Particle Swarm Model Selection for Authorship Verification

Hugo Jair Escalante, Manuel Montes, and Luis Villaseñor

Laboratorio de Tecnologías del Lenguaje
INAOE, Luis Enrique Erro No. 1, 72840, Puebla, México
{hugojair,mmontesg,villasen}@inaoep.mx

Abstract. Authorship verification is the task of determining whether documents were or were not written by a certain author. The problem has been faced by using binary classifiers, one per author, that make individual yes/no decisions about the authorship condition of documents. Traditionally, the same learning algorithm is used when building the classifiers of the considered authors. However, the individual problems that such classifiers face are different for distinct authors, thus using a single algorithm may lead to unsatisfactory results. This paper describes the application of particle swarm model selection (PSMS) to the problem of authorship verification. PSMS selects an ad-hoc classifier for each author in a fully automatic way; additionally, PSMS also chooses preprocessing and feature selection methods. Experimental results on two collections give evidence that classifiers selected with PSMS are advantageous over selecting the same classifier for all of the authors involved.

1 Introduction

Author verification (AV) is the task of deciding whether given text documents were or were not written by a certain author [13]. There is a wide field of application for this sort of methods, including spam filtering, fraud detection, computer forensics and plagiarism detection. In all of these domains, the goal is to confirm or reject the authorship condition for documents with respect to a set of candidate authors, given sample documents written by the considered authors. In the past decade this task was confined to stylography experts who should analyze sample texts from authors to make a decision about the authorship of documents. However, the increasing demand for AV techniques and its wide scope of application have provoked an increasing interest on the scientific community for developing automatic methods for AV.

The scenario we consider is as follows. For each author, we are given sample documents[1] written by her/him as well as documents written by other authors. Features are extracted from documents for representing them in an amenable way for statistical modeling, a model is then built (based on the derived representations for documents) for the author. When a new document arrives, the

[1] We consider digital text documents only, although the proposed methods can be applied to other type of documents (e.g., scanned handwritten documents) as well.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 563–570, 2009.

model must be able to decide whether the document was written by the author or not. Thus, the AV task can be posed as one of binary classification, with a classifier per author. Under this setting, sample documents for the author under consideration are considered positive examples, whereas sample documents for other authors are considered negative examples.

Usually, the same learning algorithm is used to build all of the classifiers corresponding to the set of considered authors. However, the individual problem that each classifier faces is different for distinct authors, thus there is no guarantee that using the same algorithm for all of the authors would lead to acceptable results. Also, while some features may be useful for building the classifier for author *"X"*, the same features may be useless for modeling author *"Y"*. Thus, whenever possible, specific features and classifiers should be considered for different authors. Unfortunately, manually selecting specific features and classifiers for each author is impractical and thus we must resort to automatic techniques.

This paper describes the use of particle swarm model selection (PSMS) for the problem of authorship verification. PSMS can select an *ad-hoc* classifier for each author in a fully automatic way; additionally, PSMS also chooses specific preprocessing and feature selection methods for each problem. This formulation allows us to model each author independently, which results in a more reliable modeling and hence in better verification performance. We conducted experiments on two collections comprising different numbers of authors, samples, lengths of documents and languages, which allows us evaluating the generality of the formulation. Experimental results give evidence that classifiers selected with PSMS are advantageous over selecting the same classifier for all of the authors involved. Also, the methods selected with PSMS can be helpful to gain insight into the distinctive features associated to authors. The rest of this paper describes related work on AV (Section 2); our approach to AV based on PSMS (Section 3); experimental results (Section 4) that show the relevance of the proposed formulation; and the conclusions derived from this work (Section 5).

2 Related Work

Most of the work on AV has focused on developing specific features (stylometric, lexical, character-level, syntactic, semantic) able to characterize the writing style of authors, thus putting emphasis on feature extraction and selection [7,11,10,1], see [13] for a comprehensive review. However, despite these features can be helpful for obtaining reliable models, extracting such features from raw text is a rather complex and time consuming process. In contrast, in this work we adopt a simple set of features to represent the documents and focus on the development of reliable classification models.

