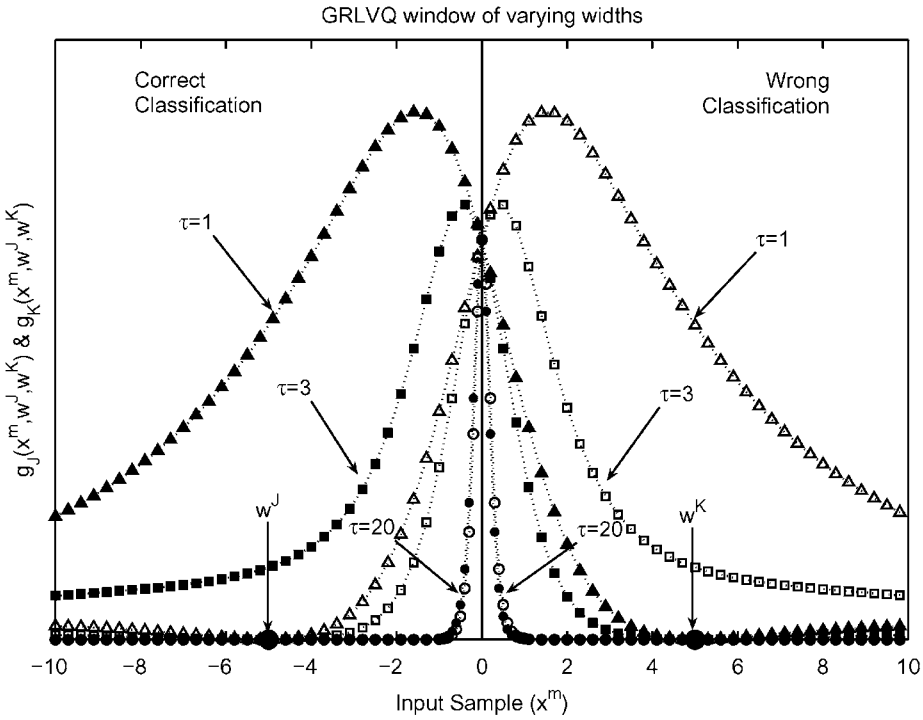

Fig. 4. GRLVQ window converges to a unimodal window about the midpoint of w^J and w^K if a time decay factor τ is applied as in (19). The filled triangle, filled square, and filled circle curves are for $g_K(\cdot)$ evaluated at $\tau = 1, 3, 20$, respectively. The open triangle, open square, and open circle curves are for $g_J(\cdot)$ evaluated at $\tau = 1, 3, 20$, respectively.

C. Improving Prototype Utilization

Poor prototype utilization (i.e., the dead neuron problem) is a classic problem with prototype-based learning methods that often results in the algorithm getting stuck in local optima. Solutions to the dead neuron problem exist in the literature and may be categorized as direct or indirect methods. Direct methods keep track of winning frequency and force the algorithm to use less frequently winning prototypes during the learning process. Indirect methods address other underlying problems that indirectly result in a solution to the dead neuron problem.

Examples of the former include conscience learning [15] and frequency-sensitive competitive learning [16]. Examples of the latter include Kohonen's LVQ-SOM [2] and supervised (relevance) neural gas (S(R)NG) [17], [18] by Hammer *et al.* where both incorporate neighborhood cooperation in the learning process. The consequence of the neighborhood cooperation in S(R)NG is robustness for multimodal data. Furthermore, if the neighborhood is large in the beginning, then all prototype vectors will likely have an opportunity to learn some important aspect of the data.

Other prototype-based learning methods do not suffer from the dead neuron problem as they do not fix the number of prototype vectors at the beginning of the training process. Some start with one prototype vector per class and add them based on some criterion (see, e.g., [19]). In these schemes, the initial prototype vector learns by force (no other labeled prototypes exist) and the newly added prototypes are initialized based on some meaningful criterion (i.e., prototypes are initialized to reduce the existing system error). It is important to note that further learning may or may not update all prototypes in each class. It may not be necessary for this to occur for the algorithm to be very successful at avoiding local minima.

We take the direct approach and combine the power of DeSieno's conscience learning from the self-organizing map (SOM) community to the in-class prototype updates in GRLVQ. This equiprobabilistic winner selection strategy allows for an optimal vector quantization and allows prototype vectors to contribute quickly [15]. DeSieno's conscience mechanism adds a bias B^p to the Euclidean distance of the input x^m and the prototype w^p, which modifies the chance of w^p becoming the winner. B^p is calculated from the winning history for each w^p so as to discourage frequent winners from winning more often and encourage the selection of infrequent winners.

Update to the frequency F^p for the winning prototype vector is

$$F^p_{\text{new}} = F^p_{\text{old}} + \beta\,(1.0 - F^p_{\text{old}})\,. \tag{20}$$

For the remaining prototypes, the frequency is adjusted as

$$F^p_{\text{new}} = F^p_{\text{old}} + \beta\,(0.0 - F^p_{\text{old}})\,. \tag{21}$$

The β term in (20) and (21) is a user-defined parameter that controls the amount of update to the frequencies F^p.

For winner selection, one uses the biased Euclidean distance between the input sample x^m and the prototype w^p

$$d_{\text{Bias}} = d^p - B^p \tag{22}$$

where B^p is defined as

$$B^p = \gamma\left(\frac{1}{P} - F^p_{\text{old}}\right). \tag{23}$$

The γ term in (23) is a user-defined parameter that controls the amount of bias applied to the Euclidean distance.

We can take advantage of DeSieno's conscience mechanism for GRLVQI by applying a separate conscience for each class. That is, during the in-class selection, we bias the Euclidean distance calculation as described by (22), where the bias is defined in (23). We then update the frequency of the winning in-class prototype vector w^J as in (20) and the frequency of the nonwinning in-class prototype vectors as in (21). During the out-of-class selection, we *do not* bias the Euclidean distance. Further, the unmodified Euclidean distance is used in the update rules for w^J, w^K, and λ.

D. Setup and Initialization of an LVQ

Setup and initialization are important design considerations of any LVQ. Properly addressed, an LVQ will perform well. Improperly addressed, an ineffective classifier can result. Three aspects encompass the setup and initialization of GRLVQ. First, we define the number of classes we are interested in processing. This is determined by the specific classification problem. Second, we define the number of prototype vectors per class. Third, we assign an initial state to the prototype vectors.

