

Fuzzy concepts in vector quantization training

Stefano Rovetta ^{a,c}, Francesco Masulli ^{b,c}

^a*University of Genova, Italy – Department of Computer and Information Sciences*

^b*University of Pisa, Italy – Department of Computer Science*

^c*The National Institute for the Physics of matter, Italy*

Abstract

Vector quantization and clustering are two different problems for which similar techniques are used. We analyze some approaches to the synthesis of a vector quantization codebook, and their similarities with corresponding clustering algorithms. We outline the role of fuzzy concepts in the performance of these algorithms, and propose an alternative way to use fuzzy concepts as a modeling tool for physical vector quantization systems, Neural Gas with a fuzzy rank function.

Key words: Vector quantization; Neural Gas; Fuzzy clustering.

1 Introduction

The problem of clustering [1] is often addressed with the partitive, centroid-based approach of the *c*-Means procedure and many other derived algorithms. In this approach clustering is viewed as *finding the reference vectors (centroids) which best explain the input data distribution according to some cost criterion*.

Vector quantization [2,3] is a different technical problem, which can be stated as follows: *find the reference vectors (codevectors) which approximate with the minimum error the input data according to some distortion criterion*. Usually the problem is also constrained by some resource limits. This is the rationale for the so called Rate/Distortion and Distortion/Rate approaches.

In this work, we analyze some approaches to the synthesis of a vector quantization codebook, and their similarities with corresponding clustering algorithms. We outline the role of fuzzy concepts (such as membership in more than one Voronoi polyhedron) in the performance of these algorithms. Then, we propose an alternative use of the fuzzy paradigm in the vector quantization training algorithm by Martinetz *et al.*, the “Neural Gas” [4].

2 Clustering, vector quantization, and fuzzy concepts

We have outlined some differences between clustering and vector quantization, yet the synthesis of a codebook for vector quantization is often approached with algorithms derived from c -Means (a standard clustering technique). One usual difference is that, since vector quantization is typically adopted for large-dimensional, large-sized training sets, minimization is performed by stochastic gradient descent (*online* training) rather than by deterministic (or *batch*) algorithms. This is because the curse of local minima is worsened by the large dimensionality and codebook size. Stochastic optimization helps escaping local minima by adding errors (due to random sampling of patterns) to the current estimate of the cost function. Therefore there is a nonzero probability of taking steps in directions other than that of the “closest” local minimum.

Several clustering algorithms have been modified in the direction of incorporating fuzzy concepts (starting with the Fuzzy c -Means algorithm [5]). A review of fuzzy concepts in clustering is provided in [6,7]. In the large majority of cases, fuzziness means that any point can belong to more than one cluster.

The relationship between the clustering and quantization problems is of a geometrical nature. In both cases, the input space is partitioned by a *Voronoi tessellation* [8], representing regions of data sharing similar properties by means of a single reference point or site or (in vector quantization jargon) codevector. However, there is a distinction in the goal to be achieved. Generally speaking, in clustering data points belonging in a single region should be the largest group of data that can be reasonably gathered in a single cluster (clusters should be as few as possible to enable understanding the structure of data), whereas in vector quantization points in a region must be so similar that the approximation error obtained by substituting data with codevectors is negligible. Thus in the case of clustering codevectors should be as many as possible, within the resource limits imposed by the overall system design.

The introduction of a fuzzy membership has a twofold meaning in clustering. On one side, data can be partially belonging to more than one cluster, and this has a conceptual interpretation: it is possible to analyze and quantify whether points are clearly clustered or there is any ambiguity in cluster attribution.

On the other side, fuzziness is a way to fight local minima during optimization. In vector quantization, the first aspect is irrelevant, since at the end of training a crisp decision must always be made. The other aspect is more important, since in the typical vector quantization application local minima are a serious issue.

3 Codebook design (training)

In the following we review how typical algorithms for the synthesis of vector quantization codebooks introduce fuzzy concepts in the minimization procedure, and what is their

effect. We will assume that N training points (individually denoted with \mathbf{x}) of dimensionality d are used to design a codebook $\{\mathbf{y}_1, \dots, \mathbf{y}_c\}$ of c reference points. The distortion assumed is the Euclidean distance $d_i = \|\mathbf{x} - \mathbf{y}_i\|$.