The AV problem has been formulated either as a one-class classification problem or as a one-vs-all multiclass classification task. In the former case, sample documents are available from a single author [10] (Did author *"X"* write the document or it was written by any other author?), while in the second case, samples are available from a set of candidate authors [7,11,1] (give the most probable candidate from a list of authors). This paper adopts the second formulation as it is

a more controlled and practical scenario. Those works that have adopted the one-vs-all paradigm consider as positive examples to documents written by an author and negative examples to documents written by the rest of the candidate authors. Then, binary classifiers are built such that they are able to determine whether unseen texts have been written by an author or not.

To the best of our knowledge, all of the reported methods adopting this formulation have used the same learning algorithm to build the classifiers for different authors [7,11,1]. However, using the same learning method for all of the authors does not guarantee that the individual models are the best ones for each author. Also, most of the related works have used the same preprocessing processes and features for all of the authors. The latter leads to obtain consistent outputs across different classifiers, which can be helpful for authorship attribution. Nevertheless, the individual modeling will not be as effective as if we consider specific methods for each author. For that reason, in this work we propose using particular models for each of the authors under consideration.

Model selection is the task of selecting the best model for classification given a set of candidates [8]. Traditionally, a single learning algorithm is considered and the task is to optimize the model's parameters such that its generalization performance is maximized [12]. A problem with most model selection techniques is that they require users to provide prior domain-knowledge or to supply preprocessed data in order to obtain reliable models [6]. PSMS is a more ambitious formulation that selects full models for classification without requiring much supervision [4]. Only a few methods have been proposed for facing the *full model selection* problem, most notably the work by Gorissen et al. [5]. Unlike the latter method, PSMS is more efficient and simple to implement, moreover, PSMS has shown to be robust against overfitting because of the way the search is guided.

3 Particle Swarm Model Selection for Author Verification

Our approach to AV follows the standard scenario described in Section 1, using PSMS for constructing the model for each author. Specifically, we are given N sample documents, each written by one of M authors. Each document d^i is represented by its bag-of-words, $\mathbf{v}^i \in [0,1]^{|V|}$, which is a boolean vector of the size of the collection's vocabulary V; each entry j in $\mathbf{v}^i_j$ indicates whether word $w_j \in V$ appears in document d^i or not. We build M training data sets for binary classification considering the bags-of-words of the N samples and assigning labels to training examples in a different way for each data set. For each author $C_i \in \{C_1, \dots C_M\}$ we build a data set D_i such that we assign the positive label $(+1)$ to documents written by author C_i and the negative one (-1) to documents written by other authors $C_{j:j\neq i}$. Thus we obtain M training sets, each of the form $D_i = \{(\mathbf{v}^1, l^1), \dots, (\mathbf{v}^N, l^N)\}$, with $l^i \in \{-1, 1\}$. At this stage we apply PSMS to select a specific classification model for each author, using the corresponding data sets. Besides classifier, PSMS selects methods for preprocessing and feature selection, and optimizes hyperparameters for the selected methods. The model selected with PSMS is trained using the available data and tested in a separate test set. The rest of this section describes the PSMS technique.

3.1 Particle Swarm Model Selection

PSMS is the application of Particle swarm optimization (PSO) to the model selection problem in binary classification [4]. Given a machine learning toolbox PSMS selects the best combination of methods for preprocessing, feature selection and classification. Additionally, PSMS optimizes hyperparameters of the selected methods. PSMS explores the classifiers space by means of PSO, which optimizes the classification error using training data; as PSO searches both locally and globally, it allows PSMS to overcome, to some extent, overfitting [4].

PSO is a bio-inspired search technique that has proved to be very effective in several domains [3]. The algorithm mimics the behavior of biological societies that share goals and present local and social behavior. Solutions are called particles, at each iteration t, each particle has a position in the search space $\mathbf{x}_i^t =< x_{i,1}^t, \ldots, x_{i,d}^t >$, and a velocity $\mathbf{v}_i^t =< v_{i,1}^t, \ldots, v_{i,d}^t >$ value, with d the dimensionality of the problem. The particles are randomly initialized and iteratively update their positions in the search space as follows $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}$, with $\mathbf{v}_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (\mathbf{p}_i - \mathbf{x}_i^t) + c_2 \times r_2 \times (\mathbf{g}_i - \mathbf{x}_i^t)$; where $\mathbf{p}_i$ is the best position obtained by $\mathbf{x}_i$, $\mathbf{g}_i$ is the best particle in the swarm, c_1 and c_2 constants and r_1, r_2 random numbers, w is the so called inertia weight, see [3] for details. The goodness of particles is evaluated with a fitness function specific for the task at hand. PSO stops when a fixed number of iterations is performed.

