Application of Neural Networks and Statistical Pattern Recognition Algorithms to Earthquake Risk Evaluation

G.Giacinto (*), R.Paolucci (+), and F.Roli (*)

*Dept. of Electrical and Electronic Eng., University of Cagliari, ITALY

Piazza D'Armi, 09123, Cagliari, ITALY e-mail {giacinto, roli}@diee.unica.it

+Dept. of Structural Eng., Technical University of Milan, ITALY

Piazza Leonardo da Vinci 32, 20133, Milan, ITALY e-mail paolucci@esdra.stru.polimi.it

Abstract

This paper reports the experimental results on the application of different pattern recognition algorithms to the evaluation of earthquake risk for real geological structures. The study area used for the experiments is related to a well-known geological structure representing a "triangular valley over bedrock". Performances obtained by two neural networks and two statistical classifiers are reported and compared. The advantages provided by the use of methods for combining multiple classifiers are also discussed and the related results reported.

Keywords: Earthquake risk evaluation; Statistical and neural classifiers; Combination of multiple classifiers.

1. Introduction

The ability to realistically predict "ground shaking" at a given location during an earthquake is crucial for seismic risk prevention strategies in urban systems, as well as for the safe design of major structures. However, the largest seismic events of the last decade have demonstrated that the observed ground shaking can be much more severe than expected and its spatial distribution poorly related to the "earthquake risk maps" previously prepared by seismologists or earthquake engineers (Faccioli, 1996). Therefore, a major improvement of the present ability to compile earthquake risk maps is required to mitigate the impact of earthquakes on urban areas, to plan land use and to prepare effective emergency plans.

In the fields of seismology and structural engineering, risk maps are obtained by "combining" data related to factors that mainly affect earthquake risk. The main "data sources" currently used are:

- data on regional seismicity, typically based on historical or seismotectonic observations;
- data on the "geological setting" of the study area;
- data on the "vulnerability" of the human and natural environment;
- data on the effects of the so-called "local soil conditions" (e.g., topographic and geological irregularities of the soil profile) on the spatial variation of ground motion during an earthquake (Sanchez-Sesma, 1987).

The latter data source allows earthquake engineers to predict risk degrees at locations characterized by different soil conditions.

In this paper, we focus on the development of pattern recognition techniques for the automatic evaluation of the effects of local soil conditions. It has been pointed out that such "site effects" were one of the main causes of concentrated damage during some of the largest earthquakes of the last decades (e.g., the earthquake that struck Mexico City in September 1985). The classical algorithms for the evaluation of the seismic site effects are briefly reviewed in Section 2, where the advantages and the potentialities of the use of pattern recognition techniques are also discussed. The formulation of earthquake risk evaluation as a pattern recognition problem is described in Section 3. Section 4 gives a brief description of neural networks and statistical pattern recognition algorithms used in the experiments. Methods used for "combining" the results provided by these algorithms are also briefly described. Section 5 describes the "study case" used for experiments. Performances obtained by different pattern recognition algorithms and by their "combination" are also reported and compared. Conclusions are drawn in Section 6.

2. Earthquake risk evaluation

First of all, it should be pointed out that the evaluation of site effects is not the only information commonly used by earthquake engineers to compile risk maps. As pointed out in the Introduction, local soil conditions strongly affect earthquake risk but additional

information should be used to completely evaluate earthquake risk for a study area. However, in the following, we will refer to site effects evaluation as the "earthquake risk evaluation problem".

The problem considered can be defined as follows. Given the local site conditions (e.g., topographic profile, geological layering and soil mechanical properties) and given the "input earthquake" (e.g., a plane wave of given amplitude and shape propagating towards the earth's surface), find the ground motion at different locations ("sites") of the study area.

The approach that has been generally used so far by earthquake engineers to solve the above problem is mainly based on different techniques for the numerical integration of the elasto-dynamics equations of motion, with the proper boundary and initial conditions (Aki and Richards, 1980). These numerical tools for the simulation of seismic wave propagation provide "solutions" that engineers usually summarize in a few parameters, such as the peak ground acceleration, the duration of motion, or other measures deemed adequate to represent the severity of ground shaking at different sites. Subsequently, according to the values of the above parameters, a risk map is compiled by assigning a certain degree of risk (e.g., low, medium or high risk) to each site.

There are three main limitations in using classical numerical tools for earthquake risk evaluation:

- the poor knowledge of the geological setting of the study area that prevents, in many cases, the creation of an accurate numerical model of the study area;
- the uncertainties in the values of local soil conditions;
- the huge computational burden required by numerical procedures to perform fully three-dimensional (3D) dynamic wave propagation analyses on realistic geologic configurations.

