Document Classification and Page Stream Segmentation for Digital Mailroom Applications

Albert Gordo, Marçal Rusiñol, Dimosthenis Karatzas and
Andrew D. Bagdanov
Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Univ. Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain.

***Abstract*—In this paper we present a method for the segmentation of continuous page streams into multipage documents and the simultaneous classification of the resulting documents. We first present an approach to combine the multiple pages of a document into a single feature vector that represents the whole document. Despite its simplicity and low computational cost, the proposed representation yields results comparable to more complex methods in multipage document classification tasks. We then exploit this representation in the context of page stream segmentation. The most plausible segmentation of a page stream into a sequence of multipage documents is obtained by optimizing a statistical model that represents the probability of each segmented multipage document belonging to a particular class. Experimental results are reported on a large sample of real administrative multipage documents.**

I. Introduction

Batch scanning of multiple documents is common in many application contexts, the most well known and commercially exploited being digital mailroom systems. One of the key steps in the batch scanning process is the segmentation of the resulting page stream into continuous sets of pages corresponding to the physical documents, a procedure also referred to as document separation.

In digital mailroom applications, page stream segmentation is typically achieved by manually introducing separator pages or machine readable marks in the stream during scanning [1]. An alternative, that also requires manual intervention, is based on capturing images of the front pages during the preparation of the batch, which are then matched to the scanned pages to establish the separation points [2]. It is important to note that such implementations do not perform any kind of document classification, which is instead treated as a separate problem at a later time.

The process of sorting and inserting separator sheets is costly and error-prone, estimated to represent about 50% of the cost of document preparation [3]. Achieving document separation and classification without any type of human intervention is thus of significant commercial interest. Nevertheless, this problem has received relatively little attention from the research community. Very few automatic solutions have been proposed, and we are unaware of any use of such methods in commercial systems.

In the context of this paper *page stream segmentation* refers to the combined problem of both finding document separation points in an ordered collection of page images and assigning the correct semantic labels to the output documents. The page stream does not contain any separator pages or other marks, while the documents in the page stream comprise sets of pages that do not necessarily bear any similarity between each other. Only document-level labels are available during training, while there is no prior information about the number of documents in the stream.

In this paper we present a supervised approach to page stream segmentation and document classification. We make use of a multipage document representation. After demonstrating the effectiveness of the multipage representation through a multipage document classification task, we make use of it to construct an estimator for the validity of document hypotheses, that integrates information about the distribution of document lengths for each class. Finally, we put everything together within a probabilistic framework to derive the optimal segmentation points in the page stream and assign semantic labels to the output documents. We evaluate our approach on a stream of administrative documents obtained through a real-life digital mailroom application used in the banking sector. We analyze different variants of the approach and demonstrate high performance on the selected dataset.

II. Related Work

The most direct approach to the segmentation of continuous data streams is to pose it as an unsupervised segmentation problem where the objective is to establish transition points based on discontinuities in certain features, or the equivalent problem of clustering sequences of input samples based on a similarity metric. Along this line of thought, maximum entropy methods have been successfully used for the segmentation of natural language sentences [4], while in the domain of video segmentation shot change detection [5] is typically achieved based on either global frame similarity metrics [6] or local keypoint matching [7].

In the domain of page stream segmentation, Collins-Thompson and Nickolov [8] treat document separation as a constrained bottom-up clustering problem. Page similarity is assessed in various ways, including structural similarity and cross-correlation of recognized text parts. Bottom-up clustering is then performed, beginning with each page in its own cluster and progressively merging pairs of clusters using a single-linkage criterion. The method relies strongly on a good OCR performance and the correct localization of certain layout features such as page number boxes.

Approaches such as the above do not address the problem of document classification and make no use of any prior knowledge about the expected document types for page stream segmentation. Instead, they implicitly assume intra-cluster homogeneity and inter-cluster separability which does not necessarily hold in the page stream segmentation scenario: it is quite frequent for documents to comprise pages that bear little similarity to each other and for documents of different types to include similar pages. A more generic approach that also addresses the document classification problem is to adopt a Markov chain formulation and pose page stream segmentation as a sequence mapping problem in which the input sequence of pages must be mapped to an output sequence of document types.

