19

Particle Swarm Optimization Applied to Parameters Learning of Probabilistic Neural Networks for Classification of Economic Activities

Patrick Marques Ciarelli, Renato A. Krohling and Elias Oliveira
Universidade Federal do Espírito Santo
Brazil

1. Introduction

Automatic text classification and clustering are still very challenging computational problems to the information retrieval (IR) communities both in academic and industrial contexts. Currently, a great effort of work on IR, one can find in the literature, is focused on classification and clustering of generic content of text documents. However, there are many other important applications to which little attention has hitherto been paid, which are as well very difficult to deal with. One example of these applications is the classification of companies based on the descriptions of their economic activities, also called mission statements, which represent the business context of the companies' activities, in other words, the business economic activities from free text description by the company's founders.

The categorization of companies according to their economic activities constitute a very important step towards building tools for obtaining correct information for performing statistical analysis of the economic activities within a city or country. With this goal, the Brazilian government is creating a centralized digital library with the business economic activity descriptions of all companies in the country. This library will serve the three government levels: Federal; the 27 States; and more than 5.000 Brazilian counties. We estimate that the data related to nearly 1.5 million companies will have to be processed every year (DNRC, 2007) into more than 1.000 possible different activities. It is important to highlight that the large number of possible categories makes this problem particularly complex when compared with others presented in the literature (Jain et al., 1999; Sebastiani, 2002).

In this paper, we proposed a slightly modified version of the standard structure of the probabilistic neural network (PNN) (Specht, 1990) so that we could deal with the multi-label problem faced in this work. We compared the PNN performance trained by a canonical Particle Swarm Optimization (PSO) and a Bare Bones Particle Swarm Optimization (BBPSO). Our results show that, in the categorization of free text descriptions of economic activities, the PNN trained by BBPSO got slightly better results than the PNN trained by PSO.

This work is organized as follows. In Section 2, we detail more the characteristics of the problem and its importance for the government institutions in Brazil. Related works are mentioned in Section 3. We describe our probabilistic neural network algorithm in Section 4. Section 5 describes the Particle Swarm Optimization algorithm and a special version named Bare Bones Particle Swarm Optimization. In Section 6, the experimental results are discussed. Finally, we present our conclusions and indicate some future paths for future research in Section 7.

2. The Problem of Multi-label Text Categorization

In many countries, companies must have a contract (Articles of Incorporation or Corporate Charter, in USA) with the society where they can legally operate. In Brazil, this contract is called a social contract and must contain the statement of purpose of the company – this statement of purpose describe the business activities of the company and must be categorized into a legal business activity by Brazilian government officials. For that, all legal business activities are cataloged using a table called National Classification of Economic Activities, for short, CNAE (CNAE, 2003).

To perform the categorization, the government officials (at the Federal, State and County levels) must find the semantic correspondence between the company economic activities description and one or more entries of the CNAE table. There is a numerical code for each entry of the CNAE table and, in the categorization task, the government official attributes one or more of such codes to the company at hand. This can happen on the foundation of the company or in a change of its social contract, if that modifies its economic activities.

The work of finding the semantic correspondence between the company economic activities description and a set of entries into the CNAE table are both very difficult and labor-intensive task. This is because of the subjectivity of each local government officials who can focus on their own particular interests so that some codes may be assigned to a company, whereas in other regions, similar companies, may have a totally different set of codes. Sometimes, even inside of the same state, different level of government officials may count on a different number of codes for the same company for performing their work of assessing that company. Having inhomogeneous ways of classifying any company everywhere in all the three levels of the governmental administrations can cause a serious distortion on the key information for the long time planning and taxation. Additionally, the continental size of Brazil makes this problem of classification even worse.

In addition, the number of codes assigned by the human specialist to a company can vary greatly, in our dataset we have seen cases where the number of codes varied from 1 up to 109. However, in the set of assigned codes, the first code is the main code of that company. The remaining codes have no order of importance.

Due to this task is up to now decentralized, we might have the same job being performed many times by each of the three levels of the government officials. Nevertheless, it is known that there has been not enough staff to do this job properly.

For all these reasons, the computational problem addressed by us is mainly that of automatically suggesting the human classifier the semantic correspondence between a textual description of the economic activities of a company and one or more items of the CNAE table. Or, depending on the level of certainty the algorithms have on the automatic classification, we may consider bypassing thus the human classifier.

2.1 Metrics for Evaluating of Multi-label Text Categorization

Typically, text categorization is mainly evaluated by the Recall and Precision metrics in the single-labled cases (Baeza-Yates & Ribeiro-Neto, 1998). Nonetheless, other authors have already proposed different metrics for multi-label categorization problems (Schapire & Singer, 2000; Zhang & Zhou, 2007).

Formalizing the problem we have at hand, text categorization may be defined as a task of assigning documents to a predefined set of categories, or classes (Sebastiani, 2002). In multi-label text categorization a document may be assigned to one or more categories. Let D be the domain of documents, $C = \{c_1, c_2, \ldots, c_{|C|}\}$ a set of predefined categories, and $W = \{d_1, d_2, \ldots, d_{|W|}\}$ an initial set of documents previously categorized by some human specialists into subsets of categories of C.