In terms of pattern recognition, it is worth noting that the above-mentioned numerical tools follow the classical "model-based" approach to engineering problem solving that demands a detailed and precise model of the physical phenomenon to be investigated (Haykin, 1996). The model of the study area allows earthquake engineers to develop a numerical "transfer function" that uses the seismic wave as input and provides the severities of ground shakings at the different locations as outputs. (From a pattern recognition point of view, the definition of

the above transfer function can be regarded as a problem of estimating an "input-output function").

On the basis of the above considerations, the pattern recognition approach seems to exhibit several features that could help to overcome the above limitations of classical numerical tools:

- pattern recognition provides a "non-parametric" approach to the solution of problems that involve the estimation of input-output functions. Pattern recognition algorithms like the k-nearest neighbor classifier or the multi-layer perceptron neural network can be used to estimate an input-output function without needing a model of the physical mechanism underlying the function;
- pattern recognition provides algorithms that are able to "learn" the desired input-output function by "examples";
- pattern recognition algorithms based on neural network models have proved they can effectively handle uncertainties in input data;
- pattern recognition algorithms exhibit reasonable computational complexities with respect to those of numerical procedures currently used for wave propagation simulation.

Therefore, the pattern recognition approach could be successfully used to overcome the lack of "models" for real study areas, to handle uncertainties in local site conditions, and to provide earthquake engineers with fast computational tools.

3. Formulation of the earthquake risk evaluation as a pattern recognition problem

As pointed out in the previous Section, the earthquake risk evaluation problem basically involves the "assignment" of "risk degrees" to different locations of a given study area. Therefore, it can be naturally formulated as a pattern recognition problem. The formulation requires the pattern recognition concepts of "patterns", "features", and "data classes" to be expressed in terms of the "elements" and "data" involved in earthquake risk evaluation. To this regard, let us use an example of a specific risk evaluation problem. Figure 1 illustrates a study area characterized by a geological structure representing a "triangular valley over bedrock". In the earthquake engineering field, this is an interesting study case, as it constitutes

a reasonable approximation of many real geological structures, such as sediment-filled alluvial valleys. The main elements and related data involved in the risk evaluation for a triangular valley are the following:

- the "shape" of the valley that can be characterized by geometrical features;
- the "sediment basin" (i.e., the soil underlying the valley) and the "bedrock" that can be characterized by their mechanical properties;
- the seismic wave that can be described by features commonly used for signal characterization (e.g., peak amplitude and fundamental frequency of the wave);
- the so-called "receivers" that are related to the locations of the study area for which risk degrees are to be evaluated.

From the above definitions, it is easy to see that "receivers" can be regarded as "patterns" for any earthquake risk evaluation problem. In order to characterize such patterns, "features" related to the position of the receivers, the "shape" of the geological structure, the mechanical properties of the soil underlying the receivers, and measures characterizing the "input" seismic wave can be used. With regard to the definition of "data classes", data classes can be easily associated to the considered risk degrees (e.g., three data classes related to "low", "medium" and "high" risk). If we assume to use "supervised" pattern recognition algorithms, "training sets" must also be created. Unfortunately, as pointed out in Section 2, poor and rough data are usually available for real geological structures. Typically, a few "accelerograph stations" record seismic motions for the locations of a large area and earthquake engineers are unable to "infer" ground shakings for the remaining locations. Consequently, the most practical way to build up training sets is to use numerical procedures for the simulation of wave propagation. For complex geological structures, "approximate" simulations could be carried out (e.g., "local" 2D simulations could be used for the simulation of complex 3D structures). Typically, due to the computational load and to the above-discussed limitations of present numerical codes, small and "noisy" training sets should be expected.

4. Pattern recognition algorithms and combination methods

Different neural networks and statistical classification algorithms were applied to the evaluation of earthquake risk. Among neural network classifiers, the multilayer perceptron (MLP) and the probabilistic neural network (PNN) were used. (A brief description of such neural networks is given in Sections 4.1 and 4.2). The well-known k-nearest neighbor (k-NN) and Gaussian classifiers were adopted to evaluate performances of classical statistical algorithms. For a description of such statistical classifiers, the reader should refer to Fukunaga (1990). To utilize the complementary characteristics of the above classification algorithms, methods for combining the results provided by multiple classifiers were also applied (Section 4.3)

