A Stream-Based Semi-Supervised Active Learning Approach for Document Classification

Mohamed-Rafik Bouguelia, Yolande Belaïd, Abdel Belaïd

2013 12th International Conference on Document Analysis and Recognition

A Stream-Based Semi-Supervised Active Learning Approach for Document Classification

Mohamed-Rafik Bouguelia, Yolande Belaïd and Abdel Belaïd
Université de Lorraine - LORIA, UMR 7503
Vandoeuvre-les-Nancy, F-**54506**, France
Email: {mohamed.bouguelia, yolande.belaid, abdel.belaid}@loria.fr

***Abstract*—We consider an industrial context where we deal with a stream of unlabelled documents that become available progressively over time. Based on an adaptive incremental neural gas algorithm (AING), we propose a new stream-based semi-supervised active learning method (A2ING) for document classification, which is able to actively query (from a human annotator) the class-labels of documents that are most informative for learning, according to an uncertainty measure. The method maintains a model as a dynamically evolving graph topology of labelled document-representatives that we call neurons. Experiments on different real datasets show that the proposed method requires on average only 36.3% of the incoming documents to be labelled, in order to learn a model which achieves an average gain of 2.15-3.22% in precision, compared to the traditional supervised learning with fully labelled training documents.**

I. Introduction

Administrations deal every day with thousands of heterogeneous administrative documents that are daily digitized and must be processed quickly and efficiently in order to enable a gain of productivity for administrations and a gain of reactivity and response time for users. These digitized documents have to be classified in different classes such as bank checks, medical receipts, invoices, prescriptions etc. The objective is to automatically redirect them to topic-specific processing and information extraction mechanisms, or redirect them to humans or services departments that are specialized in their management. The classification of such documents by topic also provide better contextual information and allows to disambiguate some terms of interest; for instance, the account number which may have a different meaning depending on whether the document's category is an incoming invoice or a service request.

However, for many industrial and real-world applications[1], the traditional state of the art methods for document classification like those surveyed in [1] are constrained by some requirements which make them difficult to use properly. There are basically two main reasons.

For the first reason, to achieve a good classification accuracy, the traditional methods need to be trained using many labelled documents, they are fully supervised (i.e. need to manually build a large enough set of labelled documents for the learning to be efficient). However, obtaining a sufficient number of labelled training documents is costly and time-consuming. Semi-supervised learning techniques [2], [3] can learn using both labelled and unlabelled data, and can therefore be used to alleviate the cost of labelling many documents. However, instead of randomly selecting the documents to be labelled, it may be more interesting to let the algorithm chose which documents are more convenient for labelling. This is referred to as active learning [4], [5], [8], i.e. the algorithm queries the labels of some documents from the human annotator according to their importance with respect to learning results.

For the second reason, most of traditional document classification methods need the whole training set (used for learning) to be available beforehand. Most semi-supervised and active learning methods also performs in a batch mode (i.e. they use all the input documents). This requirement is inconvenient when dealing with a massively and continuously arriving stream of documents (theoretically considered as an infinite stream) where the documents become available progressively over time. Therefore, beside the fact that learning is active and weakly supervised, we also consider an online learning configuration where each new document from the stream can be visited only once and used to update the learned model incrementally as soon as it is available. Some related online and semi-supervised methods for text streams are surveyed in [6], however these methods are not active and do not select informative documents for labeling during learning.

We already proposed in [7] an efficient learning method called AING for "adaptive incremental neural gas". AING is a scalable unsupervised incremental learning method which can learn online from a data-stream without being sensitive to initialization parameters. In this paper, we present a semi-supervised active extension of AING (that we call A2ING) for document classification task, which can be trained incrementally from a continuously arriving stream of documents so that it do not need the whole documents to be available beforehand, and do not need the whole training documents to be labelled. The proposed method can learn from both labelled and unlabelled documents by actively querying the labels of the most informative documents according to an uncertainty measure (described in section III-A).

This paper is organized as follows. In section 2 we briefly describe AING, the basis of the proposed method. We extend AING to A2ING, a semi-supervised active learning for document classification in section 3. Then we present our experimental evaluation on real datasets in section 4. In section 5, we give the conclusion and we present some perspectives of this work.