In multi-label learning, the training (-and validation) set $TV = \{d_1, d_2, \ldots, d_{|TV|}\}$ is composed of a number of documents, each associated with a subset of categories in C. TV is used to train and validate (actually, to tune eventual parameters of) a categorization system that associates the appropriate combination of categories to the characteristics of each document in the TV. The test set $Te = \{d_{|TV|+1}, d_{|TV|+2}, \ldots, d_{|W|}\}$, on the other hand, consists of documents for which the categories are unknown to the automatic categorization systems. After being trained, as well as tuned, by the TV, the categorization systems are used to predict the set of categories of each document in Te.

A multi-label categorization system typically implements a real-valued function of the form $f : D \times C \to \Re$ that returns a value for each pair $\langle d_j, c_i \rangle \in D \times C$ that, roughly speaking, represents the evidence for the fact that the test document d_j should be categorized under the category $c_i \in C_i$, where $C_i \subset C$. The real-valued function $f(.,.)$ can be transformed into a ranking function $r(.,.)$, which is an one-to-one mapping onto $\{1,2,\ldots,|C|\}$ such that, if $f(d_j, c_1) > f(d_j, c_2)$, then $r(d_j, c_1) < r(d_j, c_2)$. If C_i is the set of proper categories for the test document d_j, then a successful categorization system tends to rank categories in C_i higher than those not in C_i. Additionally, we also use a threshold parameter so that those categories that are ranked above the threshold τ (i.e., $c_k \mid f(d_j, c_k) \geq \tau$) are the only ones to be assigned to the test document.

We have used five multi-label metrics discussed by Zhang & Zhou (2007) to evaluate the categorization performance of PNN: *hamming loss*, *one-error*, *coverage*, *ranking loss*, and *average precision*. We now present each of these metrics:

- **Hamming Loss (hloss_j)** evaluates how many times the test document d_j is misclassified, i.e., a category not belonging to the document is predicted or a category belonging to the document is not predicted.

$$\text{hloss}_j = \frac{1}{|C|} \left| P_j \Delta C_i \right| \tag{1}$$

where $|C|$ is the number of categories and Δ is the symmetric difference between the set of predicted categories P_j and the set of appropriate categories C_i of the test document d_j. The predicted categories are those with rank higher than the threshold τ.

- **One-error (one-error$_j$)** evaluates if the top ranked category is present in the set of proper categories C_i of the test document d_j.

$$\text{one - error}_j = \begin{cases} 0 & \text{if} \quad \arg\max_{c \in C} f(d_j, c) \in C_i \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where $\arg\max_{c \in C} f(d_j, c)$ returns the top ranked category for the test document d_j.

- **Coverage (coverage$_j$)** measures how far we need to go down the rank of categories in order to cover all the possible categories assigned to a test document.

$$\text{coverage}_j = \max_{c \in C_i} r(d_j, c) - 1 \tag{3}$$

where $\max_{c \in C_i} r(d_j, c)$ returns the maximum rank for the set of appropriate categories of the test document d_j.

- **Ranking Loss (rloss$_j$)** evaluates the fraction of category pairs $\langle c_k, c_l \rangle$, for which $c_k \in C_i$ and $c_l \in \overline{C}_i$, that are reversely ordered for the test document d_j:

$$\text{rloss}_j = \frac{\left| \left\{ (c_k, c_l) | f(d_j, c_k) \le f(d_j, c_l) \right\} \right|}{|C_i||\overline{C}_i|} \tag{4}$$

where $(c_k, c_l) \in C_i \times \overline{C}_i$, and $\overline{C}_i$ is the complementary set of C_i in C.

- **Average Precision (avgprec$_j$)** evaluates the average of precisions computed after truncating the ranking of categories after each category $c_i \in C_i$ in turn:

$$\text{avgprec}_j = \frac{1}{|C_i|} \sum_{k=1}^{|C_i|} \text{precision}_j(R_{jk}) \tag{5}$$

where R_{jk} is the set of ranked categories that goes from the top ranked category until a ranking position k where there is a category $c_i \in C_i$ for d_j, and $\text{precision}_j(R_{jk})$ is the number of pertinent categories in R_{jk} divided by $|R_{jk}|$.

For *p* test documents, the overall performance is obtained by averaging each metric, that is,

$$\text{hloss} = \frac{1}{p} \sum_{j=1}^{p} \text{hloss}_j, \qquad \text{one - error} = \frac{1}{p} \sum_{j=1}^{p} \text{one - error}_j, \qquad \text{coverage} = \frac{1}{p} \sum_{j=1}^{p} \text{coverage}_j,$$

$\text{rloss} = \frac{1}{p}\sum_{j=1}^{p}\text{rloss}_j$, $\text{avgprec} = \frac{1}{p}\sum_{j=1}^{p}\text{avgprec}_j$. On the one hand, the smaller the value of *hamming loss*, *one-error*, *coverage* and *ranking loss*, the better the performance of the categorization system. On the other hand, for the *average precision*, the larger the value the better the performance. So, the problem can be formulated as an optimization problem, where the performance is optimal when hloss = one-error = rloss = 0 and avgprec = 1.
In the next section are mentioned some related works regarding the problem of economic activities classification.