4.1 Multilayer perceptron neural network

Multilayer perceptrons are artificial neural network models whose architecture consists of multiple layers of neurons with connections only between neurons in neighboring layers. A numerical value called "weight" is attached to every connection in the network. Information is processed starting from one side of the network called the "input layer" and moving through successive "hidden layers" to the "output layer". As an example, Figure 2a shows the topology of an MLP neural network with only one hidden layer. Each neuron computes a so-called "net input" from the outputs of previous neurons and from the weights of the connections. Typically, such a net input is a weighted sum, and a numerical value, called "bias", is added to the net input (Figure 2b). In MLPs, a function called "activation function", is applied to the net input. In our experiments, we used a sigmoid function. One of the most commonly used training schemes for MLPs is the error back-propagation (EBP) learning algorithm. This is a learning algorithm by "examples" based on a "gradient descent" technique. Typically, there is an output neuron for each data class and an input pattern is classified as belonging to a given class if the related output neuron has the highest activation among all the output neurons. Therefore, for each input pattern, the EBP algorithm adjusts the values of the network connections in order to maximize the activation value of the neuron

related to the correct class and to minimize the activation values of all the other output neurons.

The reader interested in a detailed description of MLPs can refer to Hertz et al. (1991).

4.2 Probabilistic neural network

Probabilistic Neural Networks (PNNs) are a model for supervised classification based on multivariate probability estimation (Specht, 1990). They are based on an extension of the Parzen approach to univariate probability estimation (Fukunaga, 1990). Given a set of N samples $\underline{X}_i$ drawn from a statistical distribution $p(\underline{X})$, the Parzen approach provides an asymptotic, unbiased and consistent estimate $\hat{p}(\underline{X})$ of the related probability density function by using an appropriate "kernel function" $k(\cdot)$ which is applied to each sample considered, i.e.:

$$\hat{p}(X) = \frac{1}{N} \sum_{i=1}^{N} k\left(\underline{X} - \underline{X}_i\right) . \qquad (1)$$

PNNs are based on an extension of such an approach to the multivariate case (Cacoullos, 1966), utilizing the Gaussian kernel function. The typical architecture of a PNN is shown in Figure 3. The network consists of an input layer, one hidden layer and an output layer. The hidden layer has as many neurons as the number of training patterns; as a kernel function, each neuron has a Gaussian type of activation function, and is centered on the feature vector of the corresponding training pattern. The output layer has as many neurons as the number of data classes considered; the activation function of each output neuron computes the sum of the inputs to the neuron. The neurons of the hidden layer propagate their outputs only to the neuron of the output layer corresponding to the class the training pattern belongs to. Given the feature vector of an unknown pattern as input to the net, the neurons of the output layer provide the estimates of the probability that the unknown pattern belongs to each of the data classes. The classification is carried out by using the "Winner Takes All" decision rule to identify the most probable class. Training PNNs consists in the optimization of the Gaussian

kernel by trials with different values of the “smoothing parameter” (Specht, 1990) which tunes the width of the Gaussian function.

4.3 Methods for combining multiple classifiers

Some methods to combine results provided by multiple classifiers have been proposed in the literature (Suen, 1992). Let us assume a pattern recognition problem with M "data classes". Each class represents a set of specific patterns. Each pattern is characterized by a feature vector $\underline{X}$. In addition, let us assume that K different classification algorithms are available to solve the classification problem at hand. Therefore, we can consider “ensembles” formed by "k" different classifiers (k=1..K). In order to exploit the complementary characteristics of available classifiers, the combination methods described in the following can be used.

4.3.1 Combination by Voting Principle

Let us assume that each classifier contained in the given ensemble performs a "hard" classification assigning each input pattern to one of the M data classes. A simple method to combine results provided by different classifiers is to interpret each classification result as a "vote" for one of the M data classes. Consequently, the data class that receives a number of votes higher than a prefixed threshold is taken as the "final" classification. Typically, the threshold is half the number of the considered classifiers ("majority rule"). More conservative rules can be adopted (e.g., the "unison" rule).