[1]Based on communication and direct collaboration on real-world industrial problem with the ITESOFT company http://www.itesoft.com

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.126

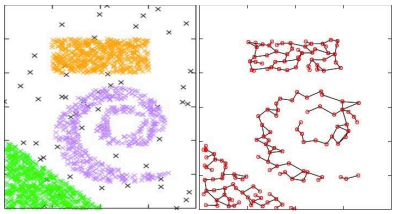

Fig. 1. AING learns the topology of data. Left: different data distribution shapes in a 2 dimensional space. Right: graph topology G of neurons

II. Overview of AING

Let $x \in \mathbb{R}^p$ be the feature-vector of a new document[2]. AING maintains a model as a graph topology G (Fig.1) of document-representatives that we call neurons. Each neuron $y \in G$ is a feature-vector which is continuously maintained and updated by AING.

We consider x to be *far enough* (respectively *close enough*) from a neuron y if the distance between x and y is higher (respectively smaller) than a threshold T_y, which is essentially defined as the mean distance from y to its neighbouring neurons (i.e. neurons $y_i \in N_y$, where N_y is the set of neurons that are linked to y by an edge) [7]. The general AING's method of operation can then be expressed as in Fig. 2 according to the following 3 cases. Let y_1 and y_2 respectively be the nearest and the second nearest neurons from x, such that $\text{dist}(x, y_1) < \text{dist}(x, y_2)$. The learning algorithm incrementally builds and maintains a model G, at each new document arrival, as follows:

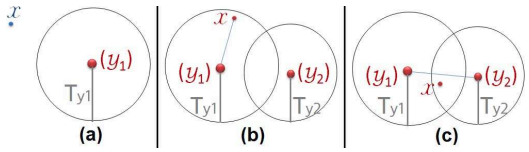

Fig. 2. AING's three cases for generating and adapting neurons

1) if $\text{dist}(x, y_1) > T_{y_1}$ (Fig.2(a)):
   - $G \leftarrow G \cup \{y_{new} | y_{new} = x\}$, i.e., a new neuron y_{new} is generated based on x.
2) if $\text{dist}(x, y_1) < T_{y_1}$ and $\text{dist}(x, y_2) > T_{y_2}$ (Fig.2(b)):
   - $G \leftarrow G \cup \{y_{new} | y_{new} = x\}$
   - Link y_{new} to y_1 by a new edge.
3) if $\text{dist}(x, y_1) < T_{y_1}$ and $\text{dist}(x, y_2) < T_{y_2}$ (Fig.2(c)), we say that x is assigned to y_1:
   - Link y_1 to y_2 by a new edge.
   - $y_1 \leftarrow y_1 + \epsilon_1 \times (x - y_1)$, i.e., updating the feature-vector y_1 to be less distant from x.
   - $\forall y_i \in N_{y_1} : y_i \leftarrow y_i + \epsilon_2 \times (x - y_i)$, i.e., updating the feature-vector of each y_1's neighbouring neuron ($y_i \in N_{y_1}$) to be slightly less distant from x.

When a document x is close enough to its two nearest neurons y_1 and y_2, it is assigned to y_1 (3^{rd} case). This later and its neighbouring neurons are updated (i.e. they move towards x) by a learning rate: ϵ_1 for y_1 and ϵ_2 for its neighbouring neurons. As discussed in [7], generally, a too big learning rate implies instability of neurons, while a too small learning rate implies that neurons do not learn enough from their assigned documents. Typical values are $0 < \epsilon_1 \ll 1$ and $0 < \epsilon_2 \ll \epsilon_1$. Let n_{y_1} be the number of documents assigned to y_1. In AING, $\epsilon_1 = \frac{1}{n_{y_1}}$ is slowly decreasing proportionally to the number of documents assigned to y_1, i.e. the more y_1 learns, the more it becomes stable, and ϵ_2 is simply heuristically set to $\epsilon_2 = \frac{1}{n_{y_1} \times 100}$, that is, 100 times smaller than the actual value of ϵ_1 (i.e. $\epsilon_2 \ll \epsilon_1$).

