

## Authors' Reply

Jianchang Mao and Anil Jain

Wang and Xia make a valid point in their comments on our paper that the cost defined in (1) in their comments (not numbered in our original paper) is constant if the Mahalanobis distance is *directly* used. Although we did not point out this problem in our original paper, we mentioned a related problem (the third difficulty associated with using the Mahalanobis distance in the k-means algorithm; see p. 18 in our original paper) which served as one of our motivations for our proposed algorithm. Moreover, as Wang and Xia also noticed, our proposed algorithm (HEC algorithm or k-means algorithm with the regularized Mahalanobis distance) was *not* derived from the objective of minimizing the cost (in fact, we never refer to this definition later in our paper). This approach is not uncommon in neural networks and pattern recognition literature. Many learning algorithms do not have a precise mathematical objective function (e.g., Kohonen's self-organizing map algorithm), but still work well in practice.

The comments by Wang and Xia failed to point out that our HEC clustering algorithm used a regularized Mahalanobis distance instead of the standard Mahalanobis distance. It is the regularized Mahalanobis distance which plays an important role in realizing hyperellipsoidal clusters.

The regularized Mahalanobis distance, which is a weighted combination of Euclidean distance and standard Mahalanobis distance, was motivated to deal with the three difficulties (p. 18 in our original paper) associated with using the Mahalanobis distance in the k-means algorithm. The third difficulty states that the k-means clustering algorithm with the Mahalanobis distance tends to produce unusually large or unusually small clusters. This is due to the fact that the Mahalanobis distance is inversely weighted by the "spread" of a cluster (which also leads to a constant cost). However, in our original paper, we did not analyze the role of the regularized Mahalanobis distance in the context of the cost.

Now, let us examine the effect of the regularized Mahalanobis distance on the cost. The Euclidean part of the regularized Mahalanobis distance makes the cost *nonconstant* with respect to different clusterings, while the standard Mahalanobis distance part remains constant. This nonconstant cost indicates that certain clusters (not necessarily hyperellipsoidal) are realizable even if the clustering algorithm is directly derived from the cost minimization. In the HEC clustering algorithm, the initial clustering (partition) is obtained using the Euclidean distance ( $\lambda = 1.0$  in the regularized Mahalanobis distance). The hyperellipsoidal clusters, on the other hand, are achieved through local adjustments under the constraint that is embedded in k-means clustering procedure, i.e., a pattern is always assigned into its nearest cluster center in term of the regularized Mahalanobis distance. The final clustering does not usually deviate much from the initial clustering except for the local shape of each cluster.

Manuscript received June 27, 1997.

J. Mao is with the IBM Almaden Research Center, San Jose, CA 95120 USA.

A. Jain is with the Department of Computer Science, Michigan State University, East Lansing, MI 48824 USA.

Publisher Item Identifier S 1045-9227(97)07528-0.

It is well known that (9) in the comments may have multiple solutions. With the HEC clustering algorithm, the solution which is closest to the initial clustering formed by the k-means algorithm with the Euclidean distance will be eventually obtained.

In summary, the comments made by Wang and Xia together with this response provide some new insights into the behavior of our HEC clustering algorithm. It further confirms that the HEC algorithm is a useful tool for understanding the structure of multidimensional data.

## Corrections to "Adaptive Critic Designs"

Danil V. Prokhorov and Donald C. Wunsch, II

Corrections are noted to the following parts of the above-mentioned paper.<sup>1</sup>

The Fig. 7 caption should read as follows.

Adaptation in ADGDHP. The critic network outputs the scalar  $J$  and two vectors,  $\lambda_R$  and  $\lambda_A$ . The vector  $\lambda_R(t+1)$  propagates back through the model, then it is split in two vectors. One of them goes into the square summator to be added to the vector  $\partial U(t)/\partial R(t)$  and to the rightmost term in (18) (not shown). The second vector is added to the vector  $\partial U(t)/\partial A(t)$  in another summator. Both of these summators produce two appropriate error vectors  $E_2(t)$ , as in (19) and (20). According to (3), the right oval summator computes the error  $E_1(t)$ . Two error vectors  $E_2(t)$  and the scalar  $E_1(t)$  are used to train the critic network. The action network is adapted by the direct path  $\lambda_A(t+1)$  between the critic and the action networks.

After (18) the third expression should read as follows:

$$\lambda_{A_k}^*(t) = \frac{\partial J(t+1)}{\partial A_k(t)} + \frac{\partial U(t)}{\partial A_k(t)}.$$

Equation (22) should read as follows:

$$A^*(t) = \arg \min_A J[R(t), A(t)]. \quad (22)$$

Equation (24) should read as follows:

$$U[t] = \frac{1}{2} \{ [y_1(t+1) - y_1^*(t+1)]^2 + [y_2(t+2) - y_2^*(t+2)]^2 + [y_2(t+3) - y_2^*(t+3)]^2 \}. \quad (24)$$

The authors are with the Applied Computational INtelligence Laboratory, Department of Electrical Engineering, Texas Tech University, Lubbock, TX 79409 USA.

Publisher Item Identifier S 1045-9227(97)08933-9.

<sup>1</sup>D. V. Prokhorov and D. C. Wunsch, II, *IEEE Trans. Neural Networks*, vol. 8, pp. 997-1007, Sept. 1997.