The K-Winner Machine Model

Sandro Ridella, member, IEEE, *Stefano Rovetta,* member, IEEE, and *Rodolfo Zunino,* member, IEEE
DIBE - Dept. Biophysical and Electronic Engineering - University of Genoa
Via all'Opera Pia 11a - 16145 Genova - Italy - e-mail: {ridella,rovetta,zunino}@dibe.unige.it

Abstract

A K-Winner Machine (KWM) selects among a family of classifiers the specific configuration that minimizes the expected generalization error. In training, KWM uses unsupervised Vector Quantization and subsequent calibration to label data-space partitions. At run time, KWM seeks the largest set of best-matching prototypes agreeing on a test sample, and provides a local-level measure of confidence. The VC-dim of a KWM classifier is worked out exactly; the resulting small values set tight bounds to generalization performance. The method applies to high-dimensional, multi-class problems with large data sets. Experimental results on both a synthetic and a real domain (NIST handwritten numerals) validate confirm the consistency of the theoretical framework.

I. Introduction

Among the several methods proposed in the literature to estimate generalization error, the formulation based on the Vapnik-Chervonenkis dimension (d_{VC}) [1] offers a general theoretical foundation, especially when one does not want to split the available data set for cross-validation [2]; due to its worst-case analysis, however, Vapnik's approach often yields a wide range of estimated classification accuracy. Thus the domain-oriented perspective of Structural Risk Minimization [3] leads the modern trend in classifier design. In this context, the present research defines a classification model called *K-Winner Machine* (KWM). The classifier design criterion used is to minimize the distortion of data representation; Vector Quantization (VQ) provides the representation paradigm.

In a VQ-based approach, the data space is mapped by a set of reference vectors ("prototypes"), which lie at significant locations and span a partitioning schema over the sample distribution. Many algorithms have been proposed for the prototype-raining problem [4-9]. The basic model of KWMs is actually independent of the specific codebook-positioning algorithm adopted.

The *training* of a KWM classifier is straightforward: first, unsupervised training places VQ prototypes according to the spatial sample distribution, regardless of sample classes; then, prototype calibration exploits class information within each data partition. In *run-time operation*, a sample is classified by checking the classes associated with the K best-matching prototypes: if all the classes agree, the sample is classified accordingly, otherwise it is discarded and processed by the classifier with a lower K.

The exact d_{VC} of a KWM classifier can be computed exactly. This value is a function of the number of prototypes, and is independent of the data dimensionality. Such a property can notably improve classification performance, especially in high-dimensional domains. In addition, the generalization error predicted by using the d_{VC} value exhibits a workable range, hence the process of using d_{VC} to design a classifier becomes feasible. Knowing the VC-dim values of a family of KWM classifiers makes it possible to label each location in the data space by both a confidence level and a bound to the expected generalization error.

Thus, in compliance with the principle of Structural Risk Minimization, a K-Winner Machine includes a set of nested classifiers characterized by increasing d_{VC} values. The principle of operation is to choose, for each test sample, the appropriate KWM-classifier configuration that maximizes the confidence in the classification output, that is, the configuration with the tightest bound to the expected generalization error. The validity of the KWM model is first demonstrated experimentally on an artificial testbed in a 2-D domain, allowing a visual interpretation of results. The practical impact is then evaluated on a real testbed, the NIST handwritten numerals database, involving a high-dimensional data space for a complex, multi-class problem.

II. The K-Winner Classifier

The K-Winner classifier schema adopts a dual-paradigm approach: first, a VQ schema uses available prototypes to render the probabilistic sample distribution; then, each prototype is labelled by the predominant class within its partition. Decoupling vector positioning from subsequent class assignment plays a key role in design

0-7695-0619-4/00 $10.00 © 2000 IEEE

aimed at generalization performance. In particular, this approach prevents an uncontrolled (and practically detrimental) explosion of the classifier's d_{VC}. Run-time operation just requires finding out the K prototypes that best match a test sample, which is classified accordingly only if all the K winning units agree.