Note that the method is incremental and do not need to save documents that are previously seen. The maintained topology (G) of neurons evolves dynamically according to new documents.

III. Improvement to semi-supervised active learning for document classification

AING is unsupervised and can not be directly applied to a classification task. In order to be suitable for a document classification task, we extend AING to learn from both labelled and unlabelled documents. However, instead of manually or randomly choosing which documents to label from the incoming stream, we let the algorithm itself decide at each new document arrival, whether or not its class-label should be queried from a human annotator.

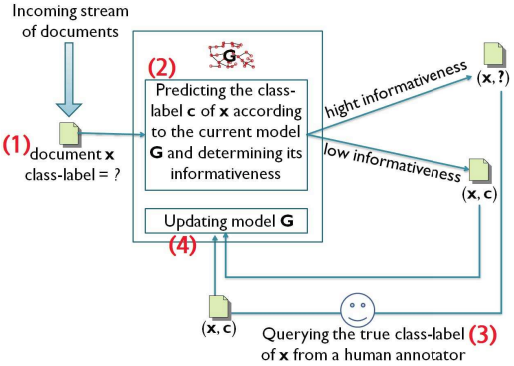

Fig. 3. The general A2ING scheme

We initially get a small number of labelled documents which may be the first few incoming documents from the stream, and use them to initialise the model G with some neurons[3]. Then, each new document represented as a feature-vector x (Fig. 3 (1)) is classified into a class c according to the current model G (Fig. 3 (2)). The classification method derives an index of uncertainty which determines the informativeness of the document x using the method in section III-A. If the method is "uncertain" about the predicted class-label of x, then it is considered informative, and its true class-label c is queried from a human annotator (Fig. 3 (3)). The classified document (x, c) is then learned (Fig. 3 (4)) in order to update and improve the model G as described in section III-B.

[2]Of course, features are application-dependent. We use in the experiments a bag-of-words representation of document, to classify them by their topics.

[3]In our experiments we initialize the model with only 1% of the documents, chosen randomly from the training set.

A. Classification and document informativeness

For each new document x, we use K-Nearest Neighbours method [9] and we derive a probability of belonging to its two most probable classes.

Let $\text{KNN}(x) = \{(y_1, c_{y_1}), ..., (y_K, c_{y_K})\}$ be the K nearest neurons selected from G, sorted in ascending order according to their Euclidean distance to x. Let $P(c|x)$ the probability that the document x belongs to the class c. It is determined as

$$P(c|x) = \frac{\sum_{(y_i, c_{y_i}) \in \text{KNN}(x)} f(y_i, c_{y_i})}{K} \quad (1)$$

$$\text{where } f(y_i, c_{y_i}) = \begin{cases} 1 & \text{if } c_{y_i} = c \\ 0 & \text{otherwise} \end{cases}$$

Let $c_1 = \underset{c}{\text{argmax}}\, P(c|x)$ and $c_2 = \underset{c \neq c_1}{\text{argmax}}\, P(c|x)$, i.e. c_1 and c_2 are respectively the first and the second most probable classes given the document x, such that $P(c_1|x) \geq P(c_2|x)$. Let the quantity $\Delta_{(c_1,c_2|x)} = P(c_1|x) - P(c_2|x)$.

A document with a small Δ value is more uncertain because the probability of belonging to its most probable class c_1 is close to the probability of belonging to its second most probable class c_2; thus the more $\Delta_{(c_1,c_2|x)}$ is close to 0 the more informative is the document x, because knowing the true class-label of such document would be useful for the model G to better discriminate between these classes in section III-B.

To decide if the class-label of a new document x should be queried (i.e. if x is informative), we define a small confidence value δ. If $\Delta_{(c_1,c_2|x)} < \delta$ then the true class-label of x is queried from a human annotator. Otherwise, the document x is classified as c_1 (its most probable predicted class).

To intuitively illustrate what are the uncertain documents (with a low Δ value) which are informative for our model, Fig. 4 (a) shows three overlapped classes of two dimensional data-points, and Fig. 4 (b) shows that the queried data-points are those lying inside an uncertainty region, they are considered informative because knowing their true class-label will help to better separate the overlapped classes.