Determining the number of prototype vectors to assign per class can have a significant impact on the classifiers ability to discern the material classes of interest. We would like to turn to existing theory to give us guidance on how to best determine the number of prototypes we should assign per class. Existing theory for LVQ classifiers based on Vapnik–Chervonenkis (VC) dimension (see, e.g., [20]) only give us guidance on the total number of prototype vectors we should assign to our classifier [11]. More recent works based on Gaussian complexities [21] suffer from the same drawback as the VC dimension theory; we are given guidance only on the total number of prototype vectors for the classifier, not the number of prototype vectors for each class. As such, we are forced to design our networks based on "rules-of-thumb" or heuristics. In our experiments, we use five prototypes per class for three classification problems of increasing difficulty. This number was empirically determined and shows good results for our experiments.

Once defined, the prototypes must be initialized. The initial state of the prototype vectors is known to affect the quality of the final classifier [2]. Several options exist for giving the classifier a "jump-start" in refining the prototypes to their final converged state. One can initialize prototype vectors based on the sample distribution or geometry of the sample set [22] by using some other algorithm to take prototypes from an initial random state to a state suitable for refinement [1], [2] or by using random initialization and ensuring prototypes converge to a local optimum [23]. The latter has theoretical results, but does not appear to work any better than the others based on our analysis. In an effort to find a reasonable solution to this problem, we found that GRLVQ suffers from poor prototype utilization discussed in Section III-C. We found that adding a conscience mechanism in the learning process allows us to use a simple random initialization scheme and obtain consistent results. Further, using a decaying learn rate for the in-class and out-of-class prototype vector updates ($\epsilon^J(t)$ and $\epsilon^K(t)$) as well as relevance factor updates ($\epsilon^\lambda(t)$) helps us achieve very high classification accuracies. We note that, for easier classification problems, high accuracies could be achieved with a constant learn rate as in [1].

In the following sections, we evaluate the classification performance of GRLVQ(I) on hyperspectral data. We test the discrimination capability of the GRLVQ(I) extracted features by comparing the classification performance of all available spectral features with the classification performance of the GRLVQ(I) extracted features using an independent classifier.

IV. Applying GRLVQ to Hyperspectral Images

We compare our improved GRLVQI to GRLVQ by using both versions to learn the relative importance of the input features of real-world hyperspectral data obtained by the NASA/JPL airborne visible/infrared imaging spectrometer (AVIRIS) [24]. We further demonstrate the quality of the reduced feature set discovered by GRLVQ(I) by comparing classification accuracy using only the features discovered by GRLVQ(I) versus all available features.

A. Data

We use training and testing samples from the Lunar Crater Volcanic Field (LCVF) scene acquired by AVIRIS in 1994. After atmospheric correction and elimination of the saturated water bands containing irrecoverable data, our hyperspectral scene has 194 spectral bands. Each spectrum (194-dimensional vector) is then normalized to unit length by dividing the vector elements by the l_2 norm of the vector. This cancels linear effects such as shading resulting from viewing geometry, making the spectral classes much more uniform. It also has the undesirable effect of eliminating the differences in geometric albedo for materials that have the same spectral signature and only differ in their albedo [25]. Fortunately, this is rarely the case, but one should be aware of the possibility and, if necessary, do postclassification processing to separate such materials.

Three classification problems of increasing difficulty are used to emphasize our improvements to GRLVQ: a 7-, 23-, and 35-class problem. A description of the 23 classes is provided in Table I along with the class labels and the number of training samples, and representative spectra are provided in Fig. 5. Further details on the 23-class data set can be found in [8] and [9]. The 7-class problem is a subset of the 23-class problem and consists of classes **A, D, E, H, I, L**, and **W**. Our 35-class problem is an extension of the 23-class problem described previously. Through an independent SOM clustering [8], [27] of the data and after careful scrutiny of their statistics and spatial distribution, 12 additional classes were identified beyond the previously known 23 classes. Class labels and corresponding mean spectra are displayed in Fig. 6. Interpretive class descriptions are unavailable at this time for the 12 additional spectral classes.

B. Design of Experiments

We consider classification problems with varying degrees of difficulty to demonstrate the effectiveness of GRLVQI over GRLVQ. We demonstrate that relevance-selected features maintain or improve the classification performance of even the most rudimentary classifier. This is accomplished by comparing

TABLE I
CLASS DESCRIPTIONS, LABELS, AND NUMBER OF SAMPLES FOR THE 7- AND 23-CLASS PROBLEMS

Class	Cover type description	# tr.	Class	Cover type description	# tr.
A	Hematite-rich cinders	72	M	Alluvium #3 (iron rich)	14
B	Rhyolite of Big Sand Spring Valley	22	N	Dry wash #1	15
C	Alluvium #1	50	O	Dry wash #2	54
D	Dry playa	160	P	Dry wash #3	45
E	Wet playa #1	115	Q	Wet playa #2	15
F	Young basalt	21	R	Wet playa #3	14
G	Shingle Pass tuff	7	S	Wet playa #4	15
H	Alluvium #2 (with mixed scrub brush, rocks, and soil)	50	T	Wet playa #5	18
I	Old basalt	36	U	Alluvium #4 (also iron rich)	36
J	Dense scrub brush stands	12	V	Wet playa #6	12
K	Basalt cobbles on playa	37	W	Ejecta blankets #2 (primarily unoxidized cinders with smaller percentage of hematite-rich cinders)	33
L	Ejecta blankets #1 (mixed hematite-rich and unoxidized cinders)	78			
				Total number of training samples	931

