Weakly supervised discriminative training of linear models for Natural Language Processing

Lina Maria Rojas Barahona, Christophe Cerisara

Weakly supervised discriminative training of linear models for Natural Language Processing

Lina M. Rojas-Barahona[1] and Christophe Cerisara[2]

[1] Université de Lorraine/LORIA, Nancy
[2] CNRS/LORIA, Nancy
{lina.rojas,christophe.cerisara}@loria.fr

Abstract. This work explores weakly supervised training of discriminative linear classifiers. Such features-rich classifiers have been widely adopted by the Natural Language processing (NLP) community because of their powerful modeling capacity and their support for correlated features, which allow separating the expert task of designing features from the core learning method. However, unsupervised training of discriminative models is more challenging than with generative models. We adapt a recently proposed approximation of the classifier risk and derive a closed-form solution that greatly speeds-up its convergence time. This method is appealing because it provably converges towards the minimum risk without any labeled corpus, thanks to only two reasonable assumptions about the rank of class marginal and Gaussianity of class-conditional linear scores. We also show that the method is a viable, interesting alternative to achieve weakly supervised training of linear classifiers in two NLP tasks: predicate and entity recognition.

1 Introduction

Unsupervised training of discriminative models poses serious theoretical issues, which prevent such models from being widely adopted in tasks where annotated corpora do not exist. In such cases, generative models are thus often preferred. Nevertheless, discriminative models have various advantages that might be desirable even in these cases, for example their very interesting capacity to handle correlated features and to be commonly equipped with many rich features. Hence, many efforts have been deployed to address this issue, and some unsupervised training algorithms for discriminative models have been proposed in the Natural Language Processing (NLP) community, for instance Unsearn [3], Generalized Expectation [4] or Contrastive Training [15] amongst others.

Our approach [3] relies on a novel approximation of the risk of binary linear classifiers proposed in [1]. This approximation relies on only two assumptions: the rank of class marginal is assumed to be known, and the class-conditional linear scores are assumed to follow a Gaussian distribution. Compared to previous applications of unsupervised discriminative training methods to NLP tasks, this approach presents several advantages: first, it is proven to converge towards the true optimal classifier risk; second, it does not

[3] This work has been partly funded by the ANR ContNomina project

require any constraint; third, it exploits a new type of knowledge about class marginal that may help convergence towards a relevant solution for the target task. But the original approach in [1] has a very high computational complexity, and we investigate in this work two options to reduce this issue: we first derive a closed-form expression of the objective function, which allows a much faster algorithmic implementation. We also propose to pre-train the classifier on a very small amount of data to speed-up convergence. Finally, we validate the proposed approach on two new binary NLP tasks: predicate identification and entity recognition.

2 Classifier risk approximation

We first briefly review the approximation of the risk proposed in [1]. A binary (with two classes: 0 and 1) linear classifier associates a score $f_{\theta^0}(X)$ to the first class 0 for any input $X = (X_1, \cdots, X_{N_f})$ composed of N_f features X_i:

$$f_{\theta^0}(X) = \sum_{i}^{N_f} \theta_i X_i$$

where the parameter $\theta_i \in \mathbb{R}$ represents the weight of the feature indexed by i for class 0. As it is standard in binary classification, we constrain the scores per class to sum to 0:

$$f_{\theta^1}(X) = -f_{\theta^0}(X)$$

In the following, we may use both notations $f_{\theta^0}(X)$ or $f_\theta(X)$ equivalently. X is classified into class 0 iff $f_{\theta^0}(X) \geq 0$, otherwise X is classified into class 1. The objective of training is to minimize the classifier risk:

$$R(\theta) = E_{p(X,Y)}[\mathcal{L}(Y, f_\theta(X))] \tag{1}$$

where Y is the true label of the observation X, and $\mathcal{L}(Y, f_\theta(X))$ is the loss function, such as the hinge loss used in SVMs, or the log-loss used in CRFs. This risk is often approximated by the empirical risk that is computed on a labeled training corpus. In the absence of labeled corpus, an alternative consists in deriving the true risk as follows:

$$R(\theta) = \sum_{y \in \{0,1\}} P(y) \int_{-\infty}^{+\infty} P(f_\theta(X) = \alpha | y) \mathcal{L}(y, \alpha) d\alpha \tag{2}$$

We use next the following hinge loss:

$$\mathcal{L}(y, \alpha) = (1 + \alpha_{1-y} - \alpha_y)_+ \tag{3}$$

where $(x)_+ = \max(0, x)$, and $\alpha_y = f_{\theta^y}(X)$ is the linear score for the correct class y. Similarly, $\alpha_{1-y} = f_{\theta^{1-y}}(X)$ is the linear score for the wrong class.