3. Related Works

The authors in (Souza et al., 2007) are among the first to tackle the problem of economic activities classification. In their work they compared the results achieved between a Nearly Neighbors algorithm approach and a Weightless Neural Network, called VG-RAM WNN, using a metric to evaluate the performance equivalent to 1 – one-error, defined in Section 2.1. In the first algorithm they got the performance of 63.36%, while VG-RAM WNN showed to be slightly better, with a performance of 67.56%. However, the use of a single metric seemed to be not enough for evaluating multi-labled problems.
A different approach was performed by (Oliveira et al., 2007). In this work were used 83 arrays of small standard PNN for classification, whose main metrics used were Recall and Precision. However, it was noted to be very difficult to merge the results returned of each neural network array node. Thus the performance of the array as a whole was harmed. Although it has found a reasonable value for the Recall, the value for the Precision was very low, since almost every neural networks returned at least one class to each instance of test.
A PNN with a slightly modified architecture to treat problems of multi-label classification was proposed in (Oliveira et al., 2008). Such neural network presents advantage over the array of small standard PNN approach, used in (Oliveira et al., 2007), because only one PNN is used to solve the problem of multi-label classification. Whereas, in the previous approach, we need to build many neural networks (83 in that case) which complicate the process of optimization.
The results achieved in (Oliveira et al., 2008) using the proposed PNN were better than the achieved using the Multi-label k-Nearest Neighbors (ML-kNN) algorithm. The ML-kNN was considered to be the best algorithm for all the database used in (Zhang & Zhou, 2007). In order to evaluate the performance of the algorithms, the authors in that work used the metrics presented in the Section 2. Moreover, the parameters of these algorithms were optimized using a Genetic Algorithm (GA).
The cited previous works used the same database that we present in this work, but the division of the database was performed in a different way for each work, making it difficult conducting a comparison of results among them. However, in this work we will divide the database in a similar way to used in (Oliveira et al., 2008), making possible a comparison among results.
Another very close multi-label problem to one we are presenting in this paper, concern with the economic activities classification, is that of patent categorization (Li et al., 2007). Our problem and that are both based on free text descriptions of variety topics. So a large volume of patents documents, are usually, up to these days, manually classified by the patent offices, this is a labor-intensive and time-consuming task. A patent document may

cite another patent document, or articles, for comparing or contrasting reasons. Therefore, besides using the content categorization approach, the authors in (Li et al., 2007) proposed to extract and use the direct hyperlink citation relationships among patent documents in order to improve the quality of the whole process of classification. Hyperlink citation is a similar strategy some researchers have been widely applied to web page classification studies. The experiments were conducted on a nanotechnology-related patent dataset from the USPTO. The training dataset contained 13,913 instances, and the testing data set 4,358 data instances. The average of category for document was 36, and the total of categories was up to 426. The results by the K_{Gra} kernel proposed approach yielded 86.67% accuracy overcome the 81% of manually processing and the results of previous work (Koster et al., 2003).
In the following, we describe a slightly modified Probabilistic Neural Network (PNN) used to solve the optimization problem of text categorization.

4. Probabilistic Neural Network Architecture

The Probabilistic Neural Network was first proposed by Donald Specht in 1990 (Specht, 1990). This is an artificial neural network for nonlinear computing, which approaches the Bayes optimal decision boundaries. This is done by estimating the *probability density function* of the training dataset using the Parzen nonparametric estimator (Parzen, 1962).
The literature has shown that this type of neural network can yield similar results, sometimes superior, in pattern recognition problems when compared with others techniques (Fung et al., 2005; Patra et al., 2002).
The original Probabilistic Neural Network algorithm was designed for single-label problems. Thus, we slightly modified its standard architecture, so that it is now capable of solving multi-label problem addressed in this work.
In our modified version, instead of four, the Probabilistic Neural Network is now composed of only three layers: the *input* layer, the *pattern* layer and the *summation* layer, as depicted in Figure 1. Thus like the original, this version of Probabilistic Neural Network needs only one training step, thus its training is very fast compared to the others feedforward neural networks (Duda et al., 2001; Haykin, 1998). The training consists in assigning each training sample w_i of class C_i to a neuron of pattern layer of class C_i. Thus the weight vector of this neuron is the characteristics vector of the sample.
For each pattern x passed by the input layer to a neuron in the pattern layer, it computes the output for x. The computation is performed by Equation 6.

$$F_{k,i}(x) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{x^t w_{ki} - 1}{\sigma^2} \right) \tag{6}$$

where x is the pattern characteristics input vector, and the w_{ki} is the k^{th} sample for a neuron of class C_i, $k \in N_i$, whereas N_i is the number of neurons of C_i. In addition, x was normalized so that $x^t x = 1$ and $w_{ki}{}^t w_{ki} = 1$. The parameter σ is the Gaussian standard deviation, which determines the receptive field of the Gaussian curve.