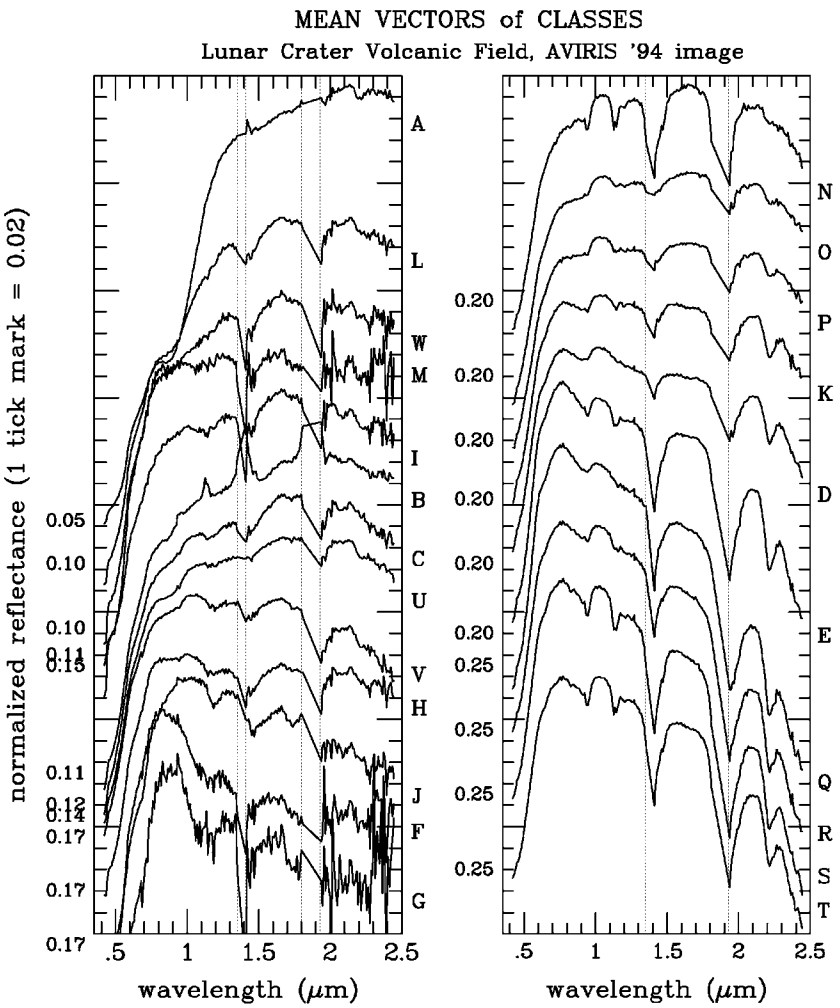

Fig. 5. Average spectra of the 23 classes described in Table I offset for viewing convenience. The dotted vertical lines indicate data fallout due to saturation of the atmospheric water vapor bands.

the results of the minimum Euclidean distance (MED) classifier's classification obtained with all (194) spectral features to the classification results obtained with the reduced feature sets produced by GRLVQ(I) for the 7-, 23-, and 35-class problems.

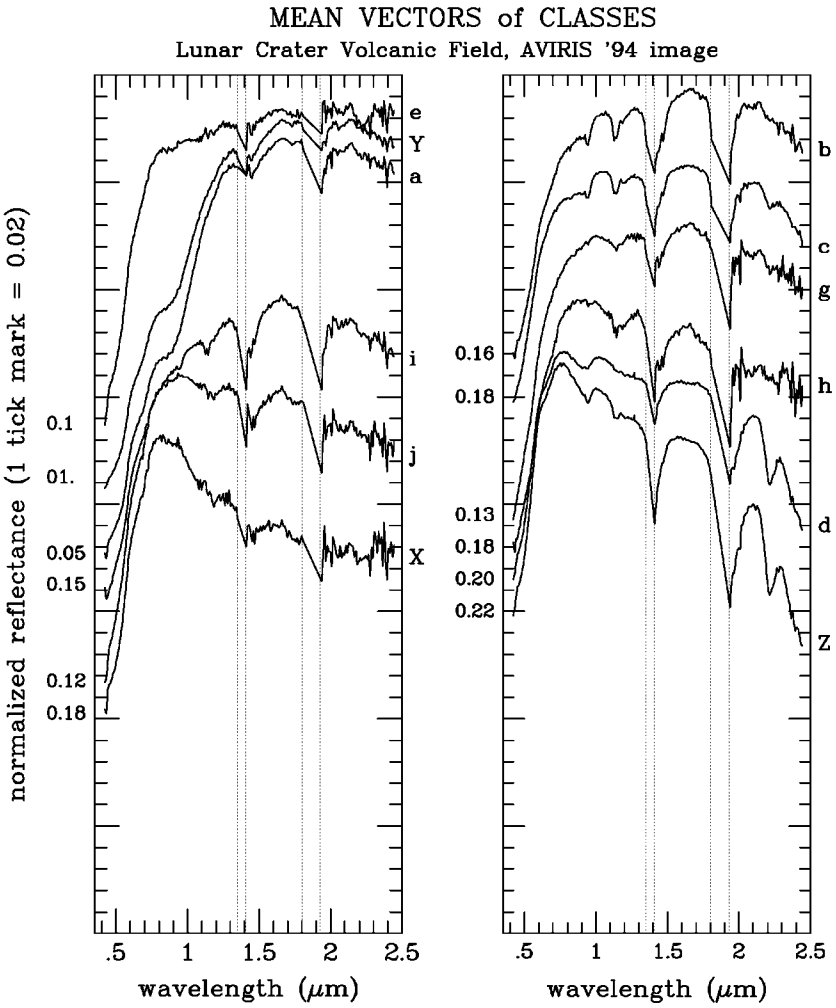

Fig. 6. Average spectra corresponding to the 12 additional classes for the 35-class problem.

TABLE II
LEARN SCHEDULE EMPLOYED FOR THE 23- AND 35-CLASS PROBLEMS. THE NUMBER OF TRAINING STEPS (TS) IS ROUNDED TO THE NEXT THOUSAND. THE FIRST THREE PARAMETERS ARE THE SAME AS IN THE ORIGINAL GRLVQ WHEREAS THE LAST TWO PARAMETERS CONTROL THE CONSCIENCE ASPECT OF THE LEARNING

Schedule of GRLVQ(I) Learning Parameters

	GRLVQ(I) Params			Consc. Params	
Training Steps (TS)	ϵ^λ	ϵ^J	ϵ^K	γ	β
$0 < TS \leq 400K$	0.005	0.025	0.025	2	0.35
$400K < TS \leq 800K$	0.0025	0.0125	0.0125	2	0.3
$800K < TS \leq 1.2M$	0.001	0.005	0.005	2	0.225
$1.2M < TS \leq 1.6M$	0.0005	0.0025	0.0025	2	0.125

We first demonstrate the effect of the in-class conditional update on the rate of convergence using the 7-class problem. For this, we *do not* use conscience learning in order to isolate the effect of the in-class conditional update. Next, we execute GRLVQ(I) on the 23-class data set to show improved classification accuracy due to the in-class conditional update and conscience learning together. Then, we repeat this with the 35-class data set, which further shows the advantage of GRLVQI over GRLVQ.

TABLE III
TEST AND TRAINING SAMPLE ASSIGNMENT FOR THE 7-, 23-, AND 35-CLASS PROBLEMS

	7-Class	23-Class	35-Class
Training Samples	347	621	976
Test Samples	173	310	488
Total Samples	520	931	1464

Table II lists the GRLVQ(I) learning parameters and the GRLVQI conscience parameters used for the 23- and 35-class problems. Ideally, γ should decay with time. However, a constant γ worked well in our classification problems serving our purpose of demonstrating that conscience learning works in a supervised setting.