Given y and α, the loss value in the integral can be computed easily. Two terms in Equation 2 remain: $P(y)$ and $P(f_\theta(X) = \alpha | y)$. The former is the class marginal and is assumed to be known. The latter is the class-conditional distribution of the linear scores,

which is assumed to be normally distributed. This implies that $P(f_\theta(X))$ is distributed as a mixture of two Gaussians (GMM):

$$P(f_\theta(X)) = \sum_{y \in \{0,1\}} P(y)\mathcal{N}(f_\theta(X); \mu_y, \sigma_y)$$

where $\mathcal{N}(z; \mu, \sigma)$ is the normal probability density function. The parameters $(\mu_0, \sigma_0, \mu_1, \sigma_1)$ can be estimated from an unlabeled corpus $\mathcal{U}$ using a standard Expectation-Maximization (EM) algorithm for GMM training. Once these parameters are known, it is possible to compute the integral in Eq. 2 and thus an estimate $\hat{R}(\theta)$ of the risk without relying on any labeled corpus. The authors of [1] prove that:

- The Gaussian parameters estimated with EM converge towards their true values;
- $\hat{R}(\theta)$ converges towards the true risk $R(\theta)$;
- The estimated optimum converges towards the true optimal parameters, when the size of the unlabeled corpus $\mathcal{U}$ increases infinitely:

$$\lim_{|\mathcal{U}| \to +\infty} \arg\min_\theta \hat{R}(\theta) = \arg\min_\theta R(\theta)$$

They further prove that this is still true even when the class priors $P(y)$ are not known precisely, but only their relative order (rank) is known. These priors must also be different $P(y = 0) \neq P(y = 1)$.

3 Risk minimization algorithm

Given the estimated Gaussian parameters, the authors of [1] use numerical integration to compute Eq. 2. But numerical integration only gives an approximate integral and requires some tuning to balance between accuracy and computation time. We thus propose next a closed-form derivation of the risk that computes the exact integral at a lower cost.

3.1 Closed-form risk estimate

Figure 1 summarizes the main steps of our proposed derivation of the risk. The full derivation is available in annex A. It exploits the following Gaussianity assumptions:

$$P(f_{\theta^i}(X)|Y = j) \sim \mathcal{N}(X; \mu_{j,i}, \sigma_{j,i}) \;\; \forall i, j \in \{0, 1\}$$

The final risk for any binary linear classifier with the hinge loss in Eq-3 is then given by:

$$\begin{aligned}\hat{R}(\theta) = P(Y = 0)\frac{1 - 2\mu_{0,0}}{4\sigma_{0,0}\sqrt{\pi}}\left(1 + \operatorname{erf}\left(\frac{\frac{1}{2} - \mu_{0,0}}{\sigma_{0,0}}\right)\right) + \\ \frac{P(Y = 0)}{2\pi}\exp\left(-\frac{(\frac{1}{2} - \mu_{0,0})^2}{\sigma_{0,0}^2}\right) + \\ P(Y = 1)\frac{1 + 2\mu_{1,0}}{4\sigma_{1,0}\sqrt{\pi}}\left(1 - \operatorname{erf}\left(\frac{-\frac{1}{2} - \mu_{1,0}}{\sigma_{1,0}}\right)\right) +\end{aligned} \tag{4}$$

$$\frac{P(Y=1)}{2\pi}\exp\left(-\frac{(-\frac{1}{2}-\mu_{1,0})^2}{\sigma_{1,0}^2}\right)$$

In the following experiments, we evaluate the gain in computation time and accuracy resulting from the use of Eq. 4 by comparing it with a numerical integration algorithm, which is applied to compute the integrals in step 2 of Figure 1.

Step 1 Given $\alpha_1 = -\alpha_0$, exploit the constraints: $\mu_{0,1} = -\mu_{0,0}, \mu_{1,1} = -\mu_{1,0}, \sigma_{0,1} = \sigma_{0,0}$ to reduce double integrals to a single integral:

$$R(\theta) = P(Y=0)\int_{-\infty}^{+\infty} N(\alpha_0;\mu_{0,0},\sigma_{0,0})N(-\alpha_0;\mu_{0,1},\sigma_{0,1})(1-2\alpha_0)_+ d\alpha_0 + P(Y=1)\int_{-\infty}^{+\infty} N(\alpha_0;\mu_{1,0},\sigma_{1,0})N(-\alpha_0;\mu_{1,1},\sigma_{1,1})(1+2\alpha_0)_+ d\alpha_0$$