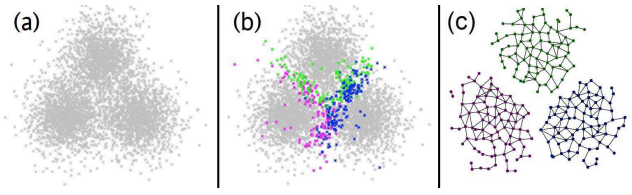

Fig. 4. (a) Three overlapping classes of two-dimensional Gaussian distribution. (b) Queried labels of uncertain data-points. (c) The obtained topology of labelled neurons after removing the inter-class border edges; the overlapped classes are well separated

Note that we can make the confidence value δ adaptive. Suppose that $\Delta_{(c_1,c_2|x)} < \delta$, in this case the true class-label of x is queried from the human annotator. Let us denote this true class-label by c_x^*. If the most probable class-label c_1 was correctly predicted (i.e. $c_1 = c_x^*$) then we can get more confident and slightly decrease the value of δ. Otherwise, if the most probable class-label c_1 was not the true one (i.e. $c_1 \neq c_x^*$) then we can get less confident and slightly increase the value of δ.

B. Updating model G

Let (x, c_x) be a new data where x is the document's feature-vector and c_x its predicted or queried class-label (as show in the previous subsection). We mainly adapt the 3rd case of the AING's algorithm (section II) in order to maximize the separation between the different overlapping classes. Let y_1 and y_2 respectively be the nearest and the second nearest neurons from the document x, such that $\text{dist}(x, y_1) < \text{dist}(x, y_2)$:

1) if $\text{dist}(x, y_1) > T_{y_1}$:
   - $G \leftarrow G \cup \{(y_{new}, c_x)|y_{new} = x\}$, i.e., a new neuron y_{new} labelled with c_x is generated based on x.
2) if $\text{dist}(x, y_1) < T_{y_1}$ and $\text{dist}(x, y_2) > T_{y_2}$:
   - $G \leftarrow G \cup \{(y_{new}, c_x)|y_{new} = x\}$
   - Link y_{new} to y_1 by a new edge.
3) if $\text{dist}(x, y_1) < T_{y_1}$ and $\text{dist}(x, y_2) < T_{y_2}$:
   - Link y_1 to y_2 by a new edge (if not linked)
   - if $c_x = c_{y_1}$:
     - $y_1 \leftarrow y_1 + \epsilon_1 \times (x - y_1)$
     - $\forall y_i \in N_{y_1}$ and $c_x \neq c_{y_i}$:
       - $y_i \leftarrow y_i - \epsilon_2 \times (x - y_i)$
   - if $c_x \neq c_{y_1}$:
     - $y_1 \leftarrow y_1 - \epsilon_1 \times (x - y_1)$
     - $\forall y_i \in N_{y_1}$ and $c_x = c_{y_i}$:
       - $y_i \leftarrow y_i + \epsilon_2 \times (x - y_i)$

The two first cases (i.e. when x is far from y_1, or close to y_1 but far from y_2) are similar to the original AING, however, the generated neuron y_{new} is labelled with the label that was associated to x.

In the third case (i.e. when x is close to both y_1 and y_2), y_2 becomes a neighbouring neuron of y_1 (it is linked to y_1 by an edge) even if they are labelled with different class-labels. We call an edge linking two neurons with different class-labels, an "inter-class border edge"; it allows the algorithm to determine a separation between classes, by moving neurons labelled differently far from each other. Indeed, if y_1 is labelled similarly to x (i.e. $c_x = c_{y_1}$), then y_1 is updated to be less distant from x, and the neighbouring neurons of y_1 which are labelled differently from x are updated to be more distant from x. On the contrary, if y_1 is labelled differently from x (i.e. $c_x \neq c_{y_1}$), then y_1 is updated to be more distant from x, and the neighbouring neurons of y_1 which are labelled similarly to x are updated to be less distant from x. Fig. 4 (c) shows the obtained topology of labelled neurons after removing the inter-class border edges, for the 3 overlapped classes of Fig. 4 (a).