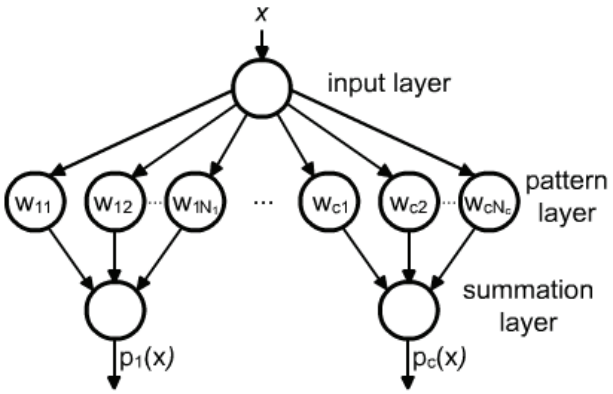


Figure 1. The modified Probabilistic Neural Network architecture

The next step is the summation layer. In this layer, all weight vectors are summed according to Equation 7, in each cluster C_i producing $p_i(x)$ values, where $|C|$ is the total number of classes.

$$p_i(x) = \sum_{k=1}^{N_i} F_{k,i}(x) \qquad (7)$$
$$k = 1,2,\ldots,N_i; \quad i = 1,2,\ldots,|C|$$

Finally, for the selection of the classes, which will be assigned by neural network to each sample, we consider the most likely classes pointed out by the summation layer based on a chosen threshold.

Differently from other types of neural networks, such as the feedforward one (Haykin, 1998), the probabilistic neural network proposed needs few parameters to be configured: the σ, (see Equation 6) and the determination of threshold value. The σ is used to narrow the receptive field of the Gaussian curve in order to strictly select only the more likely inputs for a given class. Other advantages of the probabilistic neural networks is that it is easy to add new classes, or new training inputs, into the already running structure, which is good for on-line applications (Duda et al., 2001). Moreover, it is reported in the literature (Duda et al., 2001) that it is also easy to implement this type of neural network in parallel. On the other hand, one of its drawbacks is the great number of neurons in the pattern layer, which can be, nevertheless, mitigated by an optimization on the number of the neuron (Georgiou et al., 2004; Mao et al., 2000).

Next, we propose a PSO algorithm to find out the σ parameters and tune the PNN automatically.

5. The Canonical and the Bare Bones Particle Swarm Optimization

Particle Swarm Optimisation (PSO) has its origins in the simulation of bird flocking developed by Reynolds (1987) and was further developed in the context of optimization by Eberhart and Kennedy (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995). PSO is initialised with a population of random solutions. Each potential solution in PSO is also

associated with a randomised velocity, and the potential solutions, are called *particles*, that move in the search space. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest*. Another *best* value that is tracked by the *global* version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This location is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of each particle moving toward its *pbest* and *gbest* locations (global version of PSO). Acceleration is weighted by random terms, with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations, respectively. The PSO algorithm consists basically in updating the velocities and positions of the particle, respectively as follows in Equations 8 and 9 (Clerc & Kennedy, 2002):

$$v_i(t+1) = \lambda\left[v_i(t) + c_1 rand_1\left(p_{best_i} - x_i(t)\right) + c_2 rand_2\left(g_{best} - x_i(t)\right)\right] \tag{8}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{9}$$

$$\text{with} \quad \lambda = \frac{2}{\left|\sqrt{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}\right|}, \quad \text{where} \quad \varphi = c_1 + c_2, \quad \varphi > 4$$

where:

- $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$ is the position of the i^{th} particle in the *n*-dimensional search space;
- $v_i = [v_{i1}, v_{i2}, \ldots, v_{in}]^T$ is the velocity of the i^{th} particle;
- p_{best_i} is the best previous i^{th} particle position;
- g_{best} is the best particle among all particles;
- λ is the constriction factor;
- c_1 and c_2 are positive constants;
- $rand_1$ and $rand_2$ are random numbers in the range [0;1] generated using the uniform probability distribution.

Usually, when the constriction factor is used, φ is set to 4.1 ($c_1 = c_2 = 2.05$), and the constriction factor λ is 0.729. In this paper, it is assumed minimization problems unless stated otherwise.

In the meantime different versions of PSO have been proposed by (Krohling & Coelho, 2006). In this work we focus on the Bare Bones PSO (Kennedy, 2003). The Bare Bones PSO (BBPSO) eliminates the velocity item and the Gaussian distribution is used to sampling the search space based on the global best (gbest) and the personal best (pbest) particle. So, the Equations 8 and 9 are replaced by Equation 10:

$$N = \left(\mu_i, \sigma_i^2\right) \tag{10}$$

$$\text{with} \quad \mu_i = \frac{\left(g_{best} + p_{best_i}\right)}{2}, \quad \sigma_i = \left|g_{best} - p_{best_i}\right|$$

where N denotes the Gaussian distribution.

This version of PSO presents some advantages over other versions because its reduced numbers of parameters of the algorithms to be tuned. The BBPSO is described in the Listing 1.

PSO and BBPSO Algorithms

Input parameters: swarm size P

FOR each particle i

// random initialization of a population of particles with positions x_i using uniform

// probability distribution.

$x_i = \overline{x_i} + \left(\underline{x_i} - \overline{x_i}\right) \cdot u_i$ // $\underline{x_i}$ and $\overline{x_i}$ stands for the lower and upper bound, //respectively, and u_i is a random number.

$p_{best_i} = x_i$

compute $f(x_i)$ // fitness evaluation.

$p_{gbest} := \arg\min\{f(x_i)\}$ // global best particle.