Step 2 Change integral boundaries to remove discontinuities:

$$R(\theta) = P(Y=0)\int_{-\infty}^{\frac{1}{2}} N(\alpha_0;\mu_{0,0},\sigma_{0,0})^2(1-2\alpha_0)d\alpha_0 + P(Y=1)\int_{-\frac{1}{2}}^{+\infty} N(\alpha_0;\mu_{1,0},\sigma_{1,0})^2(1+2\alpha_0)d\alpha_0$$

Step 3 Develop, simplify Gaussian squares with $N(x;\mu,\sigma)^2 = \frac{1}{2\sigma\sqrt{\pi}}N(x;\mu,\frac{\sigma}{\sqrt{2}})$

Simplify integrals with $x\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \mu\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) - \sigma^2\frac{\partial\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\partial x}$

$$R(\theta) = P(Y=0)\frac{1-2\mu_{0,0}}{2\sigma_{0,0}\sqrt{\pi}}\int_{-\infty}^{\frac{1}{2}} N(x;\mu_{0,0},\frac{\sigma_{0,0}}{\sqrt{2}})dx + \frac{P(Y=0)}{2\pi}\exp\left(-\frac{(\frac{1}{2}-\mu_{0,0})^2}{\sigma_{0,0}^2}\right) + P(Y=1)\frac{1+2\mu_{1,0}}{2\sigma_{1,0}\sqrt{\pi}}\int_{-\frac{1}{2}}^{+\infty} N(x;\mu_{1,0},\frac{\sigma_{1,0}}{\sqrt{2}})dx + \frac{P(Y=1)}{2\pi}\exp\left(-\frac{(-\frac{1}{2}-\mu_{1,0})^2}{\sigma_{1,0}^2}\right)$$

Step 4 Integrate with the error function to obtain Eq-4

Fig. 1. Main steps of the proposed derivation of the risk, further details are given in Annex A

3.2 Weakly supervised Algorithm

Our weakly supervised training algorithm (Figure 2) implements a coordinate gradient descent with finite difference.

In the training algorithm, only initialization exploits a few (we experimented with $N = 10$ and $N = 20$ annotated sentences) annotated data: the main iterations (from 2 to 11) do not use any supervised label at all. This initialization incurs no significant additional costs and provides a good enough starting point for the optimization algorithm,

Initialize the weights by training the linear classifier on N annotated sentences
for every iteration t **do**
    for every feature index i **do**
        Move temporary the weight $\theta_i(t) = \theta_i(t-1) + \epsilon$
        Compute the linear scores on the corpus $\mathcal{U}$
        Apply EM to train the 4 Gaussian parameters
        Compute the risk $\hat{R}_+(\theta(t))$ with Eq. 4
        Similarly move temporary the weight in the other direction to compute $\hat{R}_-(\theta(t))$
        Compute the gradient with finite difference and update the weights according to this gradient
    end for
end for

Fig. 2. Overview of the training algorithm

which can thus converge in a reasonable amount of time. Our preliminary experiments with random initialization actually did not converge even after several weeks on a single computer, which supports the principle of using a small supervised initialization in real NLP applications.

4 Target tasks

We validate next our proposed weakly-supervised training algorithm on two NLP tasks: predicate identification and entity recognition.

4.1 Task 1: predicate identification

We consider the task of identifying the semantic predicates in a sentence, which constitutes the first stage in any semantic role labeling (SRL) system. We thus adapt the state-of-the-art supervised MATE SRL system [2], which exploits a linear classifier for predicate identification, by training this classifier with our proposed weakly-supervised algorithm.

We use the parallel Europarl corpus, CLASSIC [13], which contains 1000 French sentences from the European parliament that have been manually annotated with semantic roles. The MATE system relies on syntactic features, which are computed as follows:

- **Part of speech (POS) tags** are automatically computed with the Treetagger [14].
- **Dependency trees** are automatically produced by a parser trained on the French Treebank, as described in [13].

The initial weights are obtained after training the linear classifier on 10 manually annotated sentences ((i.e., $N = 10$ in the algorithm presented in Section 3.2). We set the label priors $P(y)$ so that 80% of the words are not predicates. This ratio is only based on our intuition, and is not very accurate: additional experiments suggest that the non-predicate class marginal should actually be larger. However, such approximation errors should not impact too much the performances, at least theoretically.

4.2 Task 2: Entity recognition

The goal of this task is to detect whether any word form in a text refers to *an entity* or not, where an *entity* is defined as a mention of a person name, a place, an organization or a product. We use the ESTER2 corpus [5], which collects broadcast news transcriptions in French that are annotated with named entities. The following features are used to train the Stanford supervised linear classifier of the Stanford NLP toolkit [4]:

- **Character n-grams** with $n = 4$.
- **Capitalization**: the pattern "Chris2useLC", as defined in Stanford NLP, describing lower, upper case and special characters in words [10].
- **POS tags**: the part of speech tag of every word as given by the Treetagger [14].