Three splits of the data are generated using two-thirds for training and one-third for testing (Table III). Each split is a random selection of known training samples. Results are reported as the average of the *test* results of the three jackknife runs, in each of which classification accuracies are calculated as the mean of the individual class accuracies.

When comparing GRLVQI to GRLVQ, we start from the same initial state for prototype vectors. Prototypes are drawn randomly from a uniform distribution on [0.4,0.6]. Data are

TABLE IV
VALIDATING GRLVQ(I) PERFORMANCE WITH THE LCVF AVIRIS 23-CLASS DATA SET

	Hybrid ANN	GRLVQ	GRLVQI
Avg. Test Acc.	92.1%	95.1%	97.0%

scaled to the interval [0,1] and each class is assigned five prototypes.

C. Benchmark Classification of the LCVF Data Set

To make an assessment of the performance of GRLVQ(I) as a classifier (as opposed to making a relative assessment of the quality of the extracted features for classification), we classify the 23-class data with a hybrid artificial neural network (ANN), using all 194 spectral features. This hybrid ANN is more capable of exploiting the intricacies of high-dimensional data than the MED classifier, thus we get a more challenging benchmark for GRLVQ(I). The classification accuracy achieved by the hybrid ANN is 92.1% (Table IV), close to that of both GRLVQ and GRLVQI. Because GRLVQ(I) discovers the most important features for classification, we expect GRLVQ(I) to achieve a higher classification accuracy than the hybrid ANN achieves on all available (194) spectral features. We believe the classification accuracy of the hybrid ANN on the testing data to be a representative performance, for two reasons. One is that the same hybrid ANN architecture was used in previous studies to classify the entire LCVF AVIRIS scene (614×420 pixels each with 194 spectral features), where the data presented in Table I served as training spectra for the respective 23 classes [26], [9]. The classification accuracy on the entire image was evaluated according to requirements of rigorous statistical assessment based on sampling theories, and accordingly, on a collection of the rather large requisite number of "ground truth" pixels. The accuracy of the hybrid ANN classifier was ≈90%, followed by an MED classifier with 83%, and a spectral angle mapper (SAM) classifier with ≈80%. (ML classifier, also included in that study, could not be applied to the 194-dimensional data for lack of sufficient number of training samples, which is dimension dependent for the ML, and the best performance on the maximum possible number of retained spectral features was ≈51% accuracy. On the same reduced-dimensional data, the hybrid ANN, MED, and SAM produced 75%, 73%, and 67% accuracies, respectively.) While the accuracy produced by the hybrid ANN on the entire AVIRIS image is lower than the accuracy it produced on the data set used in the present study, one must keep in mind that the data in this study is a rather small set, representing the well-examined, clean spectral specimen from the LCVF image. Therefore, it is expected that classification accuracy is higher when training on part of the previous studies' trusted training samples and testing on the remaining part of that trusted sample set, compared to when all noisy pixels participate in the classification. We also want to point out that the previous studies were conducted years ago when computing power was more limited than today; therefore, it may be possible that with longer training the hybrid ANN would achieve somewhat higher classification accuracy.

The aforementioned hybrid ANN is described in several earlier publications ([8], [9], and references therein). Briefly, it consists of a 2-D SOM as a hidden layer, coupled with a categorization output layer that learns via the Widrow–Hoff learning rule. First, the SOM is allowed to learn in unsupervised mode, and afterwards the supervised learning of the output layer is turned on. The preformed clusters in the SOM help the output layer to refuse learning of inconsistent class labels thus resulting in higher overall precision of the classification.

D. GRLVQ Results

The benefit of our in-class conditional update is shown in Fig. 7, where we achieve ≈35% faster convergence to the maximum classification accuracy. Classification accuracy between our GRLVQI with our in-class conditional update rule and without conscience learning and GRLVQ are equal. For this demonstration, only the 7-class data set was used. We do not consider the speedup once conscience learning is used because the benefit of the speedup due to the in-class conditional update is absorbed in the extra processing time required to ensure that *all* prototype vectors learn.

Table V lists the summary of all results for GRLVQ(I) for the three classification problems. The addition of the conscience mechanism to the competitive winner selection increases classification accuracy from 95.1% to 97% (nearly 2%) for the 23-class problem. More impressive is the 5.6% gain in classification accuracy from 91.6% to 97.2% for the 35-class problem.

As we increase the number of surface materials for classification, we see an increase in the number of spectral components GRLVQ(I) requires to distinguish those classes. To show which spectral regions are important for classifying the 7-, 23-, and 35-class problems, we plot relevance factors with mean spectra of representative classes. For the 7-class problem, spectral features corresponding to large relevance factors fall below 0.9μm (Fig. 8). An additional 16 classes result in three significant regions for the 23-class problem: 0.44–0.68, 0.71–1.11, and 1.42–1.68 μm (Fig. 9). Further, adding 12 classes brings us to the 35-class problem and four important spectral regions: 0.44–0.68, 0.71–1.11, 1.42–1.68, and 2.1–2.24 μm with spurious relevances in the 1.94–2.48-μm range (Fig. 10).

It is reasonable to assume that we achieve higher classification accuracy largely because we are making better use of classification resources. Improved resource utilization enables better definition of decision boundaries which likely affects which spectral components GRLVQI discovers for classification. We compare relevance factors of GRLVQI (black) to relevance factors of GRLVQ (red) for the 23-class problem in Fig. 11. We see that GRLVQI not only discovers those features similarly discovered by GRLVQ, but it also places an emphasis in the 1–1.1-μm range. It is possible that the emphasis placed in that range contributes to the increase in classification accuracy achieved by GRLVQI over GRLVQ for the 23-class problem.