4.3.2 Combination by Belief Functions

It is well known that some classification algorithms can provide an estimate of the posterior probability that an input pattern $\underline{X}$ belongs to the data class ω_i :

$$\hat{p}(\underline{X} \in \omega_i / \underline{X}), \quad i = 1, \ldots, M \tag{2}$$

For example, estimates of the post-probabilities are provided by multilayer perceptrons (Serpico and Roli, 1995). Post-probabilities can be computed in a straightforward manner for the k-NN classifier. This combination method utilizes the prior knowledge available on each classifier. In particular, this method utilizes knowledge about the "errors" made by each classifier on the training set patterns. Such prior knowledge is contained in the so-called

"confusion matrices". For the z_{th} classifier C_z, it is quite simple to see that the confusion matrix can provide estimates of the following probabilities:

$$\hat{p}(\underline{X} \in \omega_i / C_z(\underline{X}) = j) \quad i = 1,\ldots,M \quad j = 1,\ldots,M \quad z = 1,\ldots,K \tag{3}$$

On the basis of the above probabilities, the combination can be carried out by the following "belief" functions:

$$bel(i) = \eta \prod_{k=1}^{K} \hat{p}(\underline{X} \in \omega_i / C_k(\underline{X}) = j_k) \quad i = 1,\ldots,M \tag{4}$$

The final classification is taken by assigning the input pattern $\underline{X}$ to the data class for which *bel(i)* is maximum.

5. Experimental results

5.1 The study case

The considered study case was a triangular valley over bedrock (Figure 1). In order to apply our pattern recognition algorithms, twenty-one “receivers” were used and a numerical procedure designed for fast analyses of 2D wave propagation within triangular valleys was applied to “predict” ground shakings for different receiver locations (Paolucci et al., 1992). For simulation purposes, we assumed an input earthquake wave represented by a “plane shear” wave propagating towards the earth surface. A Ricker type of time-dependence was implemented for this wave, since “Ricker waves” are widely used in seismic wave propagation analyses (Ricker, 1953). From an earthquake engineering viewpoint, the parameters used to characterize the valley were the following:

- the “length” of the valley (L) and the two dipping angles $\pi/2N_1$ and $\pi/2N_2$;
- the mechanical properties of the soil inside the valley (ρ_v = material density, β_v = shear wave propagation velocity, Q_v = quality factor, describing the internal dissipation of the

material due to its non-elastic behavior) and of the "bedrock", that is, the rigid basement underlying the valley (ρ_r, β_r). We assumed that no dissipation occurred inside the bedrock, and that $\rho_r\beta_r >> \rho_v\beta_v$;

- the positions of the locations ("receivers") where ground motion is measured;
- the fundamental frequency (f_p) of the input seismic wave.

The main objective of the simulations carried out was to create a "data set" containing examples of the degrees of ground shaking for receivers (i.e., "patterns") characterized by different soil conditions and different "wavelengths" of the input earthquake wave. We performed many runs of our simulator using different values of the two dipping angles and different values of the wavelength. The other measures related to the mechanical properties of the valley and the bedrock were kept constant throughout all simulations. For each run, our simulator provided the ground motion as output in terms of acceleration time histories for the twenty-one receivers at the surface of the valley. The whole simulation phase produced a data set consisting of 6300 patterns. Each pattern was related to a receiver and, in terms of pattern recognition, it was characterized by the following four features: the receiver position x/L, the two parameters defining the angles N_1 and N_2, and the "normalized wavelength" λ/L of the input wave (where $\lambda = \beta_v/f_p$ is the fundamental wavelength calculated inside the valley). In order to apply our supervised pattern recognition algorithms, each pattern was assigned to one of three "risk classes" (low, medium, and high risk) on the basis of the ground shaking values predicted by our simulator. In particular, the severity of ground shaking was computed by the peak acceleration and by the "intensity of motion" (the latter calculated as an integral measure of ground motion over its whole duration).

The obtained data set was randomly subdivided into a training and a test set of different sizes.

5.2 Results

For each kind of classification algorithm, a long "design phase" involving "trials" with different classifier architectures and learning parameters was carried out. The main objective

of these experiments was to assess the best performances provided by “single” classifiers after long design phases and to compare such performances with the ones obtained by combining the results provided by multiple classifiers.

In addition, experiments with training sets of different sizes (i.e., 10%, 20%, 30%, 40%, and 50% of the data set) were carried out in order to evaluate the effect of training set size on the performances of the different classifiers.

For the k-nearest neighbor classifier, we carried out different trials with twenty-five values of the "k" parameter ranging from 1 up to 49. For the multilayer perceptron neural network, five different architectures with one or two hidden layers and various numbers of hidden neurons (4-4-3, 4-6-3, 4-8-3, 4-6-4-3, 4-8-4-3) were considered. For all architectures, one input neuron for each feature and one output neuron for each data class was used. We trained the networks using two different values of the learning rate (i.e., 0.01, and 0.04). For each architecture and for each value of the learning rate, ten trials with different random initial-weights ("multi-start" learning strategy) were carried out. Therefore, a set of 100 MLPs was obtained. The Gaussian classifier and the Probabilistic Neural Networks needed no design phases.