The KWM design criterion is to optimize the representation of the data distribution by a Vector-Quantization mechanism. The D-dimensional data space is partitioned by a set of prototypes ("codevectors", "neurons"), $\mathrm{W} = \{\mathbf{w}_n \in \Re^D, n = 1,\ldots,N_h\}$, which lie at "significant" positions in the data space; each prototype covers the samples lying within its associate partition. The process assigning a prototype to each sample follows a best-match criterion minimizing a distortion cost. Euc lidean metrics is usually adopted to measure distortion, hence a data sample, $\mathbf{x} \in \Re^D$, is associated with the prototype, $\mathbf{w}^*(\mathbf{x}) \in \mathrm{W}$, that satisfies $\mathbf{w}^*(\mathbf{x}) = \arg\min_{\mathbf{w}\in W}\{\|\mathbf{x}-\mathbf{w}\|^2\}$. As the actual sample distribution is not known a priori, one usually resorts to an empirical estimation of the overall distortion: a training set, $\mathrm{X} = \{\mathbf{x}_i \in \Re^D, l = 1,\ldots,N_p\}$, drives vector positioning to minimize the empirical cost:

$$\hat{E}(\mathrm{W}) = \frac{1}{N_p}\sum_{i=1}^{N_p}\|\mathbf{x}_i - \mathbf{w}^*(\mathbf{x}_i)\|^2 \tag{1}$$

Even solving the reformulated problem (1) analytically is impractical from a computational point of view; a large variety of iterative approaches have been proposed in the literature for unsupervised VQ training [4-9]. As far as the KWM model is concerned, any of the above mentioned approaches is applicable.

In order to build up a classification machine based on the previous unsupervised representation, assume now that some criterion is available to assign a class to each prototype after the prototype itself has been positioned. This process is conventionally named "calibration." An implementation of the class-assignment process will be detailed in the following, when describing the construction of the KWM. From a general perspective, the calibration mechanism is requested to label the tessellation produced by the VQ training process.

The principle of operation of a KWM lies in checking the agreement of K reference sites in the data space to make a reliable decision on a test location. The twofold decision-making process, involving unsupervised spatial processing and supervised classification, characterizes both the training and the run-time operation. In the following: $\mathrm{C} = \{c^{(k)}, k = 1,\ldots,N_c\}$ is the set of possible sample classes; $\mathrm{W} = \{\mathbf{w}_n \in \Re^D, n = 1,\ldots,N_h\}$ is the codebook; $\mathrm{X} = \{(\mathbf{x}_l, c_l), \mathbf{x}_l \in \Re^D, c_l \in \mathrm{C}, l = 1,\ldots,N_p\}$ is the set of labelled training samples; $P_n = \{\mathbf{x} \in \Re^D : \mathbf{w}^*(\mathbf{x}) = \mathbf{w}_n\}$ is the data space partition that is spanned by the n-th prototype;

$\alpha_n^{(k)}, k = 1,\ldots,C$ are the shares of samples lying in P_n and belonging to each class; $\sum_{k=1}^{N_c}\alpha_n^{(k)} = 1$.

The KWM construction procedure consists of two basic steps: unsupervised VQ training and subsequent calibration.

K-Winner Classifier (training)

0. Input: training set of labelled data, X;
1. (Unsupervised prototype positioning)
   Apply an unsupervised VQ algorithm (PGAS [8,9]) to adjust the codebook, W, minimizing (1);
2. (Calibration)
   Calibrate W into a labelled codebook, W′, computed as:
   $\mathrm{W}' = \{(\mathbf{w}_n, c_n), \mathbf{w}_n \in \mathrm{W}, c_n \in \mathrm{C}, n = 1,\ldots,N_h\}$, where: $c_n = c_b, b = \max_k\{\alpha_n^{(k)}\}$.