END FOR

DO

FOR each particle i

update the position x_i according to Equations 8 and 9 if PSO

update the position x_i according to Equation 10 if BBPSO

compute $f(x_i)$ // fitness evaluation

IF $f(x_i) < f(p_{best_i})$ THEN // update of the personal best.

$p_{best_i} = x_i$

IF $f(x_i) < f(p_{gbest})$ THEN // update the global best.

$p_{gbest} = p_{best_i}$

END FOR

WHILE termination condition not met.

Output: p_{gbest}, $f(p_{gbest})$.

Listing 1 PSO and BBPSO Algorithms.

6. Experimental Results

We employed a series of experiments to compare PNNs optimized using canonical PSO and BBPSO. We used a dataset containing 3264 documents of free text business descriptions of Brazilian companies categorized into a subset of 764 CNAE categories. This dataset was

obtained from real companies placed in Vitoria County in Brazil. The CNAE codes of each company in this dataset were assigned by Brazilian government officials trained for this task. Then we evenly partitioned the whole dataset into four subsets of equal size of 816 documents. We joined to this categorizing dataset the brief description of each one of the 764 CNAE categories, totalizing 4028 documents. Hence, in all training (-and validation) set, we adopted the 764 descriptions of CNAE categories and a subset of 816 business description documents, and, as the test set, the other three subsets of business descriptions totalizing 2448 documents.

6.1 Categorization of Free-text Descriptions of Economic Activities

We pre-processed the dataset via term selection – a total of 1001 terms were found in the database after removing stop words and trivial cases of gender and plural; only words appearing in the CNAE table were considered. After that, each document in the dataset was described as a multidimensional vector using the Bag-of-Words representation (Dumais et al., 1998), i.e., each dimension of the vector corresponds to the number of times a term of the vocabulary appears in the corresponding document. Table 1 summarizes the characteristics of this dataset (dataset available at http://www.inf.ufes.br/~elias/vitoria.tar.gz).

#C	#t	Training set				Test/validation set			
		NTD	DC	CD	RC	NTD	DC	CD	RC
764	1001	4.65	0.00	1.00	100.00	10.92	74.48	4.27	85.21

Table 1. Characteristics of the CNAE dataset

In this Table #C denotes the number of categories, #t denotes the number of terms in the vocabulary, NTD denotes the average number of terms per document, DC denotes the percentage of documents belonging to more than one category, CD denotes the average number of categories for each document, and RC denotes the percentage of rare categories, i.e., those categories associated with less than 1% of the documents of the dataset. The training set is composed by 764 categories descriptions belonging at CNAE table, where each description is concerning just one category and there is only one description by category (one to one relationship), resulting in CD equal 1 and DC equal 0. As there are 764 instances of training and just one instance for category, the index RC is equal 100%. On the other hand, the test/validation set is composed by 3264 instances, where 74.48% of instances are assigned to more than one category and the average number of categories of each instance is more than 4 per document. However, like we said in Section 2, this number vary greatly. Moreover, we can note that RC value is high since there are few instances by category.

The PNNs parameters σ, in Equation 6, were optimized for each class of the dataset and just one threshold τ value for the whole neural network, resulting in 765 parameters, i. e., each particle is represented by a 765-dimensional vector. This is a quite huge amount of parameters for optimization.

To tune these parameters we divided the training set (-and validation) set into a training set, which was used to inductively build the categorizer, and a validation set, which was used to evaluate the performance of the categorizer in the series of experiments aimed at parameter optimization. The training set is composed of 764 descriptions of CNAE classes and the validation set of 816 business description documents described previously. As a result, we

carried out a sequence of experiments with PSO and BBPSO. For each one of these algorithms was carried out 48 experiments:

- 4 experiments each using algorithm with 100 particles and 500 iterations;
- 4 experiments each using algorithm with 50 particles and 500 iterations;
- 40 experiments each using algorithm with 50 particles and 100 iterations.

The two first experiments set were used to evaluate the performance of the algorithms for different population sizes. The last 40 experiments were used for a statistical analysis.

In Figures 2 and 3 are shown the performance of the PNN optimized in function of the number of iterations for 100 and 50 particles, respectively. Where is written in the legend 1st subset means that the first subset was used for validation and the 764 descriptions were used for training, in a similar way this is valid for others cases. The continuous lines are the results of the canonical PSO algorithm and the dotted lines are the results of the BBPSO algorithm. Here, the performance value is a linear combination of the several metrics, where these metrics were described in the Section 2. Thus, performance is the sum of the hamming loss, one error, coverage, ranking loss and $\overline{\text{average precision}}$, where $\overline{\text{average precision}} = 1 -$ average precision. The coverage value was divided by the factor $|C|-1$ to normalize it and keep it in the same scale of the others metrics. A strategy for optimization could be the use of weighted metrics, however in this work was regarded the same value of importance for every metrics.

In both figures the smaller the value of the performance, the better the performance of the neural network. We can observe in both figures that the BBPSO algorithm presented better results than the canonical PSO algorithm. Although the determination of the optimal swarm size is beyond the scope of this work, can be noted that exist no big differences between the results obtained with 100 particles and 50 particles. Moreover, there is a large gain of performance until the 100th iteration and a gain slower in the next iterations. Because of this and since the experiments require substantial amount of run time, we carry out others experiments using 50 particles and 100 iterations for statistical analysis purposes.