3.1 Lloyd's method

The classical approach is Lloyd/MacQueen's method [9–11], the standard c -Means clustering procedure. The k -th input vector is attributed to the Voronoi polyhedron defined by reference vector \mathbf{y}_i if $u_{ik} = 1$, where u_{ik} is a crisp membership value which is 1 if $d_{ik} = \min\{d_{1k}, \dots, d_{Nk}\}$ and 0 for all other reference vectors. The closest reference vector for a data point will be called the “winner” for that point. The updating rule is:

$$\mathbf{y}_i^{(t+1)} = \frac{\sum_{k=1}^N \mathbf{x}_k u_{ik}}{\sum_{k=1}^N u_{ik}} \quad (1)$$

This algorithm finds the minimum of a cost function based on the mean square error as a distortion criterion. Its well-known drawback lies in the huge number of local minima (for practical d and N).

3.2 MacQueen's method

The on-line version of c -Means training, which is due to MacQueen, transforms the Picard iteration of the standard version, in which at each step a necessary minimum condition is satisfied, in a stochastic optimization process. Input vectors are randomly selected, adding noise to the cost function, now optimized on the average. The updating rule is therefore:

$$\mathbf{y}_i^{(t+1)} = \mathbf{y}_i^{(t)} + \eta^{(t)} u_{ik} (\mathbf{x}_k - \mathbf{y}_i) \quad (2)$$

where t indexes the training steps, $\eta^{(t)}$ is an updating coefficient, and k is a random function of t .

Convergence is usually much slower, but local minima are escaped thanks to the “statistical” behaviour of the updating procedure, which does not necessarily reduce the cost at each step and therefore does not necessarily get trapped into sub-optimal basins.

The law for varying $\eta^{(t)}$ to ensure convergence (annealing schedule) has been studied in [12]. MacQueen [10] adopts an individual coefficient for every reference vector, equal to $1/t_i$ where t_i is the number of updates for reference vector \mathbf{y}_i so far, thus retaining the exact equivalence between the online and batch versions of c -means. Ritter *et al.* [13] propose instead an exponential decay rate $\eta^{(t)} = \eta_i (\eta_f/\eta_i)^{t/t_{\max}}$ from η_i to η_f in t_{\max} steps. This law has been used also in the Neural Gas algorithm.

3.3 Fuzzy c -Means

The most popular algorithm for clustering in the fuzzy framework, the “Fuzzy c -Means” [5] or “Fuzzy ISODATA” [14], has no direct counterpart in the vector quantization practice. Here the standard (crisp) c -Means membership is replaced by a fuzzy membership defined as a function of the point-prototype distance:

$$u_{jk} = \left[\sum_{l=1}^c \left(\frac{d_{jk}}{d_{jl}} \right)^{1/(m-1)} \right]^{-1} \quad (3)$$

so that membership values are no longer 0 or 1, and each prototype turns out to be a *weighted* mean of its cluster.

3.4 Maximum Entropy approach (the Deterministic Annealing method)

The maximum entropy approach of the Deterministic Annealing technique by Rose [15] builds on a different concept. Here a fuzzy membership in clusters is introduced by substituting the “min” selection criterion, by which a single reference vector is selected for updating on a minimum-distance basis, with a “softmin” criterion:

$$u_i = \frac{e^{-d_i/\beta}}{\sum_{j=1}^c e^{-d_j/\beta}} \quad (4)$$

The parameter β governs the fuzziness of this criterion; for $\beta \rightarrow 0$ it turns back into the crisp “min” criterion. The Deterministic Annealing approach is a sequence of deterministic minimizations (made by Picard iterations), with β decreasing at each minimization. Therefore the first minimizations are done with a high degree of fuzziness, that is, high β (with few local minima), whereas the last minimizations, with $\beta \rightarrow 0$, are potentially subject to local minima, but they take advantage of the good initialization provided by previous steps. (This is a classical simulated annealing-type procedure, where β acts as the temperature parameter.)

3.5 Possibilistic approach

Another popular fuzzy clustering approach which is not commonly used in vector quantization practice is the Possibilistic Approach by Krishnapuram and Keller [16]. Here a higher level of fuzziness is introduced by relaxing the requirement of memberships to all prototypes for each point summing up to 1, which is enforced in all other methods. This changes considerably the principle of operation of the method, and is not compatible with vector quantization goals.