The KWM classification process, denoted by KWM(k,$\mathbf{x}$), also involves two steps. The first performs an unsupervised categorization of the test sample, $\mathbf{x}$, with the k best-matching prototypes; the second considers the calibrations of the codewords and classifies the sample accordingly:

K-Winner Classifier (run-time operation) ≡ KWM(k,$\mathbf{x}$)

0. Input: test sample $\mathbf{x} \in \Re^D$, trained codebook, W′, agreement level, k
1. (K-winner unsupervised categorization)
   1.1 Sort the codebook, W″($\mathbf{x}$), arranging prototypes in order of increasing distance from $\mathbf{x}$:
   $$W''(\mathbf{x}) = \left\{\mathbf{w}_{n_r} \in W' : r < s \Rightarrow \left\|\mathbf{x} - \mathbf{w}_{n_r}\right\| \le \left\|\mathbf{x} - \mathbf{w}_{n_s}\right\|, r = 1, \ldots, N_h\right\};$$
   1.2 Extract the set $W_K(\mathbf{x}) \subseteq W''(\mathbf{x})$ including the K best-matching prototypes with respect to $\mathbf{x}$:
   $$W_K(\mathbf{x}) = \left\{\mathbf{w}_{n_r} \in W''(\mathbf{x}), r = 1, \ldots, K\right\}$$
2. (K-winner classification)
   If $\exists c^* : \forall \mathbf{w}_{n_r} \in W_K(\mathbf{x})\ c_{n_r} = c^*$ (if the classes of all K prototypes agree)
   then: Classify $\mathbf{x}$ as belonging to class c^*
   else: Discard $\mathbf{x}$

The classification outcome depends on the agreement of all the elements of the set of k neighbours associated with a sample. The fact that the reference positions in the data space are chosen in an unsupervised manner makes it possible to compute the classifier's Vapnik-Chervonenkis dimension. The computed value allows the confidence interval of the expected generalization error to be evaluated in advance, thus enabling the designer to choose the best classifier configuration for the categorization problem at hand. The number of tested hypotheses, K, is crucial to the KWM performance (whence the model's name) as it enters the actual expression for the d_{VC}, and ultimately affects the confidence in the classifier's decision. Such features represent the basic difference between KWMs and multiple-voter classifiers, as KWMs do not involve any majority mechanism in the classification.

III. THEORY FOR THE KWM MODEL

III.1 – The exact VC-dim and the generalization performance of a K-Winner classifier

The computation of the d_{VC} of a KWM can be performed by a direct procedure, and the resulting value is expressed by the following Theorem, whose proof is omitted for brevity.

Theorem 1 – The VC-dim of a K-Winner Classifier is $d_{VC} = \lfloor N_h / K \rfloor$. (2)

The theoretical result (2) points out some interesting features of the overall functioning of the KWM model. First of all, the VC-dim of the classifier is independent of the actual dimensionality of the data space. In addition, the availability of a limited, yet exact, value of d_{VC} allows one to derive reasonable bounds to the estimated generalization error, as the generalization analysis of the KWM can benefit from the large theory on the subject [10,11]. In the following, two basic results are reported that provide a worst-case estimate of the classifier's generalization error, π, on the overall data distribution. The estimator is the empirical training error, ν, defined as the ratio of misclassified training samples to the total number of training samples. In the computations, we follow the approach described in [3], and define the quantity:

$$\varepsilon = \frac{4}{N_p}\left[d_{VC}\left(1 + \ln\frac{2N_p}{d_{VC}}\right) - \ln\frac{\eta}{4}\right] \tag{3}$$

where η is the expected confidence in the result. An expression, $\Delta\pi_A$, for the bound to the generalization error is presented in [3] and can be computed from (3) as

$$\Delta\pi = \frac{\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4\nu}{\varepsilon}}\right) \tag{4}$$

Vapnik's bound (4) is usually adequate when dealing with a low error on the training set ($\nu << 1$). The bounded estimate of the overall generalization error will be given by:

$$\pi = \nu + \Delta\pi \tag{5}$$

meaning that, with a probability higher than $(1-\eta)$, the training error ν estimates the actual generalization error with tolerance $\Delta\pi$. The quantities involved in (5) depend on K, hence a specific generalization estimate characterizes each classifier configuration. To this end, it is useful to define the following classifier-specific quantities: $N(k)$ is the number of training samples that are processed by the k-WM classifier; $\Pi(k)$ is the expected number of misclassified

samples, over the actual sample distribution covered by the k-WM classifier. These definitions make it possible to state the following property, whose proof is also omitted for brevity.