E. MED Classification With Relevance-Selected Features

dAre the features discovered by GRLVQ(I) meaningful? Certainly, GRLVQ(I) s a classifier is able to use the extracted feature set to attain good classification results. If the features are meaningful for classification, then we should not see a decrease, but possibly an increase, in classification performance with an independent classifier when comparing accuracy with all available features to those discovered by GRLVQ(I). This is not to

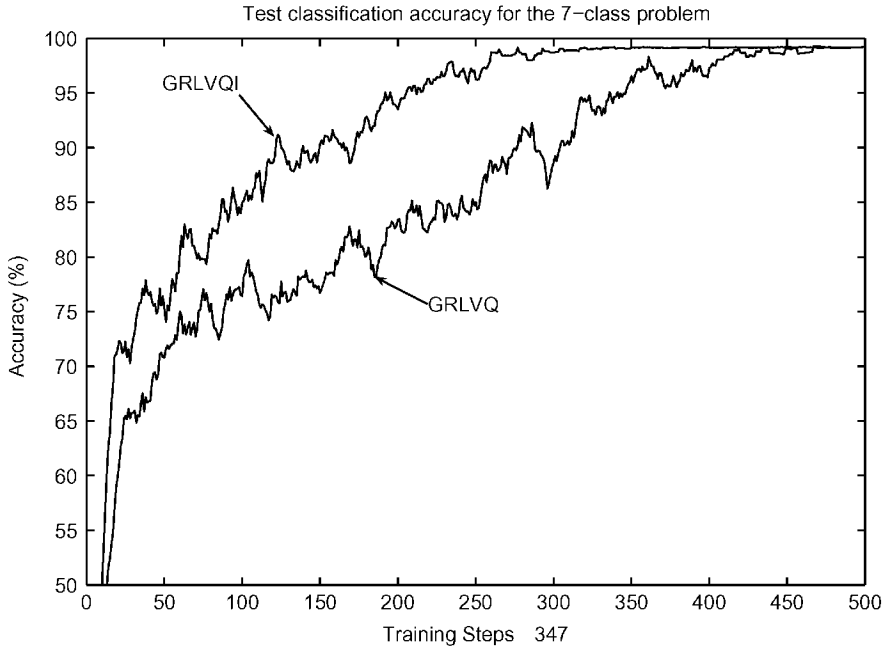

Fig. 7. Test classification accuracy comparing GRLVQI's in-class conditional update (without conscience learning) to that of GRLVQ on the 7-class data set. Results are averaged over three jackknife runs, in each of which, the accuracy is computed as the average of the individual class accuracies.

TABLE V
TEST CLASSIFICATION ACCURACY FOR GRLVQI AND GRLVQ AND THOSE FEATURES WITH RELEVANCES ≥ 0.001

GRLVQ Classification Results for 3 Classification Problems

		Accuracy	# Features
7-Class Problem	GRLVQI	99.2%	45
	GRLVQ	99.3%	45
23-Class Problem	GRLVQI	97.0%	81
	GRLVQ	95.1%	83
35-Class Problem	GRLVQI	97.2%	107
	GRLVQ	91.6%	109

say that features discovered by GRLVQ(I) are the best possible features. It shows, however, that the feature sets are useful for classification.

The quality of the extracted features in preserving the discrimination capability of the data set is demonstrated using the MED classifier. Features are selected in a cumulative fashion starting with the spectral feature with the largest relevance, and each additional spectral feature is selected in descending order of relevance. The MED classification is performed with each newly added feature. Here, we do not scale the selected features by the computed relevances λ_i; we only use the relevance factors to select those features for classification. This process continues until the classification accuracy no longer increases or begins to decline. Table VI lists the maximum achieved MED classification accuracy using relevance selected features. For comparison purposes, a baseline MED classification is provided using all available (original) 194 features. From these results, we clearly see that the set of features discovered by GRLVQ(I) is indeed better for classification than the set of all available (194) spectral features, at least for the MED classifier. This observation is consistent across all three classification problems and across feature sets obtained by GRLVQ(I).

F. Using Theory on Generalization Bounds to Estimate Classifier Performance

The results presented in Section IV-D empirically show that GRLVQI generalizes better on unseen instances than GRLVQ. This claim, in principle, could be supported by theoretical arguments, because recent publications offer results on generalization bounds specifically for LVQ-type classifiers.

Crammer *et al.* derive an upper bound for the generalization error for the LVQ2.1 family of classifiers. The generalization error is the empirical error over the training samples plus a term that is a function of the VC dimension [11], and is dependent on the dimensionality of the data. The empirical error is based on the maximal margin principle [11] and penalizes margins which are smaller than some threshold θ. However, Hammer *et al.* [21] state that the VC-dimension approach is not valid for LVQ classifiers with an adaptive diagonal metric such as RLVQ and GRLVQ, and hence GRLVQI.

Using the Radamacher–Gaussian complexity described in [28], Hammer *et al.* [21] derive upper bounds on the generalization error of LVQ classifiers that use the winner-takes-all rule and include classifiers with an adaptive diagonal metric. This formulation is independent of the dimensionality of the data. The derived generalization error is the sum of three terms. The first term is a function of the empirical error over the training samples, which penalizes misclassifications and small margins. The second term is an empirical Gaussian complexity term which can be influenced by the magnitude of the training samples x^m or the converged weights, whichever has the largest l_2 norm. The third term depends on the number of training samples and the confidence in the estimate. These last two terms are inversely proportional to the margin and hence favor

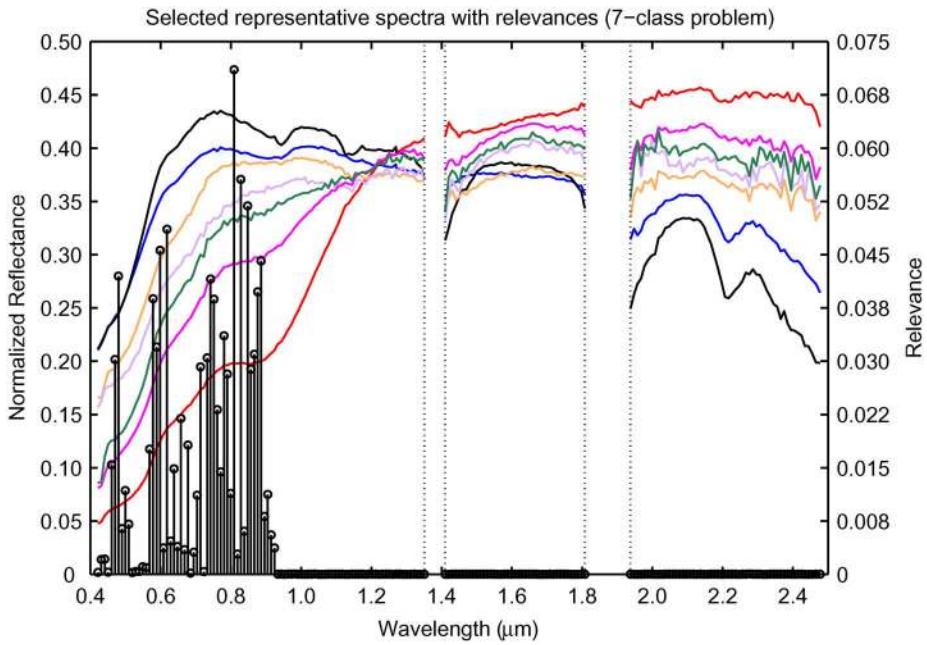

Fig. 8. Average spectra for classes A (red), D (blue), E (black), H (orange), I (purple), L (magenta), and W (green). Relevance factors are the averages of three jackknife runs obtained by GRLVQI (black stem plot) for the 7-class problem. The dotted vertical lines indicate data fallout due to saturation of the water bands.