In the Table 2 are shown the best, mean, median, standard deviation and worst results obtained in the validation with PSO and BBPSO. The results in bold indicate the best results found for each subset. We can observe in Table 2 that BBPSO finds slightly better results than the canonical PSO.

After tuning, the multi-label categorizers were trained with the 764 descriptions of CNAE categories and tested with the 2448 documents of the test set. The Table 3 shows the best, mean, median, standard deviation and worst results found in the validation with PSO and BBPSO. In this table, where is written 1st means the 1st subset for validation and the others subsets for test, in a similar way this is valid for the other subsets. Again, the results in bold are the best results found for each subset. Similarly as occurred in Table 2, Table 3 also shows that the BBPSO performs slightly better than the PSO.

The mean of results achieved for each metric are shown in Table 4 and 5 for the PNN trained by canonical PSO and BBPSO, respectively. Comparing the results found in this tables we noticed that there weren't significant differences among them, this indicates that the proposed PNN presents certain robustness on the dataset used for training/validation.

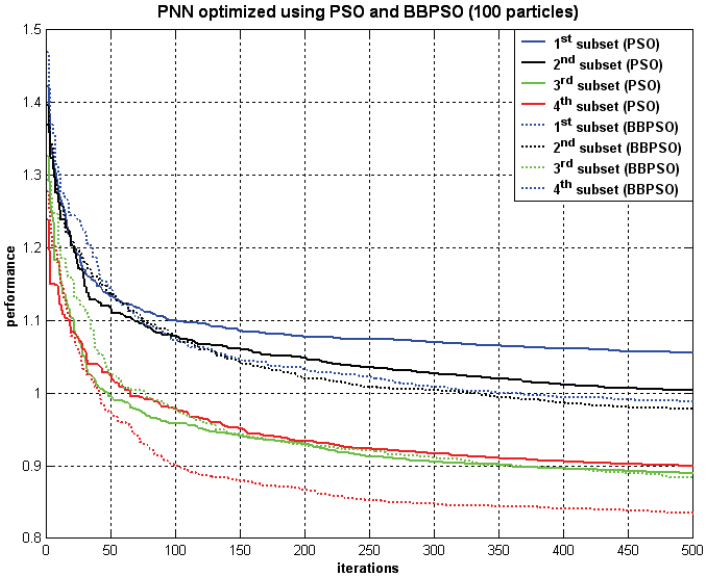


Figure 2. Experimental results of validation of the PNN using PSO and BBPSO with 100 particles and 500 iterations

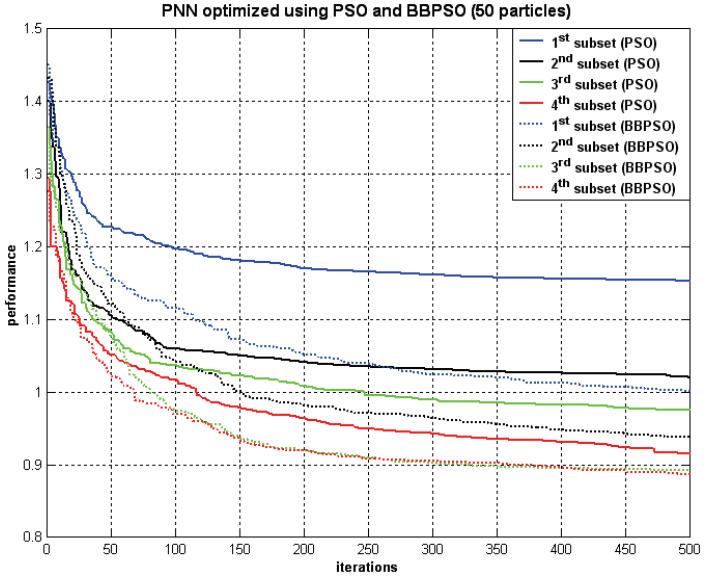


Figure 3. Experimental results of validation of the PNN using PSO and BBPSO with 50 particles and 500 iterations

Subset	Algorithm	Best	Mean	Median	Std. Deviation	Worst
1st	PSO	1.1334	1.1744	1.1617	0.0292	1.2194
	BBPSO	**1.0789**	**1.1107**	**1.1163**	**0.0187**	**1.1344**
2nd	PSO	1.0596	1.0971	1.1003	**0.0176**	1.1241
	BBPSO	**1.0334**	**1.0652**	**1.0671**	0.0192	**1.0864**
3rd	PSO	1.0320	1.0475	1.0488	**0.0087**	1.0608
	BBPSO	**0.9622**	**0.9902**	**0.9842**	0.0217	**1.0257**
4th	PSO	0.9741	1.0060	1.0072	0.0202	1.0435
	BBPSO	**0.9363**	**0.9553**	**0.9558**	**0.0110**	**0.9746**