Schmidtler and Amtrup [3] employ such a Markov chain formulation based on a bag of words description for individual pages. Noting that the first and last pages of documents are quite different than those in the middle, they model each document type as a sequence of three symbols, corresponding to first, middle, and end pages. A number of binary SVM classifiers are then trained and their scores mapped to probabilities using Platt's scaling [9]. Finally, the best sequence of document types is established through Viterbi search. In a similar fashion Meilander and Belaid [10] propose an approach based on multi-gram models and an adaptation of the forward-backward algorithm to obtain the best segmentation.

Alternatively, explicit modeling of multi-page document classes can be used along with a probabilistic framework to classify sets of incoming pages. This is the approach we follow in this paper. The main advantage of such an approach over a Markovian formulation is that there is no requirement that pages in individual documents be well-ordered. This is quite important in many real-world settings, especially when dealing with administrative documents. A frequent case is document types for which by their very nature the order of pages is non-informative, such as for example "Balances" which are unordered collations of data and analyses. Moreover, it is quite common that semantic labels available for training actually refer to container documents that comprise various sub-documents, for example a "Proof of Ownership" document which might contain anything from a contract to assorted invoices.

To our knowledge, there is no related prior art on using multi-page document representations for page stream segmentation. As a matter of fact, there is very limited work on multi-page representations per se, with the vast majority of approaches focusing on single-page document descriptors. The authors studied the use of multi-page representations both for classification [11] and for retrieval [12] in the past. In [11] multi-page representations are achieved through a bag-of-pages approach based on a codebook of page-level labels learned through an unsupervised clustering process. This representation is further improved using the Fisher vector framework [13]. In [12] a number of different multi-page document retrieval approaches are studied, including the early fusion of single-page descriptors to multi-page document representations as well as late fusion strategies based on the combination of single-page based retrieval results.

III. Multi-page Document Representation and Segmentation

In this section we describe our approach to document stream segmentation. We begin by introducing a model for multi-page document classification, then in section III-B we describe how we take advantage of the classification method to assess the validity of multipage document hypotheses. In section III-C we put everything together to perform document stream segmentation.

A. Multi-page Document Representation

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ be a set of N pages that represents a multi-page document. In this work we assume that pages are represented as d-dimensional feature vectors, *i.e.* $p_i \in \mathbb{R}^d$, though non-vectorial representations such as graph-based descriptors could be easily integrated using KPCA [14], for example. In general, different documents will contain a different number of pages, even if they belong to the same class.

As mentioned above, document models that explicitly encode page order can be inefficient and unsuitable for many applications. Because of this, a desirable goal is to devise a feature-vector representation of the document *as a whole* whose dimensionality does not depend on the number of pages, *i.e.*, we are interested in finding a transformation ϕ such as that $\phi(\mathcal{P}) \in \mathbb{R}^D$, and where the dimensionality D does not depend on the number of pages N. This will enable us to use efficient methods for classification, retrieval, clustering, segmentation, etc.

Here we make use of one of the methodologies evaluated in previous work of the authors [12] to encode sets of pages into a compact document representation. The main idea consists of performing average pooling of page representations to obtain a single representation of an entire document:

$$\phi(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} p \tag{1}$$

This representation is then L2-normalized to make signatures with different number of documents fully comparable. When representing individual pages with representations similar to bag-of-words, aggregating the pages as we propose can be related to computing the histogram of a complete document that has been "concatenated" into a single page, while the averaging and L2 normalization take into account the "length" of the document. Despite its crudeness, in the experimental evaluation in Section IV we will show how this representation can equal or outperform the performance of more complex representations while being much simpler from an engineering point of view.