IV. Experimental evaluation

We consider in our experimental evaluation, a total of five real administrative document datasets provided by different clients of ITESOFT company. Each document is firstly processed by an OCR and represented as a bag-of-words, which is a sparse feature-vector containing the occurrence

counts of words in the document. Each dataset has its feature-vectors represented in a p-dimensional space, where p is the vocabulary size[4]. The datasets are of different size and different number of classes (13 to 141 classes) which are defined by the clients who provided the documents.

- **DocSet dataset**: 519 documents for learning, 260 documents for testing, $p = 413$, 13 classes.
- **CAF dataset**: 772 documents for learning, 386 documents for testing, $p = 271$, 141 classes.
- **LIRMM dataset**: 1301 documents for learning, 650 documents for testing, $p = 277$, 24 classes.
- **MMA dataset**: 1728 documents for learning, 863 documents for testing, $p = 292$, 25 classes.
- **APP dataset**: 13161 documents for learning, 6581 documents for testing, $p = 328$, 139 classes.

We consider our proposed method "A2ING" (Active Adaptive Incremental Neural Gas) and an incremental svm method "LASVM" [8] in an active mode, where the data-points that are closest to the decision boundary of svm are those which are most likely to be labelled. We also consider some main classifiers as a reference in comparing the results (KNN [9], LogitBoost [10], NaiveBayes [11] and RandomForest [12]). As a reminder, A2ING offers two additional qualities against the considered methods (except LASVM), (1) only the labels of some informative documents are queried during learning (unlike the considered methods which are fully supervised), and (2) it processes documents one by one (unlike the considered methods which perform in a batch mode where each document may be revisited many times during learning).

We initially set the parameter δ of A2ING for all the datasets to 0.3, which represents a pretty hight initial confidence value, then it is adapted during learning as described at the end of section III-A. Beside the rate of labels which are queried during learning, we consider as evaluation measures the error rate and the weighted average precision and recall.

The obtained results are shown in Table I. Firstly, for A2ING, the number of documents that were manually labelled during learning (by querying their labels from a human annotator) is between 22.5% and 53.8% (36.3% on average) of the total number of documents used for learning; this is better than the labeled rate obtained by LASVM. The other methods, since they are fully supervised, need the whole dataset to be manually labelled (100% of labels), which is by the way usually infeasible for real-world applications. From Table I we see that for the *DocSet* and *CAF* datasets, A2ING realises almost the same performances as LogitBoost and NaiveBayes respectively. Concerning the dataset *LIRMM*, although A2ING requires only 22.5% of labels, it achieves a better performance than KNN, LogitBoost and NaiveBayes, however, RandomForest and LASVM achieved a best performance than A2ING. For the *MMA* dataset, LASVM slightly outperform A2ING; while for the *APP* datasets, A2ING achieves the best performances in terms of error rate and recall. Finally, the average results over all datasets is shown in the bottom of table I. We can see that A2ING achieves, on average, the best performances regarding the error rate, precision and recall.

