Learning Mixture Models for Classification with Energy Combination

CHI-MING TSOU
Lunghwa University of Science and Technology, Taiwan

CHUAN CHEN
Fu-Jen Catholic University, Taiwan

DENG-YUAN HUANG
Fu-Jen Catholic University, Taiwan

In this article, we propose a technique called Energy Mixture Model (EMM) for classification. EMM is a type of feed-forward neural network that can be used to decide the number of nodes for constructing the hidden layer of neural networks based on the variable clustering method. Additionally, energy combination method is used to generate the recognition pattern as the basis for classification. This approach not only improves the elucidation capability of the model but also discloses the black box of the hidden layer of neural networks. Domain experts can evaluate models built by variable clusters more easily than those built by neural networks.

Key words: classification, neural network, mutual information, latent class

Introduction

In the field of machine learning, two main challenging tasks are to identify the underlying governing rules and then utilize them as the basis for constructing a model, and to increase the explanation and prediction power of the model. Constructing models of classification for the computer to search and predict is one of the crucial functions of machine learning. Different approaches have been proposed to learn a classifier from pre-classified datasets. Among them are Decision Tree (Quinlin 1993), Support Vector Machine (Burges 1998), Naive Bayesian network classifier (Duda and Hart 1973; Langley, Iba, and Thompson 1992) and Statistical Neural Networks (Pankaj and Benjamin 1992).

Recently, Latent Class (LC) or Finite Mixture (FM) models have been proposed as classification tools in the field of neural network (Jacobs et al. 1991; Bishop 1995). Models constructed by using LC or FM are similar to a feed-forward neural network with a single hidden

layer (cf. Vermunt and Magidson 2003). The main features of these approaches are to combine variables into groups and to calculate likelihood estimation values for evaluating how effective the classification is. The final goal is to find the optimum combination that has a maximum likelihood estimation value. However, in the process of building the structure of a neural network, there is a dilemma in combining variables into groups to construct the hidden layer: if we prefer a simple structure, the accuracy will be reduced; on the other hand, if we prefer the complex structure, then the over-fitting problem (e. g., the classifier learns the training data perfectly while having a high error rate in predicting new data) may occur. This is also a well-recognized problem that exists in the field of neural networks.

How to combine variables into nodes, and how many nodes to be used as the basis for classification are two key issues for hidden layer construction of neural network models. In this study, we will propose the Energy Mixture Model(EMM) as a classifier that can be used to decide the number of nodes for constructing the hidden layer of neural networks based on the variable clustering method. The suitable number of nodes for constructing the hidden layer can be obtained by evaluating the average energy of the ensemble.

For categorical variables, we will show how to cluster the variables into subsets as a node using mutual information from information theory, and then convert to its equivalent energy state that can be used to generate the recognition patterns as criteria for classification. In addition, for continuous numeric variables, we follow the idea similar to the activation function of a neural network, and try to convert the value of variables into its equivalent energy state, which then can be used to generate recognition patterns for classification as well.

Clustering of Categorical Variables

To combine categorical variables into clusters is the first step of EMM (Energy Mixture Model) construction. According to information theory, the mutual information (cross entropy) of two discrete random variables X and Y is obtained as:

$$I(X,Y) = H(X) + H(Y) - H(X,Y). \tag{1}$$

Here, $H(X,Y)$ is the total entropy of random variables (X,Y), $H(X)$ is the entropy of X, and $H(Y)$ is the entropy of Y. Two random variables X and Y will be mutually independent if $I(X,Y) = 0$. Therefore, by computing the mutual information pairwisely for a set of random variables, one can obtain a coefficient matrix. Variables with low-

est coefficient can then be grouped to form a cluster. The purpose of grouping variables into subsets with lowest association is an attempt to confirm the assumption that if the variables are mutually independent within a node, the cross effect of variables will be minimum, and then we can multiply the percentage of each variable to gain the joint percentage before converting it to energy.