B. Document Validation

A different but related problem to multi-page document classification is document validation. Given a set of pages, the goal is to asses whether or not they form a valid document of one of K possible classes. For example, if we take a valid document and strip some of its pages, the remaining pages will not form a valid document. Similarly, if we merge pages from different documents, the results will also not be a valid

document. For tasks such as document stream segmentation, it is extremely important to have an estimator of the validity of a document hypothesis. We can define such an estimator as a function V that receives a set of pages $\mathcal{P}$ and a function ϕ that transforms the set of pages into a document representation, and produces a high or low score depending on how likely $\phi(\mathcal{P})$ is a valid document.

A naive approach to construct V is to sample "fake", invalid documents, and train a valid / invalid classifier. Then we can use the output score of the classifier as our validity measure. Unfortunately, as we will see experimentally in Section IV, this does not produce good results. We believe the reason for these low results in the naive approach is twofold. First, it does not take into account in any way the different classes of the documents. We believe leveraging this information is crucial, since separating all the valid documents from the invalid documents with only one hyperplane on a low dimensional space may just not be possible. Second, we believe that the number of pages of a document is quite relevant to decide its validity. However, our ϕ representation – as well as the bag of pages approach of [11] – does not contain explicit information about its number of pages, and any implicit information is probably lost after L2 normalization. Explicitly adding the number of pages as an extra feature in the document representation is an ad hoc solution which leads only to a small improvement in the results.

We instead propose a probabilistic approach that takes into account the distribution of document lengths for each class, not as a feature, but as part of the probabilistic model. We believe such a formulation is more principled than appending the number of pages as a feature. This formulation also considers information of each class learned independently, leading to a more expressive model.

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ represent the set of possible document classes. Intuitively, we assume the existence of an underlying generative model for each of these K document types. Based on this, we define $V(\mathcal{P}, \phi)$ as how likely it is for $\phi(\mathcal{P})$, of length $|\mathcal{P}|$, to be generated by *any* of those models, independently of how likely those classes are. For the sake of compactness, let us define the document $x = \phi(\mathcal{P})$ and the document length $n = |\mathcal{P}|$. Then, according to this definition,

$$V(\mathcal{P}, \phi) = \sum_{c \in \mathcal{C}} p(x, n|c). \tag{2}$$

Note however that this formulation encourages documents where all classes produce a high score. This usually implies that the document is very ambiguous, and may in fact be an invalid document. Instead, we propose to maximize the difference between the first and second most likely classes, similar to the *margin sampling* technique typically used in active learning:

$$V(\mathcal{P}, \phi) = \max_{\hat{c} \in \mathcal{C}} p(x, n|\hat{c}) - \max_{c \in \mathcal{C} \setminus \{\hat{c}\}} p(x, n|c) \tag{3}$$

To model $p(x, n|c)$ we make the assumption that x and n are conditionally independent given the class c. This is reasonable since, as mentioned before, after L2 normalization x no longer encodes much information about the number of pages it contains. After assuming conditional independence of x and n:

$$p(x, n|c) = p(x|c)p(n|c). \tag{4}$$

Applying Bayes to $p(x|c)$ and assuming $p(x)$ to be uniform (a typical assumption):

$$p(x, n|c) \propto \frac{p(c|x)p(n|c)}{p(c)}. \tag{5}$$

Here, $p(c|x)$ is the probability of classifying x into class c. This can be obtained, for example, after learning a classifier using documents of c as positive samples and documents of other classes – as well as fake documents – as negative samples. We represent with $p(c)$ the probability of class c, which can be obtained counting the proportion of training documents belonging to that class. Finally, $p(n|c)$ is the probability of a document of class c to contain n pages. This can be learned on the training set using Kernel Density Estimation techniques.

We note that we do not calibrate the scores of $p(c|x)$, since we observed a huge degradation of the results by doing so. Platts scaling is good when scores follow a Gaussian-like distribution, but we observed that our scores did not follow such distribution. Using more general methods such as a Weibull fitting [15] could help, but we consider that out of the scope of this work.

