On the technique of linguistic classification based on fuzzy neural network

Rybkin V.A.
Information Technologies Dept.
Tver State University
170000, Russia, Tver,
ul. Zhelyabova, 33
vladimir.rybkin@tversu.ru

Soldatenko I.S.
Computer Science Dept.
Tver State University
170000, Russia, Tver,
ul. Zhelyabova, 33
soldis@tversu.ru

Bandurina T.V.
Computer Science Dept.
Tver State University
170000, Russia, Tver,
ul. Zhelyabova, 33
band.t@mail.ru

Abstract

Rapid growth of the Internet informational resources has laid to the situation when classical search algorithms based on a template matching are no longer effective due to their practically boundless result sets. One of the possible solutions is using style classification which allows to effectively narrow search range. This work gives some guidelines for constructing linguistic classifier basing on the fuzzy sets theory and artificial neural networks. It also contains some recommendations for processes of forming and optimizing neural network's attribute space. Few examples of corresponding neural network architectures are given[1].

Keywords: Linguistic Classification, Information Search, Fuzzy Neural Networks, Fuzzy Sets.

Preface

The Internet nowadays is a large-scale geographically distributed database of heterogeneous resources. Low effectiveness of search engines can lessen the value of all its benefits and profits. The outcome of search engines which are based on classical model of template matching usually is represented by a vast list of resources that demands second search query. The significant number of works are devoted to organization of efficient search, see for example [1, 4]. One of the approaches to search process organization intended to partly solve the problem is utilization of style classification. Linguistic classifier allows to constrict search range and increase its precision.

This paper describes an approach to constructing linguistic classifier on basis of fuzzy neural networks with possibilistic interpretation of investigated parameters.

1 Text classification problem

We define linguistic classification as a process of distribution of texts in some natural language on groups each of which possesses a set of identical or similar linguistic characteristics. Examples are style classification, classification of texts by genres of the same style, etc.

Specificity of the lingustic classification task lies primarily in semantic ambiguity of natural language texts. Classification algorithms are based on the assumption that text semantics is somehow embodied in its syntax structure. For example, fraction of exclamation marks in fiction texts is on average higher than in scientific and technical ones, and so on. Thus existence of such text characterisics that allow to distinguish (with some precision) texts of different styles is assumed.

Let $T = \{t_1, t_2, ..., t_i, ...\}$ is a set of all texts in natural language. Only one natural language is assumed to be in use.

Let $C = \{c_1, c_2, ..., c_n\}$ are classes, that form separations of T.

We define the term *classification* as a table where each column is a tabular representation of characteristic function of corresponding set c_j: $\mu_j(\mathrm{t}_i) = \alpha_{ij}$, all these functions are discrete.

We define *crisp classification* as an aggregate of

[1]This work is supported by Yandex company (provides most popular russian Internet search engine)

characteristic functions of crisp sets c_j, where each function's codomain is $\{0, 1\}$ and

$$\alpha_{ij} = \begin{cases} 1, \text{ text } t_i \text{ belongs to class } c_j \\ 0, \text{ in another case.} \end{cases}$$

We define *fuzzy classification* as an aggregate of characteristic functions of fuzzy sets c_j, where each function's codomain is a segment [0, 1] and α_{ij} is a grade of membership of text t_i to class c_j.

We use notions of recall, precision and F-measure [7] to estimate crisp classification quality. Calculation of this characteristics is based on the following values: number of texts for a fixed class c_j that were classified to c_j and belong to this class (N_a), classified to c_j but are not in this class (N_b), texts from c_j which were not classified as texts of this class (N_c). Recall of classification is calculated by $Rec = N_a/(N_a + N_c)$, precision is given by $Prec = N_a/(N_a + N_b)$. Also we use a summary estimation of classification quality that depends on recall and precision:

$$F = \frac{1 + \beta^2}{\frac{\beta^2}{Prec} + \frac{1}{Rec}},$$

where β defines relation of importance of recall and precision and in practice takes values from [0,1]. Considering fuzzy case we propose to use appropriate values $N_a^\alpha, N_b^\alpha, N_c^\alpha$ accounting texts' classification to c_j with possibilities not less then predefined threshold α.