Next, we will show how to construct the mixture model by combining the variable nodes. The EM algorithm (Dempster, Laird, and Rubin 1977) has been the most popular computational method for estimating parametric mixture models. The EM is an iterative parameter optimization technique and has been widely applied to latent variable models. However, a number of key issues remain unresolved, one of which is the question concerning which local maximum should be chosen as the final estimate. In other words, the choice of local maximum is not obvious, and the final selection requires careful consideration in practice. Another open issue is generalization, this concerns the commonly encountered observation that estimating mixture models by MLE (maximum likelihood estimation) leads to over-fitting, particularly when training data are limited. As will be described below, we propose to learn the basic idea behind the EM algorithm first, and then construct the mixture models with the concept that is implied by EM algorithm.

The 'Simulated Annealing Network' adopts the concept of statistical mechanics, which states that, if P_r is the probability for a system with energy E_r, then the average energy of the ensemble of such system is given by:

$$E = \sum_r P_r E_r. \tag{2}$$

And in an equilibrium system, the material follows the canonical probability distribution which is given by:

$$P_r(E_r) \propto e^{\frac{-E_r}{k_B T}}. \tag{3}$$

Here $e^{-E_r/k_B T}$ is the Boltzmann factor, k_B is the Boltzmann constant, T is Kevin temperature, and E_r is the energy of the microstate r of the system. From equation (3), taking the negative log, we can convert the probability to its equivalent energy state accordingly. The EM algorithm for mixture distribution has a particular form (cf. Sahani 1999). The log-likelihood function for the parameters is given by:

$$l_x(\theta) = \sum_i \log \sum_{m=1}^{M} \pi_m P_{\theta_m(x_1)},\ \pi_m = p(\theta = \theta_m), \tag{4}$$

which has the log-of-sum structure common to latent variable models. The joint log-likelihood for the data is then given by:

$$l_{x,y}(\theta) = \sum_i \log \pi_{y_i} P_{\theta_{y_1}}(x_1),\ \pi_{y_i} = p(\theta - \theta_{y_i}). \tag{5}$$

Equation (5) comprises three parts. The first part ($P_{\theta_{y_1}}(x_1)$) denotes the joint probability of each latent class (y_i), here variables in latent class (y_i) are assumed to be mutually independent; the second part (π_{y_i}) denotes taking the average probability, and the third part is to take the logarithm of a probability, which denotes transferring to the equivalent energy state as we learned from the concept of canonical probability distribution. Therefore, the labeled objective of the EM algorithm is likely to search the minimum energy state, which is compatible to the idea of the simulated annealing network technique.

The mechanism for deterministically annealing the optimization is such that it converges to a more global maximum, and it can also be applied to the EM algorithm (cf. Lavielle and Moulines 1997; Jebara 1999).

The primary concept of the EM algorithm is to search for the local minimum energy state as described above. One can adopt the same idea as the basis for mixture model constructing. Under the situation of equilibrium, the average energy, which is derived from the mixture model, should be a minimum energy state. In other words, the mixture model with minimum average energy is the optimum model of the ensemble in equilibrium.

Energy Mixture Model Exposition

In this study, we will adopt the energy concept and propose the Energy Mixture Model (hereinafter referred as EMM) as a classifier. EMM deems the node in hidden layer of neural network to be a cluster of variables. Each cluster will have its own energy state, and the mixture model will be represented by the recognition pattern of the labeled classes. The structure and construction of EMM will be described as follows.

EMM is one kind of feed-forward neural network. It has many perceptron structures as shown in figure 1. The input layer of manifest variables links to a cluster of the hidden layer 1; but the structure here is different from that in a Multi Layer Perceptron (MLP). In MLP, every input variable is linked to all the nodes of hidden layer. Each cluster of the hidden layer 1 is a combination of some manifest variables, and variables are near mutually independent within the same

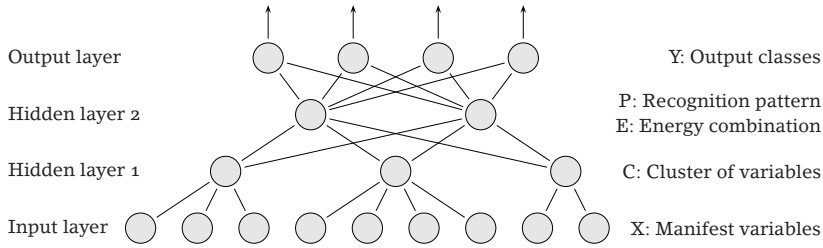

FIGURE 1 EMM structure

cluster because the cross effect of variables will be the minimum theoretically.