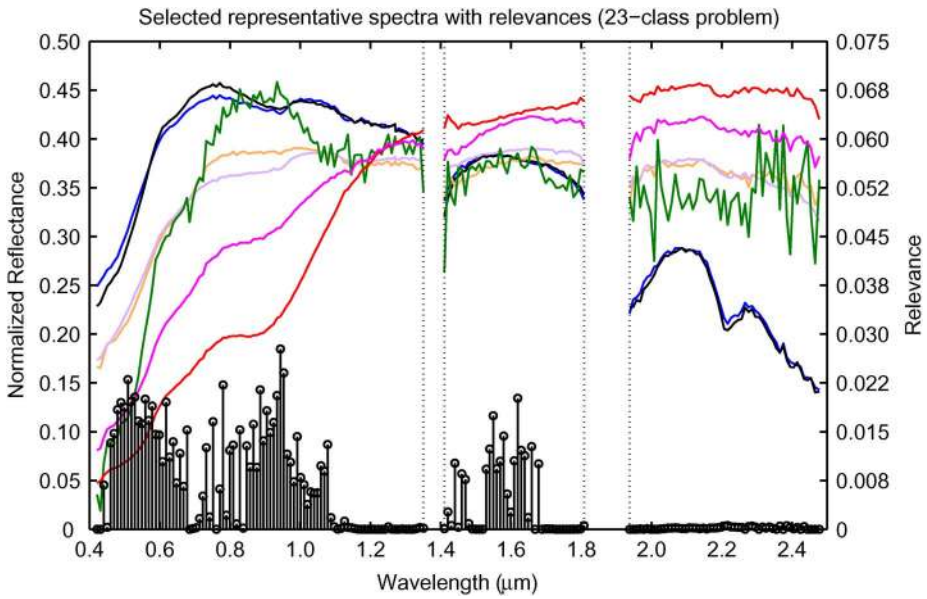

Fig. 9. Average representative spectra of classes A (red), G (green), H (orange), L (magenta), O (purple), Q (black), and R (blue). Relevance factors are the averages of three jackknife runs obtained by GRLVQI (black stem plot) for the 23-class problem. Classes were selected to show largest diversity. The dotted vertical lines indicate data fallout due to saturation of the water bands.

large margins. However, we find that even these most recent developments on the generalization error do not provide us with a means to compare the expected performance of GRLVQI with GRLVQ for richer problem sets with more than two classes.

V. Summary and Discussion

The GRLVQ is a recent neural paradigm which, until now, has gone unexploited for feature extraction for high-dimensional data such as hyperspectral images. Our analysis revealed deficiencies, which we addressed in this paper. Experimental results show that our modifications improve classification accuracy while speeding up the learning process. We demonstrated the potential of our improved GRLVQ variant GRLVQI as a powerful hyperspectral analysis tool by classifying real-world 194-band AVIRIS data. Our speedup study was conducted using only the in-class conditional update on the 7-class problem where we show a convergence speedup of $\approx$35%. We tested the addition of the conscience mechanism to GRLVQI using a 23- and 35-class problem where we report nearly 2%, and an impressive 5.6%, increase in classification accuracy over GRLVQ. The MED classifier was used for an independent evaluation of the quality of the extracted feature sets, by comparing MED classification results using all (194) spectral channels, to MED classification results using selected spectral channels based

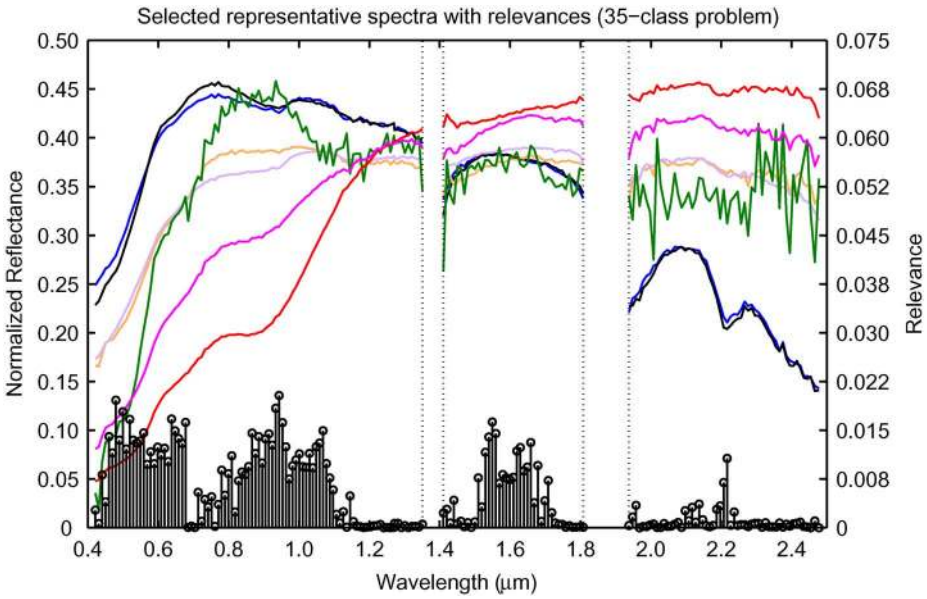

Fig. 10. Average representative spectra of classes A (red), G (green), H (orange), L (magenta), O (purple), Q (black), and R (blue). Relevance factors are the averages of three jackknife runs obtained by GRLVQI (black stem plot) for the 35-class problem. Classes were selected to show largest diversity. The dotted vertical lines indicate data fallout due to saturation of the water bands.