In PSMS the particles are full models (i.e., combinations of preprocessing, feature selection and classification methods), codified as numerical vectors. The optimization problem is minimizing an estimate of classification error. In particular, we consider the balanced error rate (BER) as fitness function; $BER = \frac{E_+ + E_-}{2}$, where E_+ and E_- are the error rates in the positive and negative classes, respectively. As the test data are unseen during training, the error of solutions (i.e., full models) is estimated with $k-$fold cross validation (CV) on the training set. Thus, the PSO algorithm is used to search for the model that minimizes the CV-BER. The selected model is considered the classifier for the corresponding author in AV. We consider the PSMS implementation included in the CLOP[2] toolbox. Table 1 shows the methods from which PSMS can choose, see [4] for further details. PSMS has reported outstanding results on diverse binary classification problems without requiring significant supervision [6,4], which makes it attractive for many applications. The application of PSMS to AV arises naturally, as we want to select specific full models for each author.

4 Experimental Results

We report results on two collections described in Table 2. The collections have heterogeneous characteristics which make them particularly useful to test the robustness of PSMS to different training set sizes, dimensionality, languages and number of authors. Both collections have predefined partitions for training/testing that have been used in previous works for authorship attribution [2,9]. We kept the

[2] http://clopinet.com/CLOP

Table 1. Classification (C), feature selection (F) and preprocessing (P) methods considered in our experiments; we show the object name and the number of parameters for each method

Object name	Type	# pars.	Description
zarbi	C	0	Linear classifier
naive	C	0	Naïve Bayes
logitboost	C	3	Boosting with trees
neural	C	4	Neural network
svc	C	4	SVM classifier
kridge	C	4	Kernel ridge regression
rf	C	3	Random forest
lssvm	C	5	Kernel ridge regression
Ftest	F	4	F-test criterion
Ttest	F	4	T-test criterion
aucfs	F	4	AUC criterion
odds-ratio	F	4	Odds ratio criterion
relief	F	3	Relief ranking criterion
Pearson	F	4	Pearson correlation coefficient
ZFilter	F	2	Statistical filter
s2n	F	2	Signal-to-noise ratio
$pc - extract$	F	1	Principal components analysis
svcrfe	F	1	SVC- recursive feature elimination
normalize	P	1	Data normalization
standardize	P	1	Data standardization
$shift - scale$	P	1	Data scaling

Table 2. Collections considered for experimentation

Collection	Training	Testing	Features	Authors	Language	Domain	Ref.
MX-PO	281	72	8,970	5	Spanish	Poetry	[2]
CCAT	2,500	2,500	3,400	50	English	News	[9]

words that appear at least in 5 and 20 documents, for the MX-PO and CCAT collections, respectively. We report average precision (P) and recall (R), as well as the F_1 measure, defined as $F_1 = \frac{2 \times R \times P}{R+P}$, and the BER of the individual classifiers.

Besides applying PSMS as described in Section 3.1 (see **FMS/1** below), we investigate the usefulness of PSMS under two other settings that have not been tested elsewhere. This is with the goal of evaluating the benefits of introducing prior knowledge provided by the user. The considered settings are as follows:

- **FMS/1** selects preprocessing, feature selection and classification methods.
- **FMS/0** selects preprocessing and feature selection methods only.
- **FMS/-1** hyperparameter optimization for a fixed classifier.

Through settings **FMS/0** and **FMS/-1**, the user provides prior knowledge by fixing a classification method. Therefore, better results are expected with these settings. Besides using PSMS for the selection of classifiers, we also used the classifiers shown in Table 1 with default parameters for comparison.

Table 3 shows the average BER and the F_1 measure obtained by methods we tried for both collections. For the **FMS/0** configuration we fixed the classifier to be *zarbi* for both collections, as this algorithm has no hyperparameters to optimize and thus PSMS would be restricted to search for preprocessing and feature selection methods. For **FMS/-1** we tried different configurations, although the

best results were obtained by fixing *neural* and *svc* classifiers for CCAT and MX-PO, respectively.