EMM FOR CATEGORICAL VARIABLES

According to the energy concept of statistic mechanics, we can calculate the percentage of manifest variables, compute the geometric average, and then convert to its corresponding energy state by taking the negative logarithm of the geometric average percentage value. The average energy of a cluster can be obtained by dividing the energy lump sum of the instances within the cluster to the total number of instances. In other words, if we assume n is the sample size, k is the number of clusters, each cluster contains m_j manifest variables, and p_{ij_l} is the percentage for each level l of the manifest variable for instance i in cluster j, then the energy E_{ij} of instance i in cluster j can be expressed as equation :

$$E_{ij} = -\log\left(\prod_{j_l}^{m_j} p_{ij_l}\right)^{\frac{1}{m_j}}, \tag{6}$$

the average energy of cluster j is then given by equation:

$$E_j = \frac{1}{n}\sum_{i=1}^{n} E_{ij}, \tag{7}$$

the total energy of instance i is shown in equation:

$$E_i = \sum_{j=1}^{k} E_{ij}, \tag{8}$$

and the total average energy of the ensemble is shown in equation:

$$E_a = \frac{1}{k}\sum_{i=1}^{n} E_i. \tag{9}$$

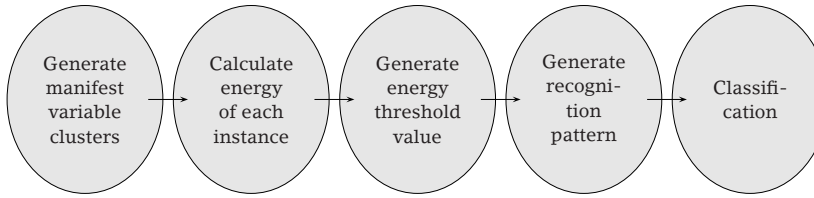

FIGURE 2 Procedure of EMM construction and classification

Next, taking the average energy(E_j) of each cluster as a threshold value, one can compare the energy of an instance in the cluster with this threshold value, and denote the result by 1 if the value is above the threshold value, otherwise denote it by 0. After that, one can obtain a recognition pattern that comprises 0 or 1. (the length of pattern will be k if the number of nodes is k). Meanwhile, taking the total energy of the instance and the recognition pattern described above, one can proceed with the classification with these two criteria. In the following, we will show how to construct EMM for categorical and continuous variables.

There are 5 steps for categorical variable EMM construction and classification as shown in figure 2. Rules for EMM construction and classification are:

1. By the use of equation (1) to calculate mutual information of two manifest variables, and take this value as the basis for constructing variable clusters.
2. For each instance, calculate the percentage of manifest variables to get energy in each cluster by equation (6).
3. Calculate the average energy of each cluster by equation (7), and calculate the total energy of each instance by equation (8), and then calculate the average energy and standard deviation for each labeled class from all the instances of the same class.
4. Take the average energy of each cluster as a threshold value, and compare the energy of each instance in each cluster against the threshold value to obtain the recognition pattern (a string of 0 and 1) for each instance.
5. Use the recognition pattern and the total energy of each instance as two criteria for classification. If there is more than one class for a specific pattern, then choose the class with total average energy plus or minus n standard deviation (n can be adjusted for optimization) that is close to the total energy of the instance as the candidate. We can also do a fuzzy classification by the use

of the class's average energy and standard deviation, or adjust the threshold value of average energy and standard deviation for optimization.