C. Page Stream Segmentation

Let us denote with $\mathcal{S} = \{p_1, p_2, \ldots, p_N\}$ a stream of N pages which we want to split into an (unkwown) number of multi-page documents, and where each segmented document belongs to one of K possible classes. For $i \leq j$ we define $\mathcal{S}_{i:j} = \{p_i, \ldots, p_j\}$ to be the subsequence of pages from $\mathcal{S}$ from page i to page j. In this way $V(\mathcal{S}_{i:j}, \phi)$ is the validity score of the L2-normalized document representation constructed with ϕ using pages p_i to p_j. Let P_j denote the score of the best possible segmentation up to page p_j of the page stream. P_j is recursively defined as:

$$P_j = \begin{cases} 0 & \text{if } j = 0, \\ \max\limits_{i<j}(P_i + \log V(\mathcal{S}_{i+1:j}, \phi)) & \text{otherwise.} \end{cases} \tag{6}$$

To obtain the score of the best possible segmentation we simply need to calculate P_n. Although this only produces a score, it is straightforward to keep track of the paths chosen and to produce the optimal segmentation boundaries. Note that we use the sum-log instead of the product to avoid underflows. The above equation can be solved with dynamic programming techniques in $\mathcal{O}(n^2)$, or in $\mathcal{O}(mn)$ if we limit the maximum length of candidate documents to m pages.

To solve Equation (6) in a reasonable time it it is important to be able to generate the $\phi(\mathcal{S}_{i:j})$ signatures efficiently independently of the number of pages they contain. We propose to use an "integral" pages representation to achieve this goal. Given a stream of pages, we first generate aggregated pages, such as that $I_i = \sum_{j=1}^{i} p_j$ and $I_0 = \mathbf{0}$. Then $\phi(\mathcal{S}_{i:j})$ can be rapidly computed as $\phi(\mathcal{S}_{i:j}) = (I_j - I_{i-1})/(j - i + 1)$ and then L2 normalized with only one subtraction between vectors independently of the number of pages.

IV. Experiments

Datasets and features. We are not aware of any public, multipage document dataset on which we can evaluate our classification and segmentation methods. Therefore, all our

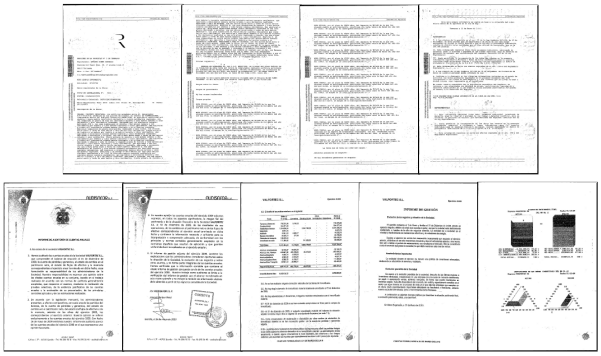

Fig. 1. Example of two multipage documents from our mailroom stream.

TABLE I. CLASSIFICATION RESULTS

	Proposed	FV 2G	FV 4G	FV 8G
Text	96.23	96.20	96.50	95.70
Visual	86.68	87.52	88.23	88.20
Text + Visual	97.50	96.66	95.73	96.04

experiments have been carried out on an in-house dataset. Our dataset consists of $7,203$ incoming document images composed of nearly $70,000$ pages sampled from a real-world banking workflow. The sample corresponds to two months of incoming documents. The dataset contains 13 different document semantic classes (invoices, tax forms, contracts, property registers, etc.) which have been manually labeled. We remark that these labels are available at the document level, but not at the page level. Fig. 1 contains examples of multipage documents from our dataset.

We have split the collection into training and test sets, each corresponding to one month of documents. The training set consists of $3,967$ documents ($38,313$ pages) and the test set $3,236$ documents ($31,424$ pages). We explore both visual and textual representations of individual pages. For textual features, all page images have been OCRed, stopwords have been removed and a stemming algorithm has been applied. After weighting word importance with a tf-idf model, pages are represented by a histogram of 300 dimensions that encodes a mixture of topics extracted through Latent Semantic Analysis [16]. The visual representation of pages we use is a histogram accumulating multi-directional runlengths at different scales [17]. The runlength histograms are reduced to 300 dimensions with Principal Component Analysis.