Theorem 2 – For any k-Winner classifier, the expected number of misclassified samples, $\Pi(k)$, is monotonically non-increasing as k increases.

Theorem 2 states that stressing confidence reduces the expected number of misclassified samples in generalization performance. Therefore, the k-level analysis defines a confidence-level characterization of the classifier, whose estimated error probability, by analogy to (5), is given by:

$$\pi(k) = \frac{\Pi(k)}{N(k)} \tag{6}$$

Expression (6) applies only when $N(k) > 0$. The peculiar situation in which $N(k) = 0$ is the consequence of an incomplete sampling, and does not affect the universal validity of the generalization theory underlying (5).

III.4 – The K-Winner Machine

The scope of Structural Risk Minimization is to develop a framework that can give the designer a criterion for choosing the most suitable classifier from a set of possible alternatives. Typically, the choice depends on the appropriate VC-dim for the application considered, for which task the K-Winner Machine sorts out a family of classifiers that are easy to implement and precisely characterized in terms of expected performance.

The class-agreement principle ruling a KWM offers the possibility of discriminating between space regions on the basis of the associate confidence in classification. In practice, from the highest prototype agreement, k, attained by the KWM on a given test location, one can infer the level of confidence in the classifier's decision: higher k values indicate greater confidence values. At the same time, generalization theory makes it possible to set a bound to the overall error probability. Thus one can observe the prototype configuration associated with each point in the sample space, and compute the related estimate of the generalization error probability. This variable-confidence mechanism allows one to label data-space regions according to their expected generalization errors. This opportunity is the principle of operation of the K-Winner Machine, whose algorithm is outlined as follows:

The K-Winner Machine (KWM)

0. Input: a test sample, **x**; a trained and calibrated codebook, W′; $k = 1$; $\pi^* = 1$;
1. While : KWM(k,**x**) does not discard **x**
   - 1.a) If ($N(k) > 0$) and ($\pi(k) < \pi^*$):
     $\pi^* = \pi(k)$
   - 1.b) $k = k + 1$
2. Output:
   - 2.a) classify **x** using KWM(k-1,**x**)
   - 2.b) prompt the expected error probability, π^*, associated with the proposed classification

---

The KWM processes a sample starting from the highest error probability related to the most uncertain situation (K=1); the associate KWM spans the entire domain space and supports the conventional Winner-Take-All (WTA) categorization schema. Then the process tries to improve the confidence in the decision by checking larger and larger sets of neighbouring prototypes; if successful, this mechanism has the effect of reducing the bound to the error probability (6). It is therefore reasonable that the additional information should yield a better estimate. This phenomenon is formally motivated by Theorem 2, which confirms that generalization improves with the selectivity of the classification machine. From a geometrical point of view, detecting an increased agreement among prototypes is equivalent to locating the sample in a higher-confidence portion of the data space. On comparison with other approaches, the KWM model has the advantage of implicitly and easily facing multi-class classification problems.

IV. Experimental Results

This section describes the experimental verification of the KWM model in a synthetic and a real-world domain. Both testbeds entail a multi-class problem: the former involves a 2-D space allowing a visual interpretation of obtained results, the latter addresses a well-known standard database related to a complex and technically very significant recognition problem (OCR).

IV.1 – Artificial domain tests: 3-Gaussians problem

This artificial testbed involved a three Gaussian distributions (a three-class problem) placed symmetrically in the 2-D plane. The experiments aimed to demonstrate the space-labelling ability of KWMs. The presented results show how the space-varying confidence in classification outcomes can help interpret separation boundaries. First, 6,000 total training samples were randomly generated and used to train a VQ codebook. Figure 1 (left) presents the training set. The number of prototypes used was set to 30. Then, each point of the relevant square area underwent KWM-based classification, and was labelled by the proper level of confidence, K.