Obviously, it is impossible to specify functions $\mu_j(\mathrm{t})$ in an explicit form. We define functions $\hat{\mu}_j(\hat{t})$ that approximate real ones. Initial information for making approximation functions is a sample classification, which is finite-dimensional table built on a representative (finite) sample of texts with equal presence of all classes. Here $\hat{t} = (p_1(t),$ ..., $p_s(t))$ is an attribute space vector which represents text t to linguistic classifier.

2 Classification parameters identification

One of the main tasks while constructing linguistic classifier is identification of classification parameters (attribute space).

We define *classification parameter* as a real-valued function with a domain T. We propose the following obvious classifications of parameters:

1. On basis of parameter construction: *constructive* (can be evaluated with the help of deterministic algorithm) and *expert* (there is no finite deterministic algorithm that can evaluate them). Constructive parameters can be further divided into *algorithmic* (can be evaluated without any dictionaries) and *dictionary* (use various dictionaries).
2. On basis of parameter's linguistic properties: *formal* (interpret text as a sequence of symbols) and *linguistic* (think of text as a complex aggregation of linguistic objects).

An attribute vector is a fixed set of real numbers that are results of given parameters computation on an input text.

Considering fuzzy case we can define several fuzzy subsets of T and use their membership functions as parameters. In this case attribute vector consists of membership grades of a given text to defined fuzzy sets. Classical way of fuzzy sets formation uses a notion of linguistic variable. Assume we have a set of crisp parameters. We transform each crisp parameter in the set to an appropriate linguistic variable [9], whose values will serve as required fuzzy sets. This is an expert way because one needs to define linguistic variables values shapes explicitly on the basis of expert linguistic knowledge.

Another method uses a predefined set of crisp parameters, too, however it also takes into consideration an a priori information about membership of texts to classes, taken from representative sample. To do this we divide all texts from representative sample into corresponding classes. Then we take first parameter p_1 and evaluate its values for all texts from the first class c_1. Obtained numbers are further used to draw bar graph. For this we mark all evaluated values of parameter p_1 on the axis of abscissas. Next we define quantization step and use it to draw half-intervals on the same axis (for distinctness we assume that right boundary belongs to half-interval). We count how many values $p_1(t_i)$ got to each half-interval.

Let k_i – number of points in i-th half-interval. For each k_i we draw a bar on the corresponding half-interval with ordinate value equals to $k_i/(\max k_j), j = 0, 1, \ldots$. Apparently approximated function will be bounded by segment [0, 1], its value being closer to upper bound when more points get to a half-interval.

It follows from the process of construction that the obtained discrete function is a membership function of some fuzzy set, defined on a non-overlapping separation of T (domain of this membership function consists of group of texts and not individual texts). The same procedure is done with all parameters in the set and all classes.

Apparently, the more parameters participate in the classification process the more precise it may be, but as a result process of preparation of texts for linguistic analysis consumes more time. Therefore next important task is attribute space optimization. We assume that there is an initial list of parameters. For optimization process we specify a function of relevance r, defined on $P \times C$. Further, we define a general estimation of parameter relevance, for example by one of the following ways: $\bar{r}(p_i) = \max_j r(p_i, c_j)$, $\bar{r}(p_i) = \sum_j r(p_i, c_j)$ or $\bar{r}(p_i) = \sum_j w(c_j)\, r(p_i, c_j)$, where $w(c_j)$ is a weigth of class c_j. Now we can sort all parameters by degree of their relevance and form a selection of the most useful ones. Different relevance functions can be taken [3], for example, χ^2 test, Information Gain (IG) or Gain Ratio (GR).

3 Attribute space construction methods

There are two ways of constructing attribute space: manual (expert) and automatic. Below we consider the second case.

First of all we give a formal model of attribute space. We define informational object as an object that contains an arbitrary amount of information – objects semantics. Different objects have different informational richness, or *semantic scope*. We define by induction a notion of informational levels in order to take into consideration this fact.

Basis of induction – zero informational level (Δ_0) – is an alphabet of natural language extended by punctuation marks, numbers, brackets and other auxiliary symbols with the exception of whitespaces which are not linguistic units but technical symbols. Induction condition is in the following. Let i-th informational level is built - Δ_i. To form Δ_{i+1} we make ordered σ-algebra upon Δ_i. The term "ordered" σ-algebra means that we use ordered union operation: $A_1 \cup A_2$ is not the same as $A_2 \cup A_1$.