Table 2. Information about the validation phase

Subset	Algorithm	Best	Mean	Median	Std. Deviation	Worst
1st	PSO	1.1465	1.1766	1.1671	0.0284	1.2332
	BBPSO	**1.1204**	**1.1361**	**1.1358**	**0.0146**	**1.1689**
2nd	PSO	1.1190	1.1632	1.1679	0.0217	1.1853
	BBPSO	**1.1107**	**1.1429**	**1.1432**	**0.0209**	**1.1798**
3rd	PSO	1.1537	1.1873	1.1912	0.0203	1.2202
	BBPSO	**1.1375**	**1.1632**	**1.1633**	**0.0184**	**1.2026**
4th	PSO	1.1841	1.2260	1.2260	0.0345	1.3057
	BBPSO	**1.1555**	**1.1810**	**1.1826**	**0.0167**	**1.2094**

Table 3. Information about the test phase

Subeset	Hamming loss	One-error	Coverage	Ranking loss	Average precision
1st	0.0056	0.3708	144.3723	0.0835	0.4725
2nd	0.0056	0.3688	142.2521	0.0874	0.4850
3rd	0.0055	0.3756	143.9365	0.0912	0.4737
4th	0.0056	0.3857	154.7565	0.0937	0.4619

Table 4. Results achieved with PNN trained by canonical PSO

Subeset	Hamming loss	One-error	Coverage	Ranking loss	Average precision
1st	0.0056	0.3544	143.9079	0.0749	0.4875
2nd	0.0056	0.3592	145.8953	0.0805	0.4937
3rd	0.0055	0.3648	147.5607	0.0842	0.4847
4th	0.0056	0.3634	155.6463	0.0874	0.4794

Table 5. Results achieved with PNN trained by BBPSO

A comparison among the results obtained in this work with the found in (Oliveira et al., 2008) is done in Table 6. The results mentioned are the mean of the four subsets for each metric, and for those in bold are the best results found for each one of the metrics. It is important to highlight that such comparison is a little unfair, since the GA algorithm was executed with 80 individuals and 100 generations whereas the PSO and BBPSO were simulated with 50 particles and 100 iterations. Nevertheless, the results achieved to PNNs are similar. Furthermore, the approach using PSO and BBPSO got the best value of coverage and one-error, respectively. We can note that there is a discrepant difference among the performance of MLkNN and the performance obtained with the PNNs.

Again we can note a certain robustness of the PNN, because its performance didn't change significantly when trained by a PSO, BBPSO or GA algorithm.

Metrics	PNN-PSO	PNN-BBPSO	PNN-GA	ML-kNN-GA
Hamming loss	0.0055	0.0055	0.0055	0.0055
One-error	0.3752	**0.3604**	0.3736	0.4952
Coverage	**146.3293**	148.2525	156.4150	303.9029
Ranking loss	0.0889	0.0817	**0.0798**	0.1966
Average precision	0.4732	0.4863	**0.4880**	0.3813

Table 6. Comparison among different approaches of classification

7. Conclusions

The problem of classifying a large number of economic activities descriptions from free text format every day is a huge challenge for the Brazilian governmental administration. This problem is crucial for the long term planning in all three levels of the administration in Brazil. Therefore, an either automatic or semi-automatic manner of doing that is needed for making it possible and also for avoiding the problem of subjectivity introduced by the human classifier.

In this work, we presented an experimental evaluation of the performance of Probabilistic Neural Network on multi-label text classification. We performed a comparative study of probabilistic neural network trained by PSO and BBPSO, using a multi-label dataset for the categorization of free-text descriptions of economic activities. The approach using PSO and BBPSO were compared with GA and it was noted that there weren't significant differences among them.

To our knowledge, this is one of the first few initiatives on using probabilistic neural network for text categorization into a large number of classes as that used in this work and the results are very promising. One of the advantages of probabilistic neural network is that it needs only one parameter to be configured. In addition, the BBPSO employed is an almost parameter free algorithm, just the number of particles needs to be specified.

A direction for future work is to boldly compare the probabilistic neural network performance against other multi-label text categorization methods. Examining the correlation on assigning codes to a set of descriptions of economic activities may further improve the performance of the multi-label text categorization methods. We are planning on doing that in future work.

8. Acknowledgments

We would like to thank Andréa Pimenta Mesquita, CNAE classifications coordinators at Vitoria City Hall, for providing us with the dataset we used in this work. We would also like to thank Min-Ling Zhang for all the help with the ML-KNN categorization tool. Thank to all the colleagues, especially Alberto Ferreira De Souza, Claudine Badue, Felipe M. G. França and Priscila Machado Vieira Lima for their technical support and valuable comments on this work. This work is partially supported by the Internal Revenue Brazilian Service (Receita Federal do Brasil) and Fundação Espírito Santense de Tecnologia— FAPES-Brasil (grant 41936450/2008) for their support of this research work. Furthermore, R. A. Krohling thanks

the partial funding of his research work provided by FAPES/MCT/CNPq (grant 37286374/2007).

9. References

Baeza-Yates, R. & Ribeiro-Neto, B. (1998). *Modern Information Retrieval*. Addison-Wesley, New York, 1st edition.

Clerc, M. & Kennedy, J. (2002). The Particle Swarm: Explosion Stability and Convergence in a Multi-dimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, vol. 6:58 – 73.

CNAE (2003). Classificação Nacional de Atividades Econômicas Fiscal. *IBGE – Intituto Brasileiro de Geografia e Estatística*, Rio de Janeiro, RJ, 1.1 edition. http://www.ibge.gov.br/concla.