3.6 The Neural Gas algorithm

The Neural Gas algorithm by Martinetz *et al.* [4] combines fuzzy membership in partitions with stochastic minimization. This algorithm has the interesting feature that membership in a Voronoi polyhedron is not defined as a direct function of the distance from the data point to the reference vector, as in previously cited methods. Rather, it is a function of its rank with respect to the list of distances from all reference vectors. Distance d_i has the rank ρ_i in the set $\{d_1, \dots, d_N\}$ when ordered decreasingly with respect to values, and this value can be written in an algebraic fashion as:

$$\rho_i = \sum_{j=1}^c \theta(d_i - d_j) \quad (5)$$

$\theta(x)$ is the Heaviside step function, taking on the values 0 for $x < 0$, 1 for $x > 0$, and 0.5 for $x = 0$. This extension is needed in the case of ties, very uncommon if the distances are real numbers; however this is the standard way to deal with ties in rank tests (such as Spearman's rank correlation or Kendall's rank correlation and coefficient of concordance). Notice that $\rho_{\text{winner}} = 0$ rather than 1, so $\rho_i \in \{0, \dots, c - 1\} \forall i \in \{1, \dots, c\}$.

The membership of the data point to the i -th encoding polyhedron is:

$$u(\mathbf{x}) = e^{-\rho_i/\lambda} \quad (6)$$

where λ is a parameter which is annealed (made smaller) during training, thereby progressively reducing the extent to which reference vectors, other than the nearest (the "winner"), are included in the updating process.

The annealing of the two parameters (λ , influence of prototypes other than the "winner", and learning coefficient) can be interpreted from the standpoint of learning machine capacity. When vectors other than the winner get updated a correlation is introduced between reference vectors, thus effectively reducing the learning capacity of the vector quantizer. As the annealing proceeds, the range of the correlation shrinks gradually, and the capacity is correspondingly increased; however, at the same time the learning coefficient is reduced, so that it is progressively more difficult to fall into local minima. (To relate vector quantization, an approximation procedure, to the theory of learning capacity it is necessary to adopt a threshold-based criterion. This analysis is introduced in [17].)

3.7 Kohonen's Self-Organizing Maps

In this review, we must also mention Kohonen's Self Organizing Maps [18], in which fuzziness is of the same nature as in the Neural Gas technique, that is, in the influence of non-winners on the update of the winner. This method is neither a clustering algorithm nor a vector quantization algorithm (it is conceived as a multidimensional scaling technique),

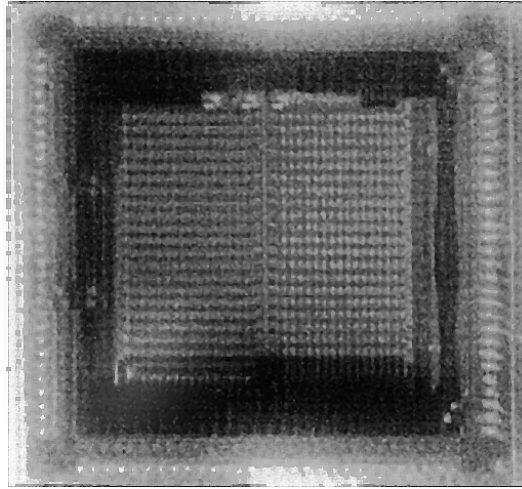


Fig. 1. An analog VLSI realization of a vector quantization encoder.

but we cannot avoid mentioning it due to its importance and influence on the subject (and particularly on many vector quantization algorithms, including Neural Gas).

3.8 Interval Vector Quantization

A different way to include fuzziness into vector quantization is making the *codevectors themselves* fuzzy. Although adopting this formulation can lead to a computationally inefficient algorithm, this approach can be simplified by representing uncertainty by means of interval values. This has been done in [19]. With reference to the image compression application, uncertain codevectors can be defuzzified by applying some additional criterion (for instance, regularity or smoothness of the overall reconstructed image), which acts as a constraint and helps obtaining better perceived quality.

4 A fuzzy model of the ranking function

The performance of the Neural Gas algorithm is remarkably good, as found in previous research by the present and other authors. This is probably due to the combination of fuzzy membership, stochastic optimization and robust evaluation through ranking. Therefore it is not surprising that this algorithm has been used as the basis for improvements [20] [21] as well as hardware implementations [22]. In the case of analog hardware implementations, other algorithms either perform worse, as we have reviewed, or imply very complex circuit structures. The Neural Gas seems the best choice in view of this trade-off, also because the sorting step can be simplified with little performance loss [23].