Classification results. In our first set of experiments we explore how our multipage document representation from Section III-A compares to the state-of-the-art method from [11]. Single pages are represented using text features, visual features, and both. We train one-versus-rest SVM classifiers using LIBLINEAR [18]. The cost parameter of the classifiers is set to the default value of 1. Small improvements can be obtained by crossvalidating this parameter, but we observed that the default value performs well in practice. When comparing to the Fisher vector approach of [11], we use vocabularies of 2, 4, and 8 Gaussians and compute gradients with respect to both the means and the variances.

Results for document classification are given in Table I. Our proposed approach obtains results comparable to the more complex Fisher vector approach, and in some cases actually outperforms Fisher vectors. For the subsequent experiments on document validation and segmentation we will use the proposed method only. From Table I we can also observe that textual features are much more discriminative than visual ones, which is in line with the findings of other authors [12]. Combining textual and visual features yields a small improvement in some cases, at the cost of a larger signature.

TABLE II. DISCRIMINATION OF VALID / INVALID DOCUMENTS

	Direct	Direct + #Pages	Prob.
mAP (Text)	33.58	43.02	71.81
mAP (Text + Visual)	34.87	41.99	70.59

Validity results. Here we evaluate the ability of different validity models to discriminate between real (or valid) and fake (or invalid) documents. Ideally, a good validity function would rank the valid documents higher than the invalid ones. To test this, we generate approximately $40,000$ invalid documents from the test set by sampling a random starting page and a length and then generating the corresponding document. The length of the document is drawn from a uniform distribution over the interval $[1, 150]$. There were no real documents in the training set longer than 150 pages. We combine these invalid documents with all the valid test documents and rank them with different validity functions. Since we are interested in the quality of the ranking, we report mean average precision.

We consider three different approaches of computing a document validity score:

- Direct, where we learn one valid / invalid SVM classifier and use the raw classification score as a validity measure. We sample invalid documents from the training set using the procedure described above to use as negative samples during learning.
- Direct + #Pages. Same as before, but with the number of pages of the document appended as an extra feature in the document descriptor.
- Probabilistic, as described in Section III-B.

Results for these document validity measures are given in Table II. The proposed probabilistic method clearly outperforms the naive approach even when adding the number of pages as a feature. We also note that visual features add very little information, or even degrade the results for this task.

Segmentation results. Finally we report results for document stream segmentation. We use the probabilistic formulation of Equation (3) and use only textual descriptors, since we observed that visual descriptors degrade the probabilistic validity results. We report several evaluation metrics:

- Precision: the proportion of automatically segmented documents that are correct according to the ground truth (both in boundary detection and in category).
- Recall: the proportion of the ground truth documents that have been correctly segmented and labeled.
- mAP: we rank all automatically segmented documents according to their classification score and report mean average precision.

TABLE III. SEGMENTATION RESULTS

	Precision	Recall	mAP	Normalized Ha.
Prob.	4.17	16.35	2.14	6.63
Prob. + Merge	22.39	40.24	22.35	1.91

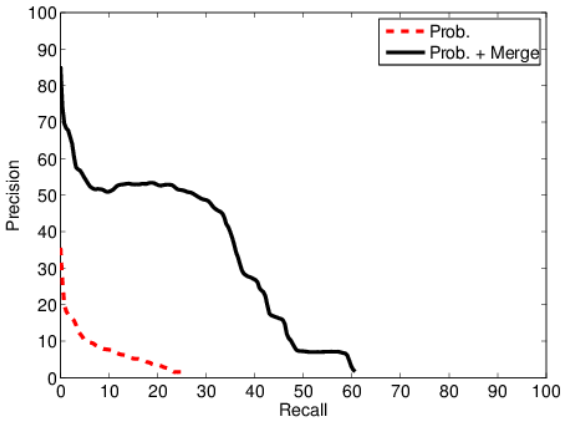

Fig. 2. Precision-recall plot for automatically segmented documents.