[4]This is the the number of meaningful or frequent words for each dataset

TABLE I. VALIDATION RESULTS

Method	Labels %	Error %	Precision %	Recall %
DocSet dataset				
A2ING	**39.1%**	**19.2**	80.9	80.7
KNN	all	25.7	76.0	74.2
LogitBoost	all	**19.2**	80.9	**80.8**
NaiveBayes	all	25.3	76.6	74.6
RandomForest	all	21.1	77.9	78.8
LASVM	51.05%	20.7	**81.3**	79.2
CAF dataset				
A2ING	**53.8%**	**28.7**	75.7	**71.2**
KNN	all	31.6	74.7	68.4
LogitBoost	all	38.0	69.9	61.9
NaiveBayes	all	**28.7**	**75.8**	**71.2**
RandomForest	all	32.6	72.5	67.4
LASVM	66.9%	29.2	73.9	70.7
LIRMM dataset				
A2ING	**22.5%**	3.8	96.1	96.1
KNN	all	5.6	94.6	94.3
LogitBoost	all	4.4	95.4	95.5
NaiveBayes	all	4.4	96.1	95.5
RandomForest	all	**3.07**	**96.9**	**96.8**
LASVM	42.2%	3.53	96.2	96.4
MMA dataset				
A2ING	**39.7%**	23.8	79.0	76.1
KNN	all	27.2	76	72.8
LogitBoost	all	27.4	73.1	72.5
NaiveBayes	all	24.4	76.5	75.6
RandomForest	all	25.7	76	74.3
LASVM	43.5%	**22.5**	**79.1**	**77.4**
APP dataset)				
A2ING	**26.4%**	**14.6**	85.3	**85.3**
KNN	all	15.8	84.4	84.2
LogitBoost	all	18.9	81.6	81.0
NaiveBayes	all	22.8	81.1	77.1
RandomForest	all	16.0	84.3	83.9
LASVM	30.2%	15.22	**85.7**	84.8
Average results over all datasets				
A2ING	**36.3%**	**18.02**	**83.4**	**81.88**
KNN	all	22.26	81.14	78.78
LogitBoost	all	21.58	80.18	78.34
NaiveBayes	all	21.12	81.22	78.8
RandomForest	all	19.69	81.25	80.24
LASVM	46.77%	18.23	83.24	81.7

Fig. 5 (left side) shows for each dataset, the obtained accuracy, according to the human labor which is expressed as the number of documents that are labeled for learning. Fig. 5 (right side) shows the corresponding number of labeled documents, according to the number of documents seen from the stream. A2ING is compared with LASVM. The Passive case on Fig. 5 represents the case where each document from the stream is manually labeled and used for learning, even if it is not informative. For all the datasets, we can see that A2ING by querying only labels the most informative documents, achieves a better or equal accuracy to LASVM, and always achieves a better accuracy than the passive mode which spend time and effort for labeling all the documents, regardless of their informativeness.

Fig. 6 (1) shows the effect of the confidence parameter δ, on the number of queried document labels. Naturally, the more δ is close to 0 the smaller is the number of labeled documents, because in this case A2ING will select only the top most uncertain (informative) documents.

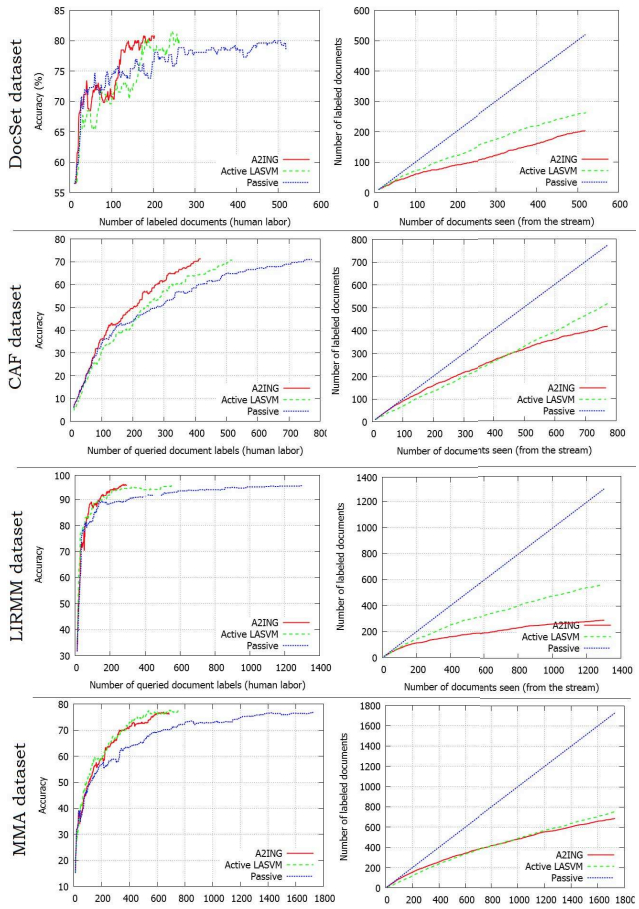

Fig. 5. Left: the obtained accuracy according to the number of labeled documents. Right: The number of labeled documents according to number of documents seen from the stream