DNRC (2007). *Ranking das Juntas Comerciais Segundo Movimento de Constituição, Alteração e Extinção e Cancelamento de Empresas*. Ministério do Desenvolvimento, Indústria e Comércio Exterior – Secretaria do Desenvolvimento da Produção, Departamento Nacional de Registro do Comércio (DNRC).

Duda, R. O.; Hart, P. E. & Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience, New York, 2nd edition.

Dumais, S. T.; Platt, J.; Heckerman, D. & Sahami, M. (1998). Inductive Learning Algorithms and Representation for Text Categorization. In *Proceedings of the 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, MD.

Eberhart, R. C. & Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39 – 43.

Fung, C. C.; Iyer, V.; Brown, W. & Wong, K. W. (2005). Comparing the Performance of Different Neural Networks Architectures for the Prediction of Mineral Prospectivity. *Proceedings of International Conference on Machine Learning and Cybernetics*, 1:pages 394 – 398.

Georgiou, V. L.; Pavlidis, N. G.; Parsopoulos, K. E.; Alevizos, P. D. & Vrahatis, M. N. (2004). Optimizing the Performance of Probabilistic Neural Networks in a Bioinformatic Task. *Proceedings of the EUNITE Conference*, pages pages 34 – 40.

Haykin, S. (1998). *Neural Networks – A Comprehensive Foundation*. Prentice Hall, New Jersey, 2nd edition.

Jain, A. K.; Murty, M. N. & Flynn, P. J. (1999). Data Clustering: a Review. *ACM Computing Surveys*, 31(3):264–323.

Kennedy, J. (2003). Bare Bones Particle Swarms. *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 80 – 87.

Kennedy, J. & Eberhart, R. C. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks* IV, pages 1942 – 1948.

Koster, C. H. A.; Seutter, M. & Beney, J. (2003). Multi-classification of Patent Applications with Winnow. *Ershov Memorial Conference*, vol. 2890, pages 546 – 555.

Krohling, R. A. & Coelho, L. S. (2006). Co-evolutionary Particle Swarm Using Gaussian Distribution to Solving Constraint Optimization Problems. *IEEE Transactions on Systems, Man and Cybernetics*, part B, vol. 36:1407 – 1416.

Li, X.; Chen, H.; Zhang, Z. & Li, J. (2007). Automatic Patent Classification using Citation Network Information: an Experimental Study in Nanotechnology. *Proceedings of the 2007 Conference on Digital Libraries*, pages 419 – 427.

Mao, K. Z.; Tan, K. C. & Ser, W. (2000). Probabilistic Neural-Network Structure Determination for Pattern Classification. *IEEE Transactions on Neural Networks*, 11:1009–1016.

Oliveira, E.; Ciarelli, P. M.; Souza, A. F. & Badue, C. (2008). Using a Probabilistic Neural Network for a Large Multi-label Problem. *10th Brazilian Symposium on Neural Networks*, pages 1 - 6.

Oliveira, E.; Ciarelli, P. M. & Lima, F. O. (2007). The Automation of the Classification of Economic Activities from Free Text Descriptions using an Array Architecture of Probabilistic Neural Network. *VIII Simpósio Brasileiro de Automação Inteligente*, pages 1 - 5.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33:pages 1065 – 1076.

Patra, P. K.; Nayak, M.; Nayak, S. K. & Gobbak, N. K. (2002). Probabilistic Neural Network for Pattern Classification. *IEEE Proceedings of the 2002 International Joint Conference on Neural Networks*, volume 2, pages 1200–1205.

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, vol. 21(4):25 – 34.

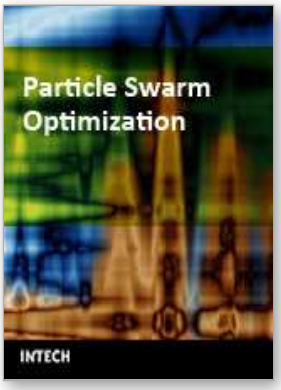
Schapire, R. E. & Singer, Y. (2000). BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing* Surveys, 34(1):1–47.

Specht, D. F. (1990). Probabilistic Neural Networks. *Neural Networks*, 3(1):109–118.

Souza, A. F.; Pedroni, F.; Oliveira, E.; Ciarelli, P. M.; Henrique, W. F. & Veronese, L. (2007). Automated Free Text Classification of Economic Activities using VG-RAM Weightless Neural Networks. *7th International Conference on Intelligent Systems Design and Applications*, pages 1 - 5.

Zhang, M.-L. & Zhou, Z.-H. (2007). ML-KNN: A Lazy Learning Approach to Multi-Label Learning. *Pattern Recognition*., 40:2038–2048.



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0
Hard cover, 476 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling.PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Patrick Marques Ciarelli, Renato A. Krohling and Elias Oliveira (2009). Particle Swarm Optimization Applied to Parameters Learning of Probabilistic Neural Networks for Classification of Economic Activities, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from: http://www.intechopen.com/books/particle_swarm_optimization/particle_swarm_optimization_applied_to_parameters_learning_of_probabilistic_neural_networks_for_clas

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821