- Normalized Hamming distance: the number of incorrectly placed boundaries between documents, divided by the number of documents in the ground truth. This metric measures the amount of work needed to manually correct an automatic segmentation.

We report our segmentation results in the first row of Table III. Note the low precision compared to recall, together with very high normalized Hamming distances. This immediately suggests that the method is oversegmenting. We propose a simple post-processing step to reduce this effect: consecutive segmented documents assigned to the same class are merged and rescored. Although this may incorrectly merge valid documents, we found a very significant improvement in the results, shown in the second row of Table III. Note however that the precision is still low with respect to recall, and that the normalized Hamming distance is still high, suggesting that oversegmentation is still happening. Therefore, techniques that avoid oversegmentation should lead to improved results.

Finally, in Figure 2 we show a precision-recall curve of the probabilistic approach, with and without merging of consecutive documents. We observe that merging significantly improve both precision and recall at no extra cost.

V. CONCLUSIONS

In this paper we described a method of representing multiple-page documents and we applied it to the problem of page flow segmentation. We proposed a probabilistic model that is used to model document validity, and is then in turn used to drive a page flow segmentation algorithm based on dynamic programming. Our technique performs well for document classification and validity scoring, and the preliminary results on document stream segmentation are encouraging. The approach is biased towards oversegmenting documents, but we believe that techniques such as hard negative mining could be used to deal with this problem and to improve our models at no extra cost at test time.

ACKNOWLEDGMENTS

This work was supported by the EU FP7 Project ADAO (IAPP-2008-230653)) and the Spanish research projects TIN2011-24631, RYC-2009-05031.

REFERENCES

[1] F. Ragnet, J. Moore, N. Raphael Saubat, E. Cheminot, and T. Lehoux, "Method for one-step document categorization and separation," US Patent 2011/0 192 894A1, 2011.

[2] J. Moore, F. Ragnet, E. Cheminot, and Y. Hoppenot, "Document separation by document sequence reconstruction based on information capture," US Patent 2012/0 128 540A1, 2012.

[3] M. Schmidtler and J. Amtrup, "Automatic document separation: A combination of probabilistic classification and finite-state sequence modeling," in *Natural Language Processing and Text Mining*, 2006, pp. 123–144.

[4] J. C. Reynar and A. Ratnaparkhi, "A maximum entropy approach to identifying sentence boundaries," in *Conference on Applied natural language processing*, 1997, pp. 16–19.

[5] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 411 – 418, 2010.

[6] R. Joyce and B. Liu, "Temporal segmentation of video using frame and histogram space," *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 130–140, 2006.

[7] C.-R. Huang, H.-P. Lee, and C.-S. Chen, "Shot change detection via local keypoint matching," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1097–1108, 2008.

[8] K. Collins-Thompson and R. Nickolov, "A clustering-based algorithm for automatic document separation," in *SIGIR 2002 Workshop on Information Retrieval and OCR: From Converting Content to Grasping Meaning*, 2002.

[9] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[10] T. Meilender and A. Belad, "Segmentation of continuous document flow by a modified backward-forward algorithm," in *Document Recognition and Retrieval*, vol. 7247, 2009, pp. 724 705–724 705–10.

[11] A. Gordo and F. Perronnin, "A bag-of-pages approach to unordered multi-page document classification," in *International Conference on Pattern Recognition*, 2010, pp. 1920–1923.

[12] M. Rusiñol, D. Karatzas, A. D. Bagdanov, and J. Llados, "Multipage document retrieval by textual and visual representations," in *International Conference on Pattern Recognition*, 2012, pp. 521–524.

[13] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *European Conference on Computer Vision*, 2010.

[14] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, 1998.

[15] W. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult, "Multi-attribute spaces: Calibration for attribute fusion and similarity search," in *Computer Vision and Pattern Recognition*, 2012.

[16] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[17] A. Gordo, F. Perronnin, and E. Valveny, "Large-scale document image retrieval and classification with runlength histograms and binary embeddings," *Pattern Recognition*, 2012.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.