Thus the first informational level is a set of words in a given alphabet. Obtained ordered σ-algebra contains a lot of meaningless and even linguistically incorrect objects, because source alphabet is composed also from different auxiliary symbols. We can define a special subset $\Delta_1^* \subseteq \Delta_1$ containing only linguistically correct words. The second informational level is formed by ordered combinations of words and punctuation marks. The third informational level consists of texts. The fourth informational level is composed of groups of texts which we define as styles.

Attribute space is actually a mapping of the third informational level into multidimensional set of real numbers. We can try to construct this mapping by using *semantic measures* on the informational levels. We demand from semantic measure to be a total and non-negative function. For example we can introduce the following trivial measure: each symbol of the source alphabet (Δ_0) gets unique number – these numbers serve as measures of appropriate symbols. Measure of i+1-th informational level object is evaluated as sum of i-th level objects measures that constitute the concerned object. This trivial measure is sensitive only to information stored in the zero informational level. If we take some word $w = l_1 l_2 \ldots l_n$ and mix its words randomly then originating word $\hat{w} = l_{i_1} l_{i_2} \ldots l_{i_k}$ will have just the same measure's value. We propose the following three definitions in order to describe this property.

If $\forall x_1 \in \Delta_i, \forall x_2 \in \Delta_i : p(x_1) \neq p(x_2) \Rightarrow x_1 \neq x_2$, then measure p is called *weak measure* of i-th level.

If $\forall x_1 \in \Delta_i, \forall x_2 \in \Delta_i : x_1 \neq x_2 \Leftrightarrow p(x_1) \neq p(x_2)$, then measure p is called *strong measure* of i-th level.

If for each level i measure p is a weak measure

and there is no such level j for each it is strong measure then measure p is called *everywhere weak measure*.

Now we can give an example of strong measure of first informational level that gives us attribute space using only representative sample of texts and no expert knowledge. In order to achieve this we use one of the widespread linguistic methods of semantic analysis – distributive method [10], which analyses distribution of words in texts. Its main idea lies in the assumption that different words met with each other in texts not randomly, but according to some rules.

Algorithm of distributive measure evaluation process is the following. We take some word and make its distribution. It will serve as an attribute vector for the word. In practice we can represent each word that goes in a context with analyzed one as a separate attribute. Fitness of this measure for higher informational levels actually follows from the definition of informational levels as σ-algebras. But we need to explain the meaning of union operation.

Let $t \in \Delta_3$ be some text: $t = s_1 s_2 \dots s_n$. Consider its arbitrary sentence $s \in \Delta_2$: $s = l_1 l_2 \dots l_m$. Let p be a distributive semantic measure as it was defined above. Its components are the following: $p(l_i) = \left(\left[\bar{l}_1|k_1\right], \left[\bar{l}_2|k_2\right], \dots, \left[\bar{l}_{n_i}|k_{n_i}\right]\right)$, where $\bar{l}_j$ is an attribute word that represents j-th dimension, k_j - strength of j-th semantic attribute. While analyzing sentences semantics we can represent it as an aggregate of all vector components of all vectors for all words that make up the sentence - $p'(s_i)$. Suppose we have q components with similar attribute words. Obviously we can replace all of them by one component: $[a|f(q, k_1, k_2, \dots, k_q)]$, where $f(q, k_1, k_2, \dots, k_q)$ contains information about overall quantity of components and about each component taken separately. In practice we can take sum, weighed mean, minimum, maximum and so on. We do the same thing for all groups of similar components and get as a result vector: $p(s_i) = \left(\left[\bar{l}_1|k_1\right], \left[\bar{l}_2|k_2\right], \dots, \left[\bar{l}'_n|k'_n\right]\right)$ which characterizes sentences semantics by its lexical structure.

We can now evaluate distributive measure for each text t, taking instead of distributive representations of words distributive representations of sentences. This measure characterizes general semantic background of lexical structure of a text.

4 Neural network architecture

One of the most promising tools for object classification is artificial neural network, because of its ability to learn and huge computational capabilities. Attribute space and set of classes designed for classification process serve as input and output of such network.

Example of a simple network architecture is shown on Fig. 1.