EMM FOR CONTINUOUS VARIABLES

The underlying concept of converting a continuous random variable value to its equivalent energy state is to mimic the idea of the activation function in the neural network. In other words, selection of energy conversion function for the continuous variable in the EMM model is similar to the selection of an activation function in the neural network. The purpose of converting a random continuous value to a binary state, with '0' denoting low energy state and '1' denoting high energy state, is to get a recognition pattern.

Moreover, a good way to convert a continuous numeric random variable to its equivalent energy without introducing scaling problem is to define X/μ as the energy conversion function, here μ is the mean of each random variable; if the mean value of the random variable is unknown, then one can take the sample mean $\overline{X}$ instead. If we find the result is poor, then one can try another type of conversion function to improve accuracy rate. Obviously, this kind of approach is very similar to that for a neural network. After converting the continuous variable to its corresponding energy value, then we can follow all the steps described above for EMM construction and classification.

Examples of EMM

In the following, we will use Soybean as a sample dataset to show how to make classification by the use of EMM. There are 376 instances in Soybean datasets, with 35 manifest variables, all are categorical variables, and 19 groups as the labeled classes (cf. http://www.ics.uci.edu/~mlearn/MLsummary.html). Since there are many missing values in the dataset, in order to simplify the procedure of data analysis we first convert the datasets into binary format by grouping the level of each manifest variable with minimum entropy.

VARIABLES CLUSTER AND EMM RECOGNITION PATTERN

First, use equation (1) $I(X,Y) = H(X) + H(Y) - H(X,Y)$ to compute the mutual information of two manifest variables and work out a coefficient matrix. Combine the two variables with the lowest value in the coefficient matrix into a cluster, and adjust the value of the 'reduced' coefficient matrix accordingly based on the highest value rule. Repeat this step to obtain the variables clusters. In this case, we combine the two variables with the highest degree of independence.

TABLE 1 Results of cluster average energy of Soybean sample

(1)	1	2	3	4	5	6	7	8	9	10	11
(2)	0.6046	0.6229	0.4305	0.6429	0.5134	0.7176	0.5634	0.6313	0.6413	0.6341	0.5371

NOTES (1) cluster; (2) average energy.

TABLE 2 Part of results of EMM recognition pattern of Soybean sample

No.	Pattern	Labelled class
1	01000000000	Alternarialeaf-spot, brown-spot, frog-eye-leaf-spot
2	01000000100	Bacterial-blight, brown-spot, frog-eye-leaf-spot, phyllosticta-leaf-spot
3	01000000110	Bacterial-pustule
4	01000001100	Powdery-mildew
59	11111111111	2-4-d-injury, cyst-nematode, herbicide-injury

There are still many options for setting the rules. One can get various cluster results according to the rules one sets. This is similar to the work of feature selection with neural networks.

Having obtained the clusters of variables, one can proceed with computing the energy of each instance in each cluster, follow the procedures narrated in figure 2 and calculate the total average energy for the ensemble with equation (10):

$$-\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k}\log\left(\prod_{j_l}^{m_j} p_{i_{j_l}}\right)^{\frac{1}{m_j}}, \tag{10}$$

here n is the number of instances, k is the number of clusters, and m_j is the number of manifest variables in each cluster. The objective of EMM is to find out a combination of clusters that can generate the lowest total average energy.

In this case, we try several clustering options, and one of the cluster average energy results with 11 clusters is shown in table 1. Part of the recognition pattern(s) generated for each labeled class (each digit in the pattern corresponds to the energy comparison result against the threshold value of the cluster, 0 denotes lower/equal and 1 denotes higher) is shown in table 2. The average energy and the standard deviation for 19 labeled classes are shown in table 3.