From Table 3, we can see that classifiers selected with PSMS show better performance than the individual methods. Interestingly, the best results were obtained with the **FMS/0** configuration. Note that we fixed a non-parametric classifier and PSMS selected for preprocessing and feature selection methods. Thus, despite the individual performance of *zarbi* is low, its performance after selecting appropriate methods for preprocessing and feature selection is significantly improved. The performances of the **FMS/1** and **FMS/-1** settings are competitive as well outperforming most of the individual classifiers for both collections. Therefore, in absence of any knowledge about the behavior of the available classifiers it is recommended to use PSMS instead of trying several classifiers and combinations of methods for preprocessing and feature selection.

Table 3. Average BER and F_1-measure for the considered methods in both collections

Col./Clas.	zarbi	naïve	lboost	neural	svc	kridge	rf	lssvm	FMS/-1	FMS/0	FMS/1
BER											
MX-PO	34.64	30.24	29.08	28.59	30.81	31.90	48.01	33.52	26.18	**23.68**	26.88
CCAT	14.24	26.21	15.12	41.50	29.18	27.69	47.01	36.64	35.34	**13.54**	16.39
F_1											
MX-PO	46.26	52.93	53.18	59.25	54.57	52.52	6.66	48.76	58.28	**60.37**	57.09
CCAT	59.69	55.73	47.11	28.46	56.46	51.85	10.58	38.54	44.11	61.17	**63.41**

Table 4 shows the models selected by PSMS under the **FMS/1** configuration for the MX-PO data set. We can see the variety of methods selected, which are different for each author. The BER of the first three authors is below the mean of individual classifiers, while the BER of models for the last two authors is high, even when non-linear models are used for the latter. This suggest that R. Castellanos and R. Bonifaz are more complex to model, and that better features may be needed for building the respective classifiers.

Table 4. Full models selected by PSMS, under **FMS/1**, for the MX-PO collection

Poet	Preprocessing	Feature Selection	Classifier	BER
E. Huerta	standardize(1)	-	zarbi	10.28
S. Sabines	-	-	zarbi	26.79
O. Paz	normalize(0)	Zfilter(3070,0.56)	zarbi	25.09
R. Castellanos	normalize(0)	Zfilter(7121,0.001)	kridge(rbf-γ =0.45)	33.04
R. Bonifaz	shift-scale(1)	-	neural(u=3;iter=15)	35.71

Table 5 shows statistics on the selection of methods for the CCAT data set. As with the MX-PO data set, the classifier that is mostly selected is *zarbi*, used for 68% of the authors, *naïve*, *neural* and *lssvm* come next, whereas *logitboost* and *rf* were not selected. The BER for linear classifiers is below the average BER for **FMS/1**, while the BER of non-linear methods is above the mean, giving evidence of the linearity of the problem. Most of the selected models included methods for

preprocessing and feature selection. The BER of classifiers that used feature selection methods was higher than that of classifiers that were not used. The most used feature selection method was $pc - extract$, used for 19 models; other considered methods were *Ftest* (5), *Ttest* (5), *aucfs* (4) and *svcrfe* (3).

Table 5. Statistics on selection of methods when using PSMS for the CCAT collection

Classifiers						Feature Selection		Preprocessing	
zarbi	**naïve**	**neural**	**svc**	**kridge**	**lssvm**	**With**	**Without**	**With**	**Without**
Frequency of selection									
68%	10%	10%	2%	2%	8%	76%	24%	88%	12%
BER									
14.22	9.88	23.42	3.82	44.01	33.51	14.14	22.28	15.64	16.50

Figure 1 shows the per-author F_1-measure, the best result obtained was for the author *'Karl-Penhaul'* ($F_1 = 96.91\%$), which considered the three preprocessing methods, and *Ftest* for feature selection together with a naïve classifier. The classifier was built on 104 out of the 3,400 features (i.e., words) available, this means that about 100 words are enough for distinguishing this author; interestingly, 35 out of the 104 words selected as relevant were not used in any document of this author, the relevant words *'state'* and *'also'* were used in 41 out of 50 documents written by *'Karl-Penhaul'*.