In analog hardware, when the functions implemented are non-ideal there can be a variable effect on training performances. In particular, the rank function (5) uses the Heaviside step as a crisp distance comparison.

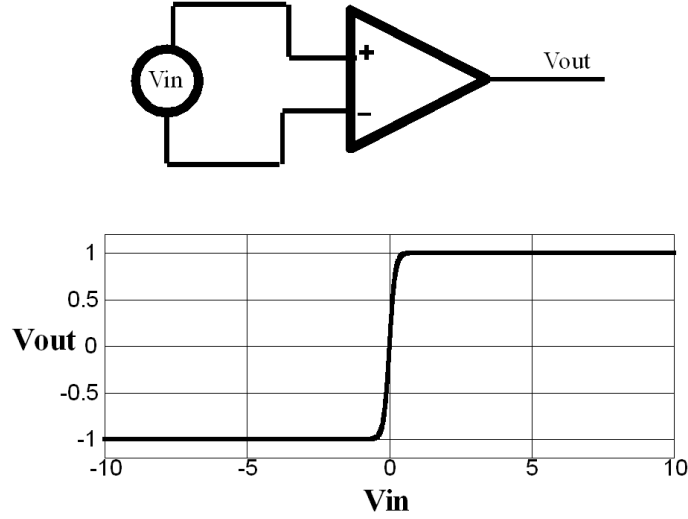


Fig. 2. A low-performance operational amplifier implements an approximate step function.

The step function in analog hardware is simply built by means of a saturating amplifier with large gain, which means typically an open-loop operational amplifier. However, Equation (5) has a c^2 space complexity, so circuit topologies should be made very inexpensive in terms of silicon area. This means very simple topologies (typically two stages). Consequently, the operational amplifier will feature a finite gain which implies a deviation from the ideal behavior.

The input-output relationship of an operational amplifier at middle frequencies is a hyperbolic tangent saturating (approximately) at the + and – power voltages. This suggests a very natural fuzzy model for the non-ideal rank function.

In a fuzzy perspective, it is more natural to define the relation “larger” among two (conventional) numbers as a degree to which one number is larger than another. We should mention that the problem of ranking fuzzy quantities has been reviewed for instance by Bortolan and Degani [24]. and, more recently, by Wang and Kerre [25,26]. However, we are not dealing with fuzzy quantities, but with a *fuzzy evaluation* of crisp quantities. This approach is reasonable in very common situations such as presence of noise or other uncertainties in the measure of signals. In this case, two values which are very close cannot be reliably ranked, and a statement such as “ a is larger than b ” is more naturally expressed in fuzzy terms.

As a numerical illustration, suppose that we are to compare (a) $d_1 = 3$ with $d_2 = 4$, and (b) $d_1 = 3$ with $d_2 = 3.01$. Clearly in both case (a) and case (b) we can rightfully say that $d_2 > d_1$, but it is also clear that in (a) this is “more true” than in (b). With a given level of uncertainty (for instance due to a given quantity of additive noise) it is also possible to quantify how much (a) or (b) are “true”.

Therefore, we can make the following substitution:

$$\theta(d_j - d_i) \approx \frac{1}{1 + e^{(d_j - d_i)/\beta}} \quad (7)$$

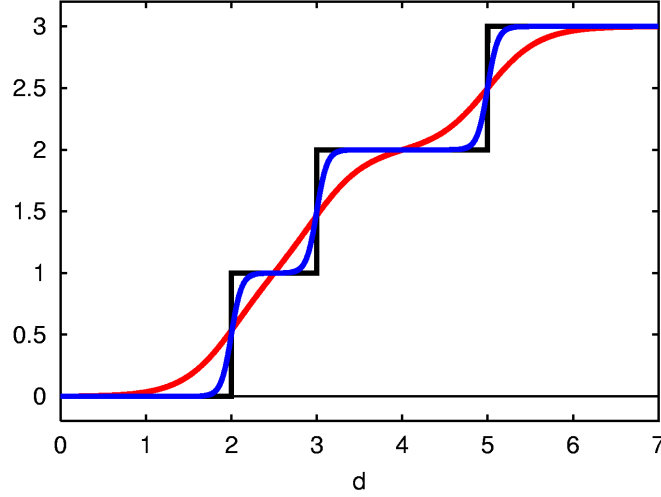


Fig. 3. Comparing crisp and fuzzy rank functions.

and

$$\frac{1}{1 + e^{(d_j - d_i)/\beta}} \xrightarrow{\beta \rightarrow 0} \theta(d_j - d_i) \quad (8)$$

so the computation of fuzzy rank can be expressed as

$$\rho_j = \sum_{i=1, j \neq i}^n \frac{1}{1 + e^{(d_j - d_i)/\beta}} \quad (9)$$

The parameter β here acts as a fuzzification parameter, such that for large β the ranking function is definitely fuzzy, while for $\beta = 0$ we obtain the original, crisp ranking function.