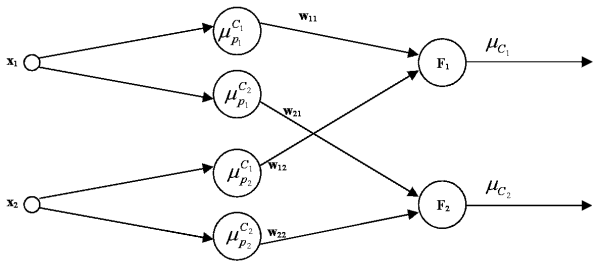

Figure 1: Example of network architecture for fuzzy attribute vector

Distribution functions' approximation for corresponding fuzzy sets is carried out on the first layer. On the second layer calculation of weighed normalized sum of inputs is performed, $F_1 = \frac{w_{11} \cdot \mu_{p_1}^{c_1}(x_1) + w_{12} \cdot \mu_{p_2}^{c_1}(x_2)}{w_{11} + w_{12}}$, $F_2 = \frac{w_{21} \cdot \mu_{p_1}^{c_2}(x_1) + w_{22} \cdot \mu_{p_2}^{c_2}(x_2)}{w_{21} + w_{22}}$, $x_1 = p_1(t)$, $x_2 = p_2(t)$. This network is learned by means of back propagation method [5].

Besides of the considered above method of forming of input parameter's possibilistic distribution by descrete function, it is possible to build a deterministic neural network for distributions realization. In this case each distribution is represented by the result of binary operation over two continuous parametric functions and its parameters are modified during a network learning process. The example of classification system for modeling possibilistic distributions with deterministic subsystem is presented on Fig. 2.

Layer 1. This layer contains subsystems N $\mu_{p_j}^{C_i}$ that are deterministic 2-layered neural networks

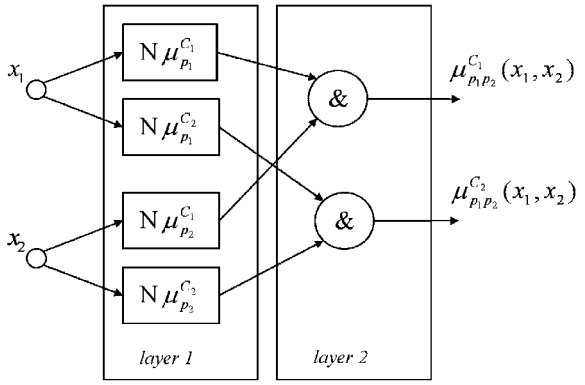

Figure 2: Modeling of possibilistic distributions of classification parameters using a deterministic subsystem

illustrated on Fig. 3. Here $F(x)$ is a sigmoidal activation function, $F(x) = \frac{1}{1+\exp(-x)}$, a_i and b_i are adjustable weights.

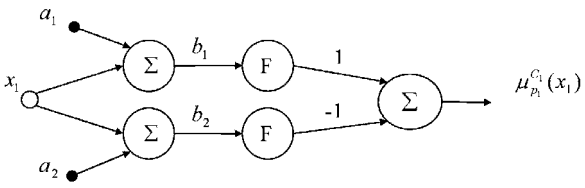

Figure 3: Subsystem N $\mu_{p_j}^{C_i}$

Layer 2. Each neuron of this layer is the fuzzy neuron "AND" [6].

Each class fuzzy set approximated by the network is characterized by membership function which defines implication $A \rightarrow B$. This impication has the following form: if x_1 is C_1^j and x_2 is C_2^j ... and x_k is C_k^j then $(x_1, x_2, \ldots, x_k)$ is C^j, where C_i^j is a fuzzy set of parameter x_i evaluated for all texts for class c_j and C^j is a resulting fuzzy set of texts of class c_j. Implication membership function is defined as a t-norm on C_i^j membership functions that are realized by deterministic neural networks with sigmoidal activation functions.

Methods described above are applicable only in the case when parameter's joint distributions can be represented by an appropriate operation over distributions of separated parameters, i.e. when all input parameters are independent. Since in our case an a priori knowledge is represented only as a sample of texts we don't have any information about parameter's correlation. Therefore in general some errors most probably will occur during the indentification of related parameters that leads to construction of wrong distribution functions and finally to inadequate behavior of the network.

One of the feasible solutions is based on the assumption of presence of input parameters correlation. It uses a network with neurons that realize the functions of joint distribution of parameters (Fig. 4).