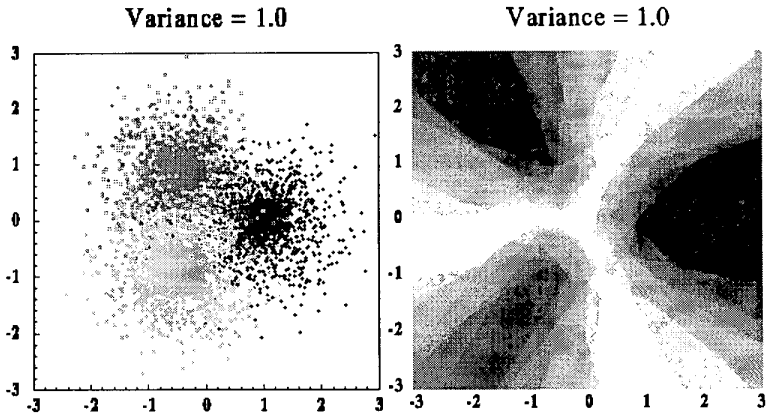

Fig. 1 – Confidence maps for the Gaussian experiment. In the right plot, darker areas indicate higher confidence Left: training set; Right: confidence areas indicate levels of K

Such an approach made it possible to plot homogeneous space regions, that is, those holding equally labelled points. The resulting graph (Fig.1 right) provides a "confidence map" and allows an easy and intuitive interpretation of the KWM functioning: the gray levels identify homogeneous regions, darker areas stand for higher confidence. Each gray level indicates the KWM classifier that rules a specific region of the data space, in other words, each level suggests the appropriate value of K to be used at each space location. As such, the map can also be regarded as a map of VC-dims, which sets variable bounds (5) (6) to the error probability.

IV.2 – Real-domain test: the NIST handwritten numerals database

The NIST handwritten digits database provides a significant testbed in an interesting real-world domain. This multi-class problem involves three distinct data sets, which, in the following, will be denoted by The training set "LS") consists of 60,000 samples each, whereas the test set ("TS") includes 58,646 samples. In the codebook training, the number of prototypes N_h=300 seemed to provide a suitable tradeoff between representation accuracy and codebook complexity, so it was always used as a default in all experiments.

Data dimensionality prevents one from drawing a confidence map for visual interpretation of KWM results. Nevertheless, the possibility of labelling space locations by confidence values clarified some interesting aspects. Training samples are used to spot "landmarks" in the confidence map. The training set was reconsidered by the empirical confidence-evaluation procedure and, for each sample, the confidence level resulting from the KWM algorithm was evaluated. Grouping samples of the same class but wth different k values made it possible to count the different confidence levels within each class, and therefore to deduce the k distribution within the class itself.

The results of these measurements are given in Figure 2: more than 85% of the samples belonging to class "0" were correctly classified with k=10. No sample of class "1" could be classifed with k=10, due to the few (twelve) prototypes calibrated with that class. The graph helps understand the mapping supported by the calibrated codebook. Classes "4" and "9" exhibit critical situations; one might ascribe them to strong overlaps with neighbouring classes. Measuring uncertainty in this way might help the designer to build a classifier that accounts for such distribution, for example, by treating critical classes selectively.

The final experimental phase addressed the verification of theoretical predictions about the expected generalization error. To this end, TS was used to measure the classification error The evaluation procedure was the following: for each test run, the theoretical bound, $\pi(k)$, to the generalization error (as computed by 6) was compared with the empirical error rate. Figure 3 presents the obtained results on TS. The comparison between empirical costs and the theoretical bound shows that, for the NIST testbed, too, Computational Learning Theory turned out to yield an expected result not too far from empirical evidence.