The two expressions (5) and (9) for the rank function $\rho(\cdot)$ are compared in a simple example, illustrated in Figure 3, where the following set of values is used: $\{d, 2, 3, 5\}$. The diagram is a plot of $\rho(d)$ (in the two expressions, crisp and fuzzy) for d in the range $[0, 7]$. Two plots are shown for the fuzzy expression, one for $\beta = 0.05$ and another for $\beta = 0.25$ (smoother).

The fuzzy ranking function is directly implemented by the op-amp-based circuitry outlined above. The fuzzification parameter is the inverse of the amplifier gain (the crisp and fuzzy version coincide for gain $\rightarrow \infty$ or for $\beta \rightarrow 0$). Therefore the fuzzy Neural Gas is a realistic model for the hardware implementation of the algorithm.



Fig. 4. The “Lena” image.

5 Experimental performance

5.1 *Experimental setup*

The fuzzy model for the Neural Gas has been tested by comparison with the standard version on some problems, both artificial and real:

- (1) Centers-only (toy problem, very trivial): place three codevectors on three points. For initial “consistency checks”.
- (2) Centers-plus-noise (toy problem): place three codevectors on a set of points generated by a superposition of three Gaussians plus 60% random points.
- (3) Lena (real problem). Vector quantization of the standard benchmark image “Lena”, shown in Figure 4, with codebooks of size 16 and 256.
- (4) Four images (real problem). Vector quantization of more benchmark images, shown in Figure 5.

The training of both algorithms was performed with identical initialization parameters (for the scheduling of updating coefficient and of range of influence of non-winners,

5.2 *Experimental results*

The first problem was used to ensure that the training steps were not too different, to validate the software (written in C). For $\beta = 0$ the two algorithms are indeed identical.

The second problem highlighted that, for low values of β , there are no significant differences in performance between the two algorithms. In some experiments the fuzzy version



Fig. 5. Four benchmark images (from <http://links.uwaterloo.ca/bragzone.base.html>).

outperformed the standard version, but this is not a typical behavior.

The training on the Lena image was a test of these outcomes on a real problem. In Figure 6 is shown a typical training trace (mean square error versus training steps), put on a logarithmic scale to show that the two traces are different, but converge to the same solution. The thin trace is standard Neural Gas and the thick trace (only slightly different in some locations) is the fuzzy modification.

Finally, the four additional images, which are greyscale and are of size 256x256, were used to confirm the previous results. The images were obtained from the “Waterloo Repertoire”, available online at <http://links.uwaterloo.ca/bragzone.base.html>. Codevector sizes used are 16, 64 and 256. Results on the concordance of the two methods are outlined in the following table. For each test, the maximum deviation of the fuzzy version over the standard version (in percentage of RMS error) is indicated. The final codebooks have always been found to be equal according to the following definition. Two codebooks A and B are considered equal if, for any codevector in codebook A, the closest codevector in codebook B is within a preselected distance threshold. This threshold has to be selected case by case, taking into account codebook cardinality and making it less than the minimum distance between two codevectors of any codebook.