In table 4, we list some EMM classification results. The column 'before adjustment' means use the calculated threshold value, and column 'after adjustment' means fine tune the threshold value of each cluster. Case 3 has the lowest total average energy 222.1. Case 2 has the highest accuracy rate before and after adjustment, and Case 1 is the case that has the lowest total average energy before Case 2. Case

TABLE 3 Results of EMM average energy and standard deviation of labelled classes

Class	Name	Average energy	Standard deviation
1	2-4-d-injury	11.95	0.0059
2	Alternarialeaf-spot	5.23	0.3212
3	Anthracnose	6.57	0.5444
4	Bacterial-blight	5.60	0.1148
5	Bacterial-pustule	6.34	0.5862
6	Brown-spot	5.39	0.1796
7	Brown-stem-rot	6.54	0.4630
8	Charcoal-rot	7.23	0.1081
9	Cyst-nematode	11.42	0.0131
10	Diaporthe-pod-&-stem-blight	8.91	0.0396
11	Diaporthe-stem-cancer	5.86	0.0575
12	Downy-mildew	6.51	0.1743
13	Frog-eye-leaf-spot	5.46	0.1928
14	Herbicide-injury	11.61	0.0067
15	Phyllosticta-leaf-spot	5.86	0.2454
16	Phytophthora-rot	8.43	0.8507
17	Powdery-mildew	6.03	0.1095
18	Purple-seed-stain	6.05	0.2682
19	Rhizoctonia-root-rot	6.83	0.5405

TABLE 4 Part of the results of classification for Soybean sample

Case	Number of clusters	Total average energy	Accuracy rate before adjustment	Accuracy rate after adjustment
1	9	225.7	82.7	86.7
2	11	227.8	89.3	92.5
3	10	222.1	58.2	78.9

2 has the highest after adjustment accuracy rate 92.5%, the error rate is 7.5%, which implies that the manifest variables are not completely independent . EMM is a model with the property of probability, hence the result will be determined by the instance data with the character of probability, and the problem of over-fitting should be avoided.

Next, we take IRIS dataset (cf. http://www.ics.uci.edu/~mlearn/MLsummary.html) as a sample for studying EMM of continuous variables. IRIS contains three labeled classes of 50 instances each, where each class refers a type of iris plant. There are four continuous numeric variables, sepal length, sepal width, petal length, and petal width, all in centimeter units (cm).

We proceed with 3 cases:

1. Convert the numeric variables into binary format base on the sample mean for each variable.
2. Make a discretization for the variables by splitting the variable into 6 partitions with sample mean and standard deviation as quantiles.
3. Make a conversion to its corresponding energy state by the formula X/μ. The results are shown in table 5.

The accuracy rate for the binary case is 64%, and it is 76% for the discretization case. However, for the energy case, the accuracy rate before adjustment will increase to 94.6% and after adjustment will be 98.7% (only 2 instances are misclassified).

PERFORMANCE COMPARISON OF EMM

In order to examine the effectiveness of EMM, the experimental procedure utilized by Kohavi(1995) is adopted here, which can serve as a cross validation for EMM.

We choose Soybean-large and Vehicle as two datasets for experiment, and compare the results provided by Kohavi. Meanwhile, we also choose MLP neural network models from Neural Connection version 2.0 which is developed by SPSS to get some results for comparison. Soybean-large and Vehicle are two real-world large-scale datasets, Soybean large has 35 attributes which are all categorical variables, Vehicle has 18 attributes which are all continuous variables.

The procedure is initiated by taking 100 random samples from each dataset, followed by constructing EMM by the rest of instances, and finally completed by validating the testing samples to get the accuracy rate of the model. The experiment is repeated 50 times, the average accuracy rate and standard deviation are calculated after finishing the experiment. The results are shown in table 6.