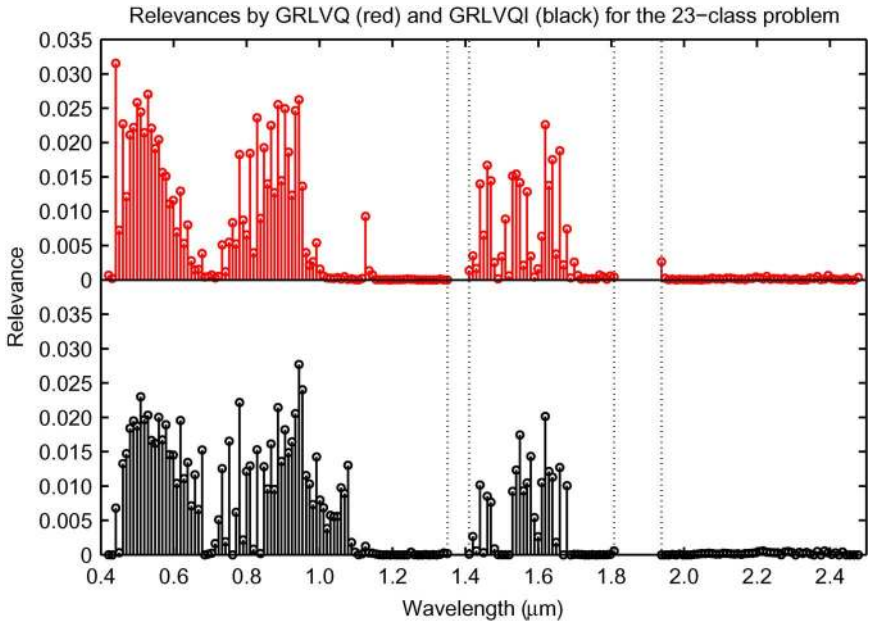

Fig. 11. Relevance factors from GRLVQI (black stem plot) with those obtained by GRLVQ (red stem plot) for the 23-class problem. The dotted vertical lines indicate data fallout due to saturation of the water bands.

on GRLVQ(I) computed relevance. In all cases, classification accuracy improved using the reduced feature set.

While the results presented here are strong and we do not know of any previous work with similar achievement in hyperspectral data analysis, there remains a suite of open issues that could or should be addressed in subsequent research. We discuss several such issues in the following. These, in our view, could contribute to further improvements.

Using five prototypes per class produced good classification results in our simulations. However, it might be more profitable to vary the number of prototypes across classes based on class characteristic. Characteristics such as class size, geometric relationship between classes, and modality of the class distribution can each influence this design consideration. One natural thought is to use a SOM in a preprocessing phase and derive an ideal distribution of the number of prototypes for each class. One can map labeled training samples to prototype vectors in the SOM and observe the number of prototype vectors assigned to each class label. This observed number can be used directly, or in proportion, as the number of prototypes to assign to each class in the GRLVQ(I) network.

Conscience learning in a supervised setting is new and is a major contributing factor to the improved classification performance of GRLVQI. The sensitivity of GRLVQI to the choice of conscience parameters is unclear, however. For example, we show good classification results using the same parameters for

TABLE VI
INDEPENDENT VERIFICATION OF FEATURE SETS SELECTED BY GRLVQI AND OF THOSE SELECTED BY GRLVQ USING THE MED CLASSIFIER. ACCURACY IS AVERAGED OVER THREE JACKKNIFED RUNS, IN EACH OF WHICH THE ACCURACY IS CALCULATED AS THE MEAN OF THE INDIVIDUAL CLASS ACCURACIES

Relevance selected features used in the MED classifier		Accuracy
7-Class Problem	GRLVQI Features	99.0%
	GRLVQ Features	99.0%
	Baseline MED	98.5%
23-Class Problem	GRLVQI Features	96.2%
	GRLVQ Features	96.2%
	Baseline MED	93.3%
35-Class Problem	GRLVQI Features	92.1%
	GRLVQ Features	92.1%
	Baseline MED	90.8%

all classes, but we recognize that one might achieve better classification or faster convergence by considering separate conscience parameters for each class based on characteristics we described previously. This work also leaves the use of different (global) sets of conscience parameters unexplored.

We demonstrated the power of GRLVQ(I) for classification of hyperspectral data. GRLVQ(I) was able to identify a significantly reduced feature set that improved classification accuracy for both GRLVQ(I) itself and with an independent MED classifier, in comparison to using all available (194) spectral features. Although optimal classification accuracy is a requirement built into the feature extraction process, it remains unknown if a better set of features (i.e., one yielding a higher classification accuracy) can be extracted by employing a different cost function in GRLVQ(I), or with any other method.

The MED classifier is a relatively simple classifier which benefited greatly from the feature extraction capabilities of GRLVQ(I). Other classifiers could also benefit from the extracted feature set and may yield improved classification results. For example, the ML classifier may be unemployable for high-dimensional data because of the lack of sufficient number of training samples, which number is dependent on the number of input features used (see a case study in [8] and [9]). ML could become applicable after GRLVQ(I) feature extraction. We used the MED as independent evaluator because of its cost effectiveness and because it does not require guesses for additional parameters (such as prior probabilities, etc.). We surmise that more sophisticated classifiers could benefit similarly, and perhaps produce even better classification than those obtained by GRLVQ(I). While deemed computationally expensive for the present initial evaluation of GRLVQ(I) feature extraction, our prime candidate for a follow-up experiment with more advanced independent classifiers will be the hybrid neural architecture used by Merényi [8], [9]. That network was used in earlier work for a benchmark classification of the entire noisy, real AVIRIS image from which the samples used in this study were selected, and it achieved over 90% classification accuracy (better than an MED classifier produced for the same) using the unreduced original 194 spectral features. Thus, one can investigate if this more sophisticated classifier will yield an improved accuracy with the same reduction of features as presented here, or equal accuracy with less number of retained features.

From the original 194 spectral features, GRLVQ(I) discovers a significantly reduced set relevant for classification. As the classification problem becomes more complex, additional features are required to distinguish between increasing number of classes. This observation is consistent when comparing classification results for GRLVQ(I). The highly correlated spectral bands appear to limit the reduction of features GRLVQ(I) requires for classification. This is seen from the spectral plots with relevance factors in Figs. 8–10: the relevance values next to large (small) relevance values are also large (small). An important and interesting future direction is to research ways to address the multiple band correlations as part of the relevance learning, with the objective of achieving the same high-quality classifications, but with even fewer retained features.

The theory of generalization bounds for LVQ classifiers has progressed in recent years. These works, however, are based on a two-class assumption and do not offer much insight to network design and performance indication for more difficult classification problems. We remain hopeful that future developments will arm researchers with the much needed tighter design and analysis theory to assist with the rich classification problems presented by today's remote sensing community.

ACKNOWLEDGMENT

The authors would like to thank Dr. T. Villmann for many thought-provoking discussions on GRLVQ and experimental results and Dr. B. Hammer for helpful private communications. The views expressed in this paper are those of the author and do not reflect the official policy or position of the United States Air Force, the U.S. Department of Defense, or the U.S. Government.