At the end of the above-mentioned long design phase, a set of 127 classifiers was trained and tested on the selected data set.

The performances obtained by the above classifiers on the test set are summarized in Table 1. Table 1 refers to classifiers trained by the “10% training set” (i.e., the training set containing 10% of the patterns forming the data set). For the k-NN and the MLP classifiers, the lower, the mean, and the higher classification accuracies obtained in the aforementioned design phase are shown. The “design complexity” column gives the number of “trials” carried out for each classifier (using different architectures and learning parameters). It is worth noticing that the MLP provided the best classification accuracy and it outperformed the k-NN classifier (92.95% vs. 85.84%). This result seems to indicate that the k-NN classifier suffered from the small training set size (10% of the data set). This conclusion is confirmed by results obtained using larger training sets (Figure 4). For each kind of classifier, Figure 4 shows the trend of the classification accuracy as a function of the training set size. The difference in accuracy between the k-NN and the MLP classifiers is reduced as the size is increased.

In order to prove that the combination of different classifiers generates satisfactory classification accuracies with "reduced" design phases, we combined the results provided by different ensembles formed by just three classifiers obtained without any design phase (i.e., an a priori fixed “k” value was used for the k-NN and a single random weight trial was performed for each MLP architecture considered). Table 2 shows the results provided by three different classifier ensembles. For each ensemble, the classifiers results were combined by the majority rule and by the belief function method. It is worth noting that the design phase necessary to produce these classifier ensembles involves the training and the testing of just three classifiers (i.e., design complexity=3). This fast design phase can be used to obtain satisfactory performances that are close to the ones provided by the best single classifier obtained after a design phase involving 127 classifiers (MLP with 92.95% accuracy; see Table 1).

Other similar experiments, that are not reported for the sake of brevity, confirmed the conclusion that the combination of different classification algorithms can be used to obtain satisfactory classification accuracies with reduced design phases.

6. Conclusions

The potentials of the use of pattern recognition techniques to evaluate earthquake risk for real geological structures have been evaluated in this paper. The reported results point out that pattern recognition techniques allow earthquake engineers to classify the risk degrees of different sites with satisfactory accuracy. In particular, they can be used to overcome the limitations of numerical procedures currently used for risk evaluation. On the other hand, such procedures can be effectively used for the risk evaluation of small parts of a large study area in order to create training sets required by supervised algorithms. Finally, from the pattern recognition viewpoint, the reported results point out that the combination of different classification algorithms can be used to obtain satisfactory classification accuracies with very short design phases.

Acknowledgments

This research was partially supported by the EC Environment Research Programme (TRISEE Project, 3D Site Effects and Soil Foundation Interaction in Earthquake and Vibration Risk Evaluation, Contract:ENV4-CT96, Climatology and Natural Hazards). The authors wish to thank Prof. E. Faccioli for his helpful comments and suggestions.

References

Aki K. and P. Richards (1980). Quantitative Sesimology, Theory and Methods, W.H. Freeman and Co., San Francisco.

Cacoullos, T. (1966). Estimation of multivariate density. Ann. Inst. Stat. Math. 18, pp. 179-189.

Faccioli E. (1996). On the use of engineering seismology tools in ground shaking scenarios. State of the art paper, 11th World Conf. on Earthquake Engineering, Paper n. 2007, Acapulco, Mexico.

Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition. Academic Press, Inc., New York, 2nd edition.

S.Haykin (1996). Neural Networks expand SP's Horizons. IEEE Signal Processing Magazine, March 1996, pp. 24-49

Hertz, J., A. Krogh and R.G. Palmer (1991). Introduction to the Theory of Neural Computation. Addison Wesley Pub. Co., The Advance Book Program.

Paolucci R., Suarez M. and F.J. Sanchez-Sesma (1992). Fast computation of SH seismic response for a class of alluvial valleys. Bulletin of the Seismological Society of America, Vol. 82, pp. 2075-2086.

Ricker N. (1953). The form and the laws of propagation of seismic wavelets. Geophysics, Vol. 18, pp. 10-40.

Sanchez-Sesma F.J. (1987). Site effects on strong ground motion. Soil Dynamics and Earthquake Engineering, Vol. 6, pp. 124-132.

Serpico, S.B. and F. Roli (1995). Classification of multisensor remote-sensing images by structured neural networks. IEEE Transactions on Geoscience and Remote Sensing 33, 562-578.