Three calculation results that are based on EMM methodology, but with different extents (levels) of adjustments, are used in the cross validation. The first one is before adjustment, which means making the validation before adjusting the threshold value of each variables cluster or random variable, this shows the original accuracy rate of EMM; the second one is adjustment without resisting over-fitting, which means adjusting the threshold value of each variables cluster or random variable but ignoring the over-fitting problem; and the third one is adjustment with resisting over-fitting, which means adjusting the threshold value of each variables cluster or random variable, with a constraint that the adjustment will be accepted only

TABLE 5 Results of IRIS EMM fitness

EMM model	Before adjustment	After adjustment
Binary	64.0	64.0
Discretization	76.0	78.6
Continuous numeric	94.6	98.7

TABLE 6 Performance comparison results of EMM Datasets

Datasets	Soybean-large	Vehicle
Attribute	Categorical	Continuous
Number of attributes	35	18
Number of categories	19	4
Total size	683	846
Sample size	100	100
C4.5	0.705±0.0022*	0.601±0.0016*
Naïve Bayesian	0.798±0.0014*	0.468±0.0016*
MLP neural network	0.662±0.08	0.505±0.06
EMM before adjustment	0.704±0.06	0.495±0.05
EMM adjustment without resisting over-fitting	0.769±0.06	0.545±0.05
EMM adjustment with resisting over-fitting	0.801±0.06	0.631±0.05

NOTE *Cf. Kohavi 1995.

for both model and prediction accuracy rate improvement to avoid the over-fitting problem.

From the results in table 6, the before-adjustment accuracy rate of EMM for categorical attribute dataset Soybean-large is slightly higher than MLP Neural Network, is nearly the same as C4.5, but is slightly lower than Naïve Bayesian. For the case of adjustment without resisting over-fitting, the accuracy rate of EMM is slightly higher than C4.5 but still lower than Naïve Bayesian. Rather, for the case of adjustment with resisting over-fitting, the accuracy rate of EMM is slightly higher than Naïve Bayesian. On the other hand, the cases for the continuous attributes dataset Vehicle exhibit a different trend. The accuracy rate of EMM for the case of before adjustment is slightly higher than Naïve Bayesian, but lower than MLP Neural Network, and C4.5 for the case of adjustment without resisting over-fitting is slightly higher than Naïve Bayesian and MLP Neural Network, but lower than C4.5. Rather, for the case of adjustment with resisting over-fitting, the accuracy rate for EMM is slightly higher than C4.5, Naïve Bayesian and MLP Neural Network. These results indicate that the performance of EMM can be improved by avoiding the over-fitting problem while adjusting the threshold value for the model.

Conclusions

EMM is a type of feed-forward neural network. Clusters in this model are similar to the hidden layers in a neural network. The EMM approach can be used to decide the number of nodes for constructing the hidden layers of neural network which are based on the variable clustering method. Hence, EMM not only improves the elucidation capability of the model but also discloses the black box of the hidden layers of neural network. Domain experts can evaluate EMM models more easily than other means and this is the major contribution of EMM to knowledge discovery.

References

Bishop, C. M. 1995. *Neural networks for pattern recognition.* Oxford: Oxford University Press.

Burges, C. J. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2 (2): 121–167.

Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of Royal Statistical Society* B 39:1–38.

Duda, R., and P. Hart. 1973. *Pattern classification and scene analysis.* New York: Wiley.

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation* 3 (1): 79–87.

Jebara, T. 1999. Beyond local optimization: Annealing the bounds. Http://vismod.media.mit.edu/tech-reports/TR-507/node45.html.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* ed S. Mellish, 1137–1143. San Mateo, CA: Kaufmann.

Langley, P., W. Iba, and K. Thompson 1992. An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence,* 223–228. San Jose, CA: AAAI Press.

Lavielle, M., and E. Moulines. 1997. A simulated annealing version of the En algorithm for non-Gaussian deconvolution. *Statistics and Computing* 7 (4): 229–236.

Pankaj, M., and W. W. Benjamin. 1992. *Artificial neural networks: Concepts and theory.* Los Alamitos, CA: IEEE Computer Society Press.

Quinlin, J. R. 1993. C4.5: *Programs for machine learning.* San Mateo, CA: Kaufmann.

Sahani, M. 1999. Latent variables models for neural data analysis. PhD thesis, California Institute of Technology.

Vermunt, J. K., and J. Magidson. 2003. Latent class models for classification. *Computational Statistic & Data Analysis* 41 (3–4): 531–537.