On the other hand, the worst result was obtained for *'Peter-Humphrey'* ($F_1 = 14.81\%$), which used *normalize* for preprocessing and an *lssvm* classifier. When we used the *zarbi* classifier with the **FMS/0** setting, the classifier selected for this author obtained $F_1 = 45.71\%$, such classifier used the three preprocessing methods and *Zfilter* for selecting the top 234 more relevant features. This represents an improvement of over 30% in terms of F_1 measure, and an important improvement in terms of processing time, also, the result suggest the author *'Peter-Humphrey'* can be better modeled with a linear classifier.

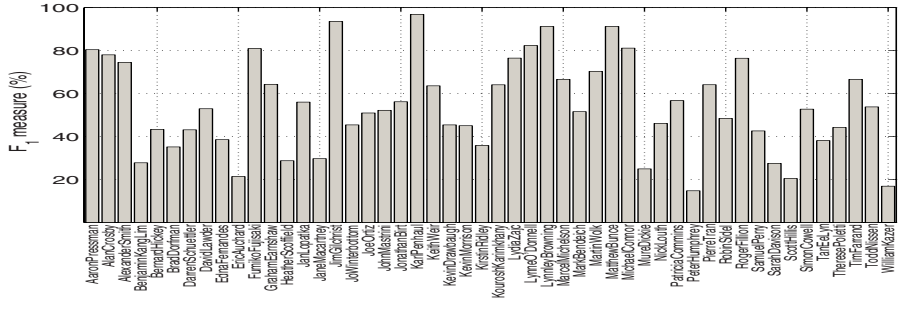

Fig. 1. F_1 measure of different classifiers in CCAT collection

5 Conclusions

We have described the application of particle swarm model selection (PSMS) to the problem of authorship verification. The proposed approach allows us to model each author independently, developing *ad-hoc* models for each author. This is an advantage over previous work that has considered a same learning algorithm for all of the authors. PSMS also selects methods for preprocessing and feature selection, facilitating the design and implementation processes to users. Experimental results show that the proposed technique can obtain reliable models that perform better than those in which the same learning algorithm is used for all of the authors. Results are satisfactory, despite we have used the simplest set of features one may try (i.e., the bag-of-words representation); better results are expected by using more descriptive features. PSMS can also be helpful for analyzing what features are more important for building classifiers for certain authors, which allows us to gain insight into the writing style of authors. Future work includes extending the use of PSMS for the task of authorship attribution and analyzing the writing style of authors by using models selected with PSMS.

Acknowledgements. We thank E. Stamatatos for making available the CCAT data and reviewers for their comments that have helped us to improve this paper. This work was supported by CONACyT under Project Grant CB-2007-01-83459.

References

1. Argamon, S., Marin, S., Stein, S.: Style mining of electronic messages for multiple authorship discrimination. In: Proc. of SIGKDD 2003, pp. 475–480 (2003)
2. Coyotl-Morales, R.M., Villaseñor-Pineda, L., Montes-y-Gómez, M., Rosso, P.: Authorship attribution using word sequences. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 844–853. Springer, Heidelberg (2008)
3. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. Wiley, Chichester (2006)
4. Escalante, H.J., Montes, M., Sucar, E.: Particle swarm model selection. Journal of Machine Learning Research 10, 405–440 (2009)
5. Gorissen, D., Tommasi, L., Croon, J., Dhaene, T.: Automatic model type selection with heterogeneous evolution. In: Proc. of WCCI 2008, pp. 989–996 (2008)
6. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Analysis of the IJCNN 2007 ALvsPK challenge. Neural Networks 21(2–3), 544–550 (2008)
7. Van Halteren, H.: Linguistic profiling for author recognition and verification. In: Proc. of ACL 2004, pp. 199–206 (2004)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
9. Houvardas, J., Stamatatos, E.: N-gram feature selection for author identification. In: Euzenat, J., Domingue, J. (eds.) AIMSA 2006. LNCS (LNAI), vol. 4183, pp. 77–86. Springer, Heidelberg (2006)
10. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proc. of ICML 2004, p. 62 (2004)
11. Luyckx, K., Daelemans, W.: Authorship attribution and verification with many authors and limited data. In: Proc. of COLING 2008, pp. 513–520 (2008)
12. Momma, M., Bennett, K.: A pattern search method for model selection of support vector regression. In: Proc. of SIAM-CDM (2002)
13. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology 60(3), 538–556 (2006)