Specht, D.F. (1990). Probabilistic neural networks. Neural Networks 3, 109-118.

Suen et al. (1992). Methods for combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. on Systems, Man, and Cyb., Vol. 22, No. 3, May/June 1992, pp. 418-435.

FIGURE CAPTIONS

Figure 1. A geological structure representing a triangular valley over bedrock is depicted.

Figure 2. An example of MLP neural network topology (a) and neuron computation (b).

Figure 3. Typical architecture of a probabilistic neural network

Figure 4. Trend of the classification accuracy as a function of the training set size for the considered classification algorithms.

TABLE CAPTIONS

Table 1. Classification accuracies in % obtained by considered classifiers on the test set

Table 2. Classification accuracies in % obtained by three different classifier ensembles

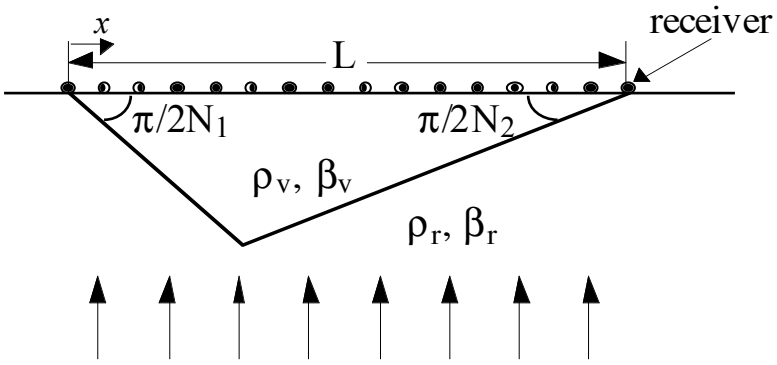

FIGURE 1

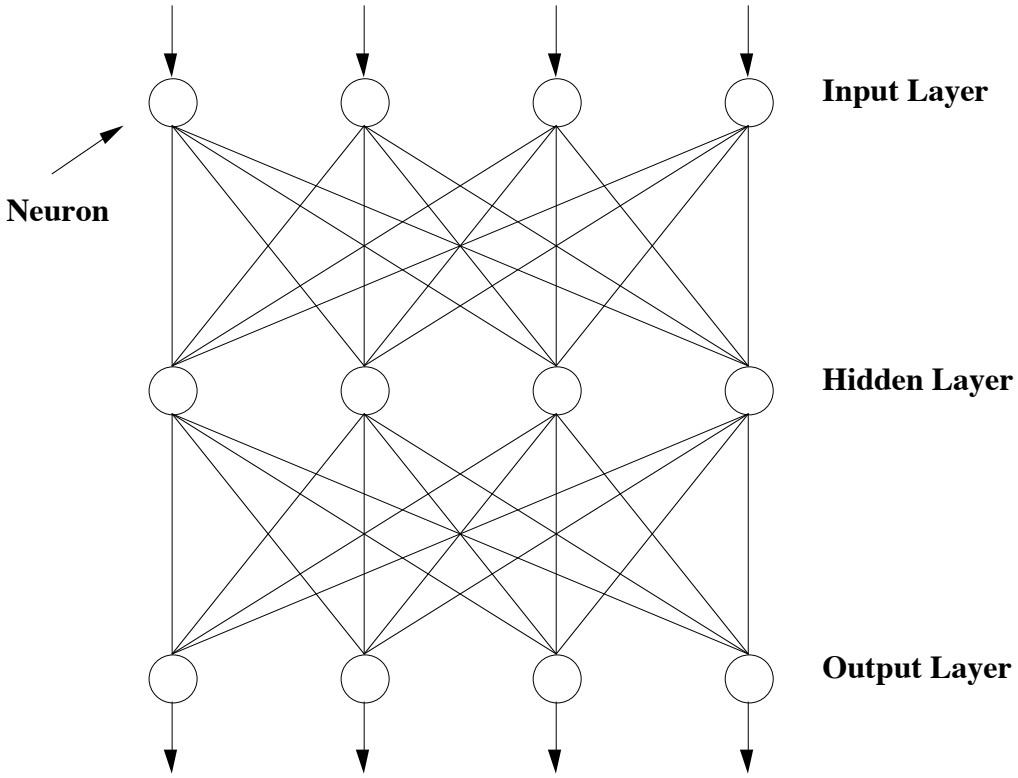

FIGURE 2a

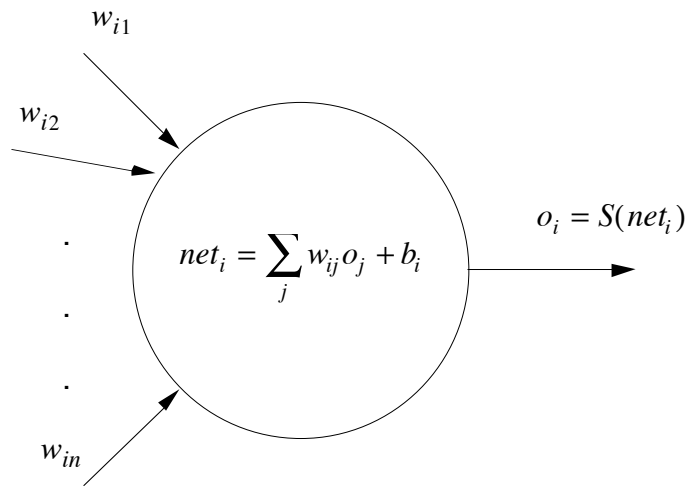

FIGURE 2b

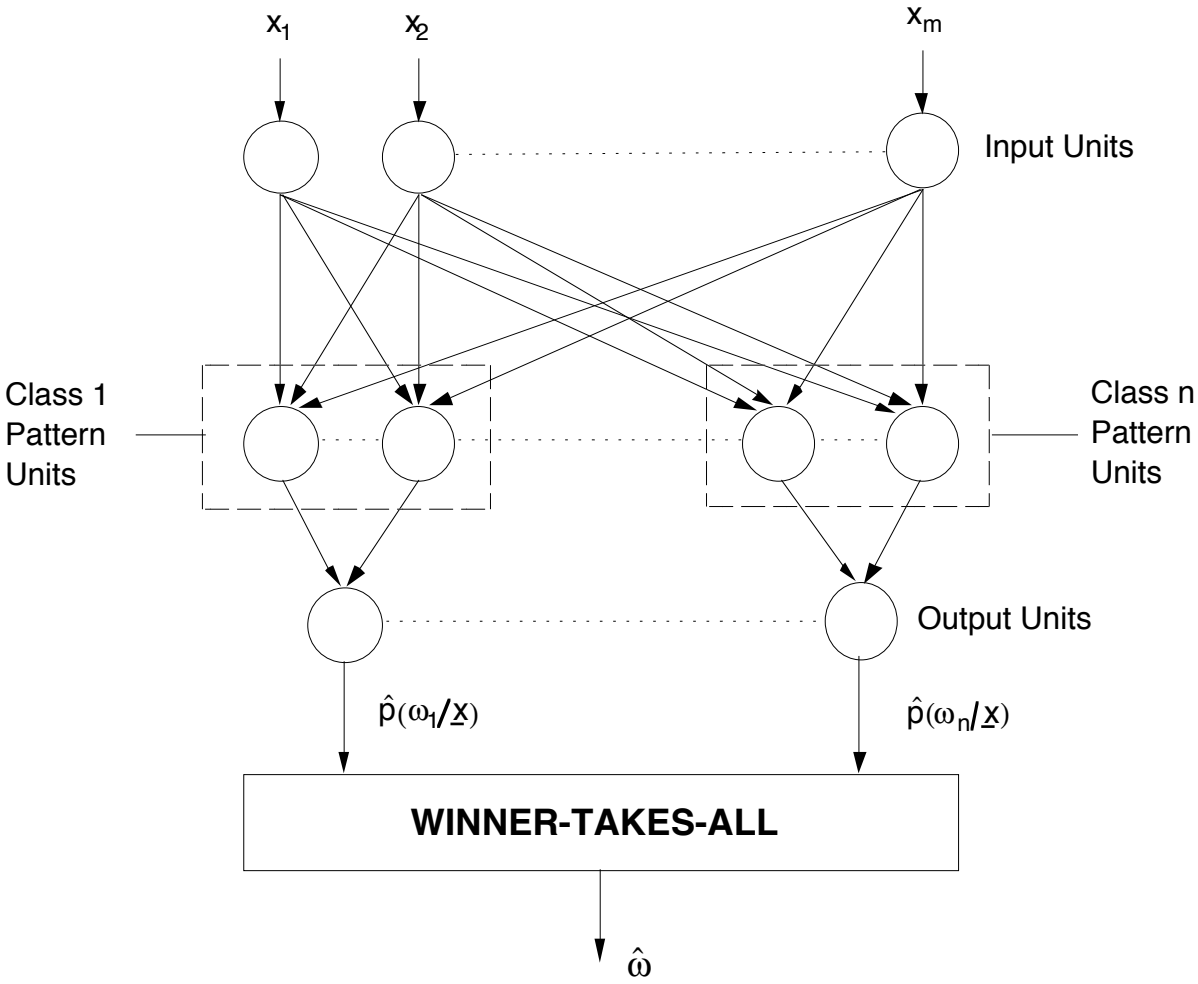
x_1
x_2
x_m
Input Units
Class 1
Pattern
Units
Class n
Pattern
Units
Output Units
$\hat{p}(\omega_1/\underline{x})$
$\hat{p}(\omega_n/\underline{x})$
WINNER-TAKES-ALL
$\hat{\omega}$