REFERENCES

[1] B. Hammer and T. Villmann, "Generalized relevance learning vector quantization," *Neural Netw.*, vol. 15, pp. 1059–1068, 2002.
[2] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Berlin, Germany: Springer-Verlag, 2001.
[3] J. A. Benediktsson, J. R. Sveinsson, and K. Arnason, "Classification of very-high-dimensional data with geological applications," in *Proc. Multisens. Airborne Campaign Eur.*, Oct. 1994, pp. 13–18.
[4] T. Moon and E. Merenyi, "Classification of hyperspectral images using wavelet transforms and neural networks," in *Proc. SPIE: Wavelet Appl. Signal Image Process. III*, A. F. Laine, M. A. Unser, and M. V. Wickerhauser, Eds., Sep. 1995, vol. 2569, pp. 725–735.
[5] E. Merényi, R. Singer, and W. H. Farrand, "Classification of the LCVF AVIRIS test site with a Kohonen artificial neural network," in *Proc. 4th Airborne Geosci. Workshop*, Washington, DC, Oct. 25–29, 1993, pp. 117–120.
[6] X. Zhang, N. H. Younan, and C. G. O'Hara, "Wavelet domain statistical hyperspectral soil texture classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 615–618, Mar. 2005.
[7] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 5, pp. 461–473, May 1995.
[8] E. Merényi, "Precision mining of high-dimensional patterns with self-organizing maps: Interpretation of hyperspectral images," in *Quo Vadis Computational Intelligence: New Trends and Approaches in Computational Intelligence. Studies in Fuzziness and Soft Computing*, P. Sin-cak and J. Vascak, Eds. Berlin, Germany: Physica-Verlag, 2000, vol. 54 [Online]. Available: http://www.ece.rice.edu/~erzsebet/publications.html
[9] E. Merényi, W. H. Farrand, J. V. Taranik, and T. B. Minor, "Classification of hyperspectral imagery with neural networks: Comparison to conventional tools," *Photogrammetric Eng. Remote Sens.*, Jul. 2007, Special Issue on Artificial Intelligence, submitted for publication.

[10] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neural Information Processing Systems 8*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. Cambridge, MA: MIT Press, 1996, pp. 423–429.
[11] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin analysis of the LVQ algorithm," in *Advances in Neural Information Processing Systems 15*, S. T. S. Becker and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2002, pp. 462–469.
[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.
[13] A. Sato and K. Yamada, "A formulation of learning vector quantization using a new misclassification measure," in *Proc. 14th Int. Conf. Pattern Recognit.*, Aug. 1998, vol. 1, pp. 322–325.
[14] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.
[15] D. DeSieno, "Adding a conscience to competitive learning," in *Proc. IEEE Int. Conf. Neural Netw. I*, New York, Jul. 1988, pp. 117–124.
[16] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive learning algorithms for vector quantization," *Neural Netw.*, vol. 3, pp. 277–290, 1990.
[17] B. Hammer, M. Strickert, and T. Villmann, "Learning vector quantization for multimodal data," in *Proc. Int. Conf. Artif. Neural Netw.*, London, U.K., 2002, pp. 370–376.
[18] B. Hammer, M. Strickert, and T. Villmann, "Supervised neural gas with general similarity measure," *Neural Process. Lett.*, vol. 21, no. 1, pp. 21–24, 2005.
[19] F. Poirier and A. Ferrieux, "DVQ: Dynamic vector quantization-An incremental LVQ," in *Proc. Int. Conf. Artif. Neural Netw.*, 1991, pp. 1333–1336.
[20] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 1st ed. New York: Academic, 1999.
[21] B. Hammer, M. Strickert, and T. Villmann, "On the generalization ability of GRLVQ networks," *Neural Process. Lett.*, vol. 21, no. 2, pp. 109–120, Apr. 2005.
[22] M. Su, T. Liu, and H. Chang, "An efficient initialization scheme for the self-organizing feature map algorithm," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 1999, vol. 3, pp. 1906–1910.
[23] A. Sato, "An analysis of initial state dependence in generalized LVQ," in *Proc. 9th Int. Conf. Artif. Neural Netw.*, Sep. 1999, vol. 2, pp. 928–933.
[24] R. O. Green, in *Summaries 6th Annu. JPL Airborne Geosci. Workshop, 1. AVIRIS Workshop*, Pasadena, CA, Mar. 4–6, 1996.
[25] E. Merényi, R. Singer, and J. Miller, "Mapping of spectral variations on the surface of mars from high spectral resolution telescopic images," *ICARUS*, vol. 124, pp. 280–295, 1996.
[26] E. Merényi, J. Taranik, T. Minor, and W. Farrand, "Quantitative comparison of neural network and conventional classifiers for hyperspectral imagery," in *Summaries 6th Annu. JPL Airborne Geosci. Workshop, 1. AVIRIS Workshop*, R. Green, Ed., Pasadena, CA, Mar. 4–6, 1996, vol. 1, pp. 171–174.
[27] T. Villmann, E. Merényi, and B. Hammer, "Neural maps in remote sensing image analysis," *Neural Netw.*, vol. 16, pp. 389–403, 2003.
[28] P. Bartlett and S. Mendelson, "Radamacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, 2002.

Michael J. Mendenhall received the B.S. degree in computer engineering from Oregon State University, Corvallis, in 1996, the M.S. degree in computer engineering from the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, OH, in 2001, and the Ph.D. degree in electrical engineering from Rice University, Houston, TX, in 2006.

Currently, he is an Assistant Professor of Electrical Engineering at the AFIT. His research interests are in precise clustering and classification of hyperspectral images, hyperspectral-aided tracking and identification, hyperspectral signature modeling, automatic target recognition, and the broader topic of computational intelligence.

Erzsébet Merényi (M'98–SM'05) received the M.Sc. degree in mathematics and the Ph.D. degree in computer science from the Szeged (Attila József) University, Szeged, Hungary, in 1975 and 1980, respectively.

Currently, she is a Research Professor at the Electrical and Computer Engineering Department, Rice University, Houston, TX. She was a Staff Scientist at the Lunar and Planetary Laboratory, University of Arizona, Tucson. Her interests include artificial neural networks, self-organized learning, manifold learning, segmentation and classification of high-dimensional patterns, data fusion, data mining, knowledge discovery, and application to information extraction from multi- and hyperspectral remote sensing imagery from Earth and space science missions, from hyperspectral medical imagery, and from multivariate medical data.