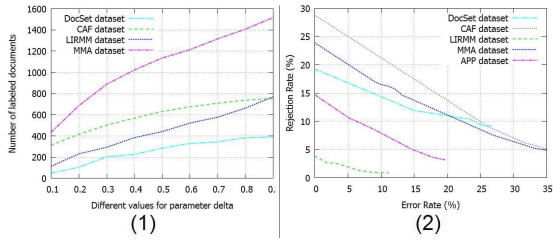

Fig. 6. (1) The final number of queried document labels with different δ values. (2) Error rate optimization, by rejecting uncertain test documents

The uncertainty measure defined in section III-A to determine the informativeness of a document for learning, may also be used as a measure to decide whether a document should be rejected or not during testing. Fig. 6 (2) shows how the error rate can decreases by rejecting uncertain documents (having $\Delta_{(c_1,c_2|x)} < \delta$), where the rejection rates are determined by testing with variable values of the parameter δ, starting from 0 (no rejection) and scaling by $+0.05$ each time. Rejecting a document may have a cost, however, in an industrial context, we may prefer to reject uncertain documents, because we give more importance to minimizing the number of errors than to maximizing the number of recognized documents, since doing an error is more costly. Note that in a real-world industrial problem, we estimate that the cost of making a classification error is two times higher than the cost of wrongly rejecting a document.

V. Conclusion and Future Work

Based on AING, this paper presented a learning approach for document classification task. The proposed method is suitable for an industrial setting where documents become available progressively over time, while their labels are not available. Indeed, it can (1) learn online from a continuously arriving stream of documents, and (2) only query during learning the labels of the most informative documents, thus saving annotation time and effort. The method inherits from AING the non-sensitivity to initialization parameters and does not require the number of classes or neurons to be known. Beside the fact that the proposed method is incremental and that it considerably reduces the required number of labelled documents, the experimental results on real document datasets show that it is efficient compared to active and even fully supervised classifiers.

Nonetheless, further work still needs to be done. It may be more costly for a human to label a given document, when it is not clear at first sight from which class the document is. Indeed, depending on the document type and quality, there may be a variable labelling cost for different documents. Thus, beside reducing the number of the manually labelled documents, it may be interesting to consider a cost of labelling different documents, e.g., a function of the average time required to label a given type of documents. Another direction is to deal with the class-imbalance problem and to provide some theoretical bound on the number of queried labels that are required by A2ING to achieve a given level of accuracy.

References

[1] N. Chen, D. Blostein, *A survey of document image classification: problem statement, classifier architecture and performance evaluation*. IJDAR, pp 1-16, 2007

[2] X. Zhu, *Semi-supervised learning literature survey*. Computer Sciences Technical Report 1530, University of Wisconsin-Madison, 2008.

[3] M. Zaki, H. Yin, *Semi-supervised Growing Neural Gas for Face Recognition*. JMMA, pp 425-435, 2008

[4] S. Ertekin, J. Huang, L. Bottou, C.L. Giles. *Learning on the Border: Active Learning in Imbalanced Data Classification*. CIKM, pp 127-136, 2007

[5] Y. Fu, X. Zhu, B. Li. *A survey on instance selection for active learning*. KAIS, pp 1-35, 2012

[6] CC. Aggarwal, CX. Zhai, *A survey of text clustering algorithms*. Mining Text Data. Springer US, pp 77-128, 2012

[7] M-R. Bouguelia, Y. Belaid and A. Belaid, *An adaptive incremental clustering method based on the growing neural gas algorithm*. ICPRAM, pp 1-8, 2013

[8] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, *Fast kernel classifiers with online and active learning*. JMLR, pp 1579-1619, 2005

[9] L. Jiang, Z. Cai, D. Wang, and S. Jiang, *Survey of improving k-nearest-neighbor for classification*. FSKD, pp 679-683, 2007.

[10] J. Friedman, T. Hastie, and R. Tibshirani, *Additive Logistic Regression: a Statistical View of Boosting*. Annals of Statistics, pp 337-407, 1998

[11] D. Lowd, P. Domingos: *Naive Bayes models for probability estimation*. ICML, pp 529-536, 2005

[12] L. Breiman, *Random Forests*. Machine Learning, pp 5-32, 2001