FIGURE 3

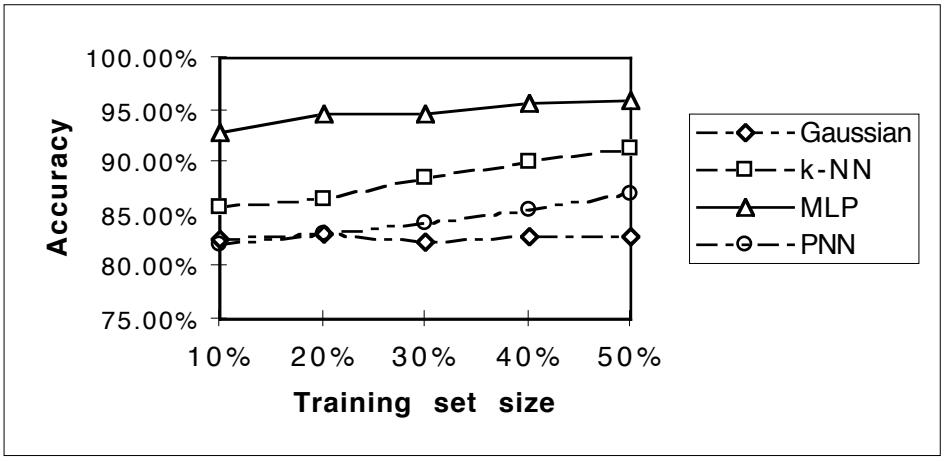
100.00%
95.00%
90.00%
85.00%
80.00%
75.00%
Accuracy
10%
20%
30%
40%
50%
Training set size
Gaussian
k-NN
MLP
PNN

FIGURE 4

TABLE I

Classifier		*Low Risk*	*Medium Risk*	*High Risk*	*Overall Accuracy*	Design Complexity
Gaussian	Lower Accuracy	94.45	39.85	64.21	82.76	1
	Mean Accuracy	94.45	39.85	64.21	82.76	
	Higher Accuracy	94.45	39.85	64.21	82.76	
k-NN	Lower Accuracy	95.58	10.28	40.91	78.93	25
	Mean Accuracy	98.19	25.57	55.72	82.05	
	Higher Accuracy	99.75	51.65	68.85	85.84	
MLP	Lower Accuracy	96.60	26.02	71.82	87.64	100
	Mean Accuracy	98.60	53.13	88.04	90.78	
	Higher Accuracy	99.61	75.13	95.43	92.95	
PNN	Lower Accuracy	88.59	59.14	72.06	82.14	1
	Mean Accuracy	88.59	59.14	72.06	82.14	
	Higher Accuracy	88.59	59.14	72.06	82.14	

TABLE II

	Low Risk	**Medium Risk**	**High Risk**	**Overall Accuracy**
MLP 4-6-4-3	98.38	63.71	79.48	90.87
PNN	88.59	59.14	72.06	82.14
k-NN (k=25)	98.55	23.22	54.39	81.79
Combination by Belief Functions	98.75	60.03	79.48	90.62
Combination by Majority Rule	97.82	47.21	71.69	90.84
MLP 4-8-3	98.94	47.97	86.16	90.04
PNN	88.59	59.14	72.06	82.14
k-NN (k=25)	98.55	23.22	54.39	81.79
Combination by Belief Functions	98.89	47.34	86.16	89.99
Combination by Majority Rule	98.21	40.36	75.65	89.98
MLP 4-4-3	98.99	46.70	84.43	89.65
PNN	88.59	59.14	72.06	82.14
k-NN (k=5)	96.32	36.17	66.13	83.66
Combination by Belief Functions	99.14	47.46	81.58	89.46
Combination by Majority Rule	96.47	47.84	76.14	88.84