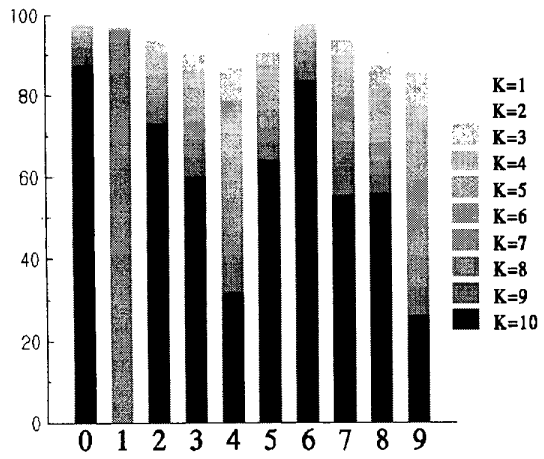

Fig. 2
Confidence distribution over classes

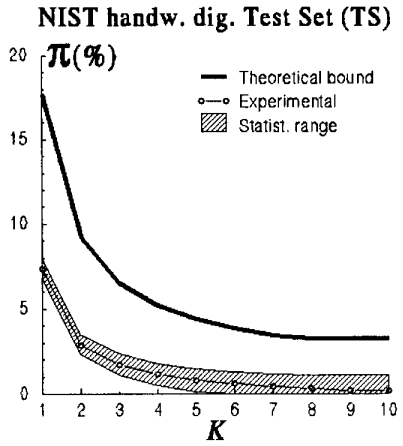

Fig.3 – NIST database: bounded and experimental generalization performance (Test Set)

V. Conclusions

The basic idea to use a prototype-based representation paradigm to span a data space for classification purposes is not entirely new, especially when using WTA mechanisms to drive space-dependent decision-making processes. In this regard, a novel aspect of the presented research lies in using several space locations to derive a measure of confidence in the classification outcome. The KWM training procedure is also quite standard (VQ codebook positioning + calibration), hence the related model can fully benefit from a vast literature.

In fact, the crucial issue associated with the KWM model lies in defining and characterizing a family of classifiers in terms of their VC-dims and consequent expected generalization error. This makes it possible to assign a confidence level to each point in the data space, and thus to select the most appropriate classifier providing the tightest bound to the generalization performance. From this viewpoint, both theoretical and practical demonstrations have shown that there is a sharp relationship between class overlap (problem complexity) and the shape of the related confidence map (expected generalization error). Such features endow KWMs with the classification accuracy of classical surface-based representation models, also providing the local-level inspection ability typical of prototype-based paradigms.

References

[1] Vapnik VN, *Estimation of Dependences Based on Empirical Data*, 1982, Springer-Verlag

[2] Prechelt L "Automatic early stopping using cross validation: quantifying the criteria" *Neural Networks*, 1998, vol. 11, No.4, pp. 761-767

[3] Vapnik VN *The nature of statistical learning theory*, Springer Verlag, New York, 1995

[4] Kohonen T, Self-organization and Associative Memory 3rd Ed., 1989, Springer Verlag

[5] DeSieno D, "Adding a conscience to competitive learning" Proc. IEEE Int.Conf. on Neur.Net., SanDiego, 1988, vol.I, pp.117-124

[6] Martinetz TM, Berkovich SG, Schulten KJ " "Neural Gas" network for vector quantization and its application to time-series prediction" *IEEE Trans. Neural Networks*, 1993, vol.4, No.4, pp.558-569

[7] Lloyd SP "Least squares quantization in PCM" *IEEE Trans. Inf. Theory*, 1982, vol.28, No.2, pp.127-135.

[8] Ridella S, Rovetta S, Zunino R "Plastic algorithm for adaptive vector quantization" *Neural Computing and Applications*, 1998, vol.7., No.1, pp.37-51

[9] Rovetta S, Zunino R "Efficient training of Neural Gas vector quantizers with analog circuit implementation" *IEEE Trans. Circuits and Sys.II*, 1999, in press

[10] Anthony M, Biggs N *Computational Learning Theory*, 1992, Cambridge Univ. Press

[11] Shawe-Taylor J, Anthony M "Sample sizes for multiple-output threshold networks" *Network Computation in Neural Systems*, Feb 1991, vol. 2, No.1, pp.107-117