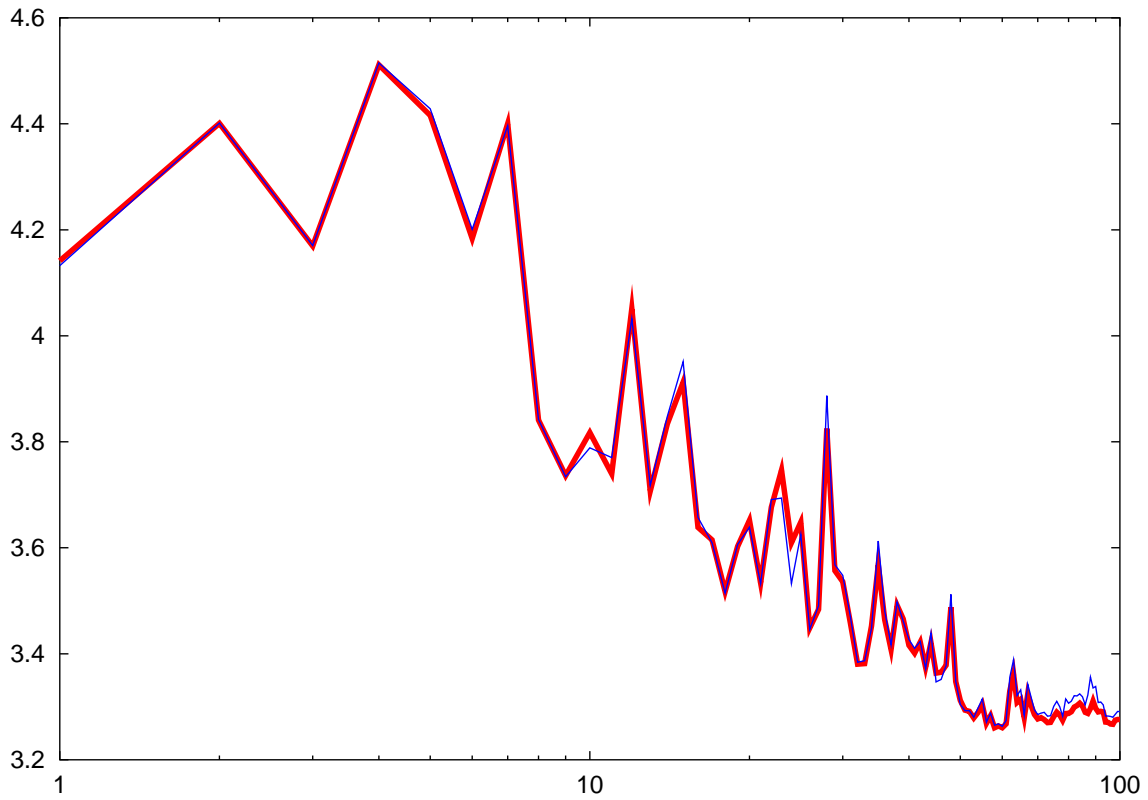


Fig. 6. Trace of mean square error during training on the “Lena” image.

Test	Max. discordance in RMS
goldhill 16	0.8%
goldhill 64	0.9%
goldhill 256	1.3%
bridge 16	0.0%
bridge 64	0.5%
bridge 256	0.5%
bird 16	0.2%
bird 64	1.0%
bird 256	2.1%
camera 16	1.7%
camera 64	1.7%
camera 256	1.1%

The remarkable fact that final codebooks were always coincident is a confirmation of the good properties of the Neural gas algorithm, which proves to be stable under the perturbations induced by reasonable values of β .

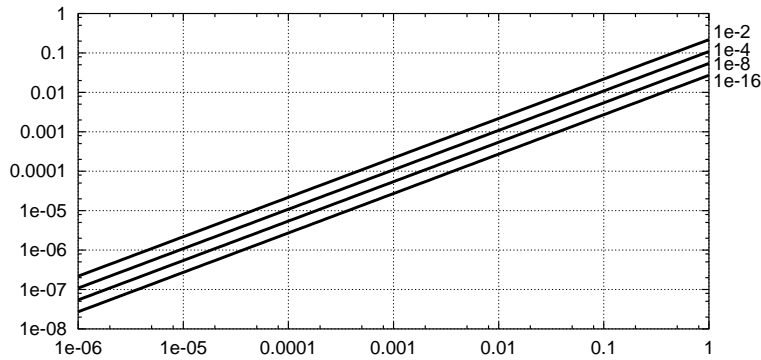


Fig. 7. Nomogram for the calculation of β .

5.3 Choice of parameter β

The acceptable value of β depends linearly on the difference between distances that has to be resolved. The nomogram in Figure 7 is a plot of β versus $\Delta_{ij} = d_i - d_j$ for different values of the accepted error (annotated on the right margin).

A method to select the acceptable β can be based on the distribution of the distance differences in the training set. The Δ_{ij} 's are tabulated and sorted; then a given tolerance is selected (for instance, 1%) and the corresponding quantile is identified on the table. This corresponds to a given value of Δ and therefore (through the nomogram) to the required acceptable value of β .

6 Conclusion

In this paper we have reviewed some uses of fuzzy concepts in vector quantization training. The main application is to enhance convergence, but we have also proposed using fuzzy ranking for the modeling of hardware implementations. The method shows that, for reasonably chosen uncertainty levels (β), the neural gas algorithm is remarkably stable.