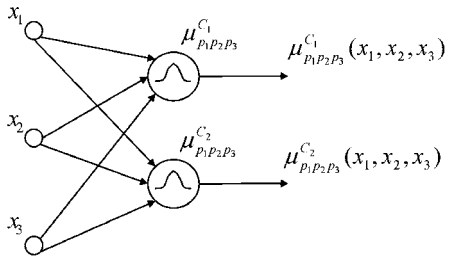

Figure 4: Modeling the joint distribution function of the correlated possibilistic classification parameters

Functions $\mu_{p_1 p_2 \ldots p_n}^{C_i}(x_1, x_2, ..., x_n)$ can be constructed by the method analogous to the considered above method of construction of discrete distribution function for one parameter.

Discovering of parameters with independent distribution functions can increase the speed of computation process and network learning.

On Fig. 5 the sample of self-organizing architecture is presented. Network setup is based on automatic determination of the dependences among input parameters.

Layer 1. At this layer the possibilistic distribution functions of input parameters and joint distribution functions for all combinations of input parameters are realized.

Layer 2. Each neuron of this layer is a fuzzy neuron 'AND' which computes grade of membership to the class c_j of text with parameters p_i taking values equal x_i in different cases of parameters dependencies.

For example, neuron n_1 computes $\mu_{p_1 p_2 p_3}^{C_1}(x_1, x_2, x_3)$ under condition when parameters p_1 and p_2 have joint distribution functions

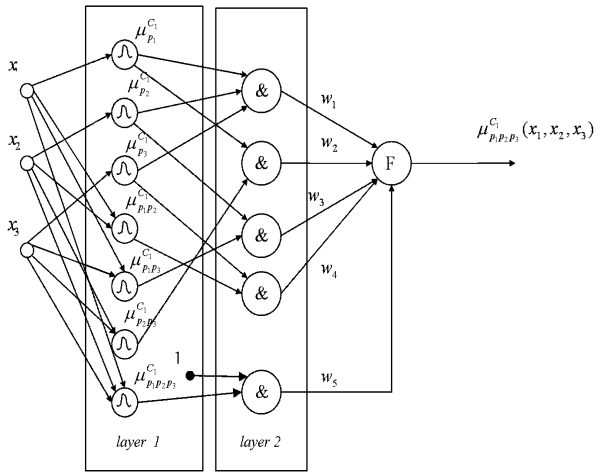

Figure 5: The sample of self-organizing architecture

and can not be represented by independent possibilistic distribution functions.

Neural network learning process includes two phases. On the first phase the setup of parameters for each possibilistic distribution is carried out. Second phase consists of selection of weights' w_i optimal values and allows to identify the independent parameters — network structure can be optimized by removing neurons representing joint distribution functions of independent parameters.

Conclusion

Obviously, the proposed method is not the only one while talking about linguistic classification primarily intented for information search. Method based on discriminant analysis [2] also finds successful application. Another group of classification methods is based on probabilistic models, primarily Bayes family [7]. But in spite of the fact that these methods allow to achive in practice satisfactory results, classification made by such classifiers does not take into consideration fuzziness of information being processed. Utilization of the theory of fuzzy sets allows to formalize this property and effectively control it.

Acknowledgments

The work was supported by Yandex company (`www.yandex.ru`).

References

[1] I.Z. Batyrshin and R.N. Gilmutdinov, "About problem of information search in Internet," Researches in Computer Science, IPI AN RT, vol. 8., 2004.

[2] P.I. Braslavsky, "The using of document' stylistic parameters in Internet information search," Proceedings of VI working meeting on electronic publications EL-PUB–2001, Novosibirsk: IVT SO RAN, 2001.

[3] F. Debole and S. Sebastiani, "Supervised term weighting for automated text categorization," In Proceedings of the 18th ACM Symposium on Applied Computing, 2003, pp. 784–788.

[4] V.N. Gudivada, "Information search on World Wide Web," Computer Weekly, No. 35, 1997, pp. 19–21, 26, 27.

[5] S. Osowski, "Sieci neuronowe do przetwarzania informacji," Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2000.

[6] D. Rutkowska, M. Piliński, L. Rutkowski, "Sieci neuronowe, algorytmy genetyczne i systemy rozmyte," Wydawnictwo Naukowe PWN, Łódź, 2004.

[7] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," Information Retrieval 1999, pp. 69–90.

[8] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 412–420.

[9] L.A. Zadeh, "The concept of a linguistic variable and its applications to approximate reasoning - I," Information Sciences, vol. 8, 1975, pp. 199–249.

[10] M. Kay and M. Röscheisen, "Text-translation alignment. Computational Linguistics", 1993, 19 (1), pp. 121–142.