Acknowledgments

This work was funded by the Italian National Institute for the Physics of Matter (INFN) and by the Italian Ministry of Education, University and Research under a "PRIN - Cofin2002" grant.

References

- [1] Anil K. Jain and Richard C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1988.

- [2] Allen Gersho, “On the structure of vector quantizers”, *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 157–166, March 1982.
- [3] R.M. Gray, “Vector quantization”, *IEEE Acoustic, Speech and Signal Processing Magazine*, vol. 1, pp. 4–29, 1984.
- [4] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten, “‘Neural gas’ network for vector quantization and its application to time-series prediction”, *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.
- [5] James C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum, New York, 1981.
- [6] A. Baraldi and P. Blonda, “A survey of fuzzy clustering algorithms for pattern recognition. I”, *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 778–785, 1999.
- [7] A. Baraldi and P. Blonda, “A survey of fuzzy clustering algorithms for pattern recognition. II”, *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 786–801, 1999.
- [8] Franz Aurenhammer, “Voronoi diagrams-a survey of a fundamental geometric data structure”, *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.
- [9] S. Lloyd, “Least squares quantization in pcm”, *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.
- [10] J. MacQueen, “Some methods for classification and analysis of multivariate observations”, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. Le Cam and J. Neyman, Eds. University of California, January 1967, vol. I, pp. 281–297.
- [11] Y. Linde, A. Buzo, and R.M. Gray, “An algorithm for vector quantizers design”, *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.
- [12] G. Geman and D. Geman, “Stochastic relaxation, gibbs distribution and bayesian restoration of images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.
- [13] H.J. Ritter, T.M. Martinetz, and K.J. Schulten, *Neuronale Netze*, Addison-Wesley, München, Germany, 1991.
- [14] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters”, *Journal of Cybernetics*, vol. 3, pp. 32–57, 1974.
- [15] Kenneth Rose, “Deterministic annealing for clustering, compression, classification, regression, and related optimization problems”, *Proceedings of IEEE*, vol. 86, no. 11, pp. 2210–2239, November 1998.
- [16] Raghu Krishnapuram and James M. Keller, “A possibilistic approach to clustering”, *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, May 1993.
- [17] David Cohn, Eve A. Riskin, and Richard Ladner, “Theory and practice of vector quantizers trained on small training sets”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 54–65, January 1994.
- [18] T. Kohonen, *Self-Organizing Maps*, Springer, 2001.

- [19] Sandro Ridella, Stefano Rovetta, and Rodolfo Zunino, “IAVQ–interval-arithmetic vector quantization for image compression”, *IEEE Transactions on Circuits and Systems, Part II*, vol. 47, no. 12, pp. 1378–1390, December 2000.
- [20] T. Hoffmann and J.M. Buhmann, “An annealed neural gas network for robust vector quantization”, in *Proceedings of the International Conference on Artificial Neural Networks – ICANN96, Bochum, Germany*, 1996, pp. 151–156.
- [21] Sandro Ridella, Stefano Rovetta, and Rodolfo Zunino, “Plastic algorithm for adaptive vector quantization”, *Neural Computing and Applications*, vol. 7, no. 1, pp. 37–51, 1998.
- [22] Stefano Rovetta and Rodolfo Zunino, “Efficient training of vector quantizers with analog circuit implementation”, *IEEE Transactions on Circuits and Systems, Part II*, vol. 46, no. 6, pp. 688–698, June 1999.
- [23] Fabio Ancona, Sandro Ridella, Stefano Rovetta, and Rodolfo Zunino, “On the importance of sorting in ‘neural gas’ training of vector quantizers”, in *Proceedings of the 1997 International Conference on Neural Networks, Houston, USA*, June 1997, pp. 1804–1808.
- [24] G. Bortolan and R. Degani, “A review of some methods for ranking fuzzy sets”, *Fuzzy sets and systems*, vol. 15, pp. 1–19, 1985.
- [25] W. Wang and E.E.. Kerre, “Reasonable properties for the ordering of fuzzy quantities (I)”, *Fuzzy sets and systems*, vol. 118, pp. 375–385, 2001.
- [26] W. Wang and E.E.. Kerre, “Reasonable properties for the ordering of fuzzy quantities (II)”, *Fuzzy sets and systems*, vol. 118, pp. 386–405, 2001.