The part of speech tags as well as capitalization of words are common important features for entity recognition, while character n-grams constitute a smoother (less sparse) alternative to word forms and are also often used in this context. The label priors $P(y)$ are set so that 90% of the words are not entities and only 10% of the words are entities. The initial weights are obtained after training the linear classifier on 20 manually annotated sentences (i.e., $N = 20$ in the algorithm presented in Section 3.2). The same set of features is used both in the supervised initialization and the weakly supervised risk minimization iterations.

5 Results and Discussion

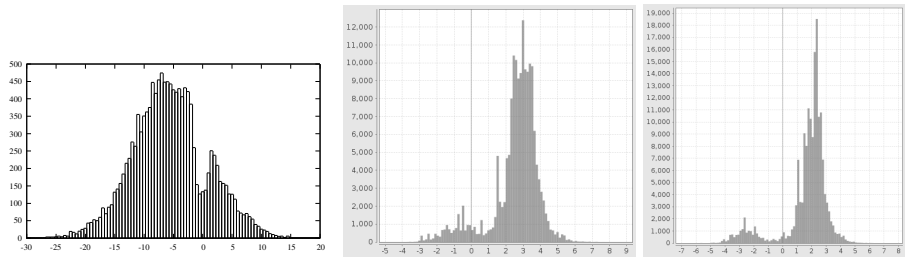
5.1 On the Gaussianity assumption

The proposed approach assumes that the class-conditional linear scores are distributed normally. We invite the interested reader to read [1], where theoretical arguments are given that support the validity of this assumption in various contexts. However, this assumption can not always be taken for granted, and the authors suggest to verify it empirically.

Figure 3(a) thus shows the distribution of $f_\theta(X)$ for the first task with the initial weights on the CLASSIC corpus. We can observe that this distribution can be well approximated by two Gaussians (one for each Y), which confirms the validity of the Gaussianity assumption in this case. For this task, we have also tracked during the whole search the Kurtosis metric as a rough approximate measure of Gaussianity, but we have not observed any strong variation of this measure, which suggests that the distribution of the scores does not vary too much during the search.

Likewise for the second task, the distributions of $f_\theta(X)$ with the initial and final weights (i.e. the weights obtained after training) on the ESTER2 corpus are shown in Figure 3(b) and (c) respectively. These distributions are clearly bi-normal on this corpus, which suggest that this assumption is reasonable in both our NLP tasks.

[4] `http://nlp.stanford.edu/nlp`



(a) (b) (c)

Fig. 3. Distribution of $f_\theta(X)$ on (a) the CLASSIC corpus, using the initial weights trained on 10 sentences; (b) the ESTER2 corpus, using the initial weights trained on 20 sentences; (c) the ESTER2 corpus, using the weights at iteration 6340 of the gradient descent algorithm. The largest mode is on the left in (a), because the *predicate* class is class 0, while the largest mode is on the right in (b) and (c) because the *entity* class is class 1.

5.2 Study of the risk estimator

We now study the risk estimate for oracle parameters that have been trained with an increasing number of supervised labels. The objective is to check whether decreasing Eq. 4 may lead to parameters that are close to the supervised ones. This is an important question for every unsupervised model, where the resulting solution is optimal only with respect to the chosen features and given constraints. Because of their "implicit" definition, the resulting clusters may indeed differ from the expected ones.

Figure 4 shows the risk computed with Eq. 4 on parameters trained in a supervised way on a larger and larger labeled corpus, both with MATE for task 1 and the Stanford software for task 2.

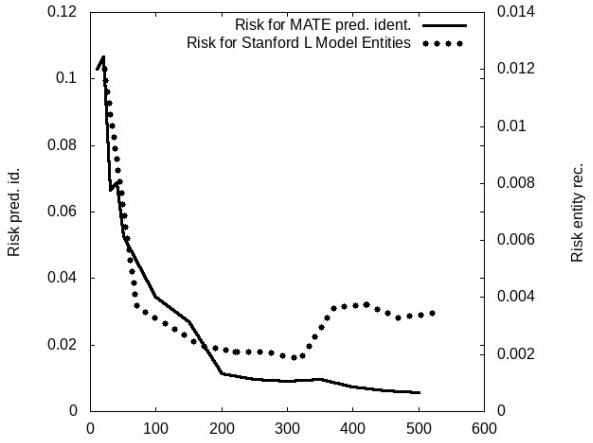


Fig. 4. $\hat{R}(\theta)$ for both supervised classifiers (MATE SRL and Stanford linear) as a function of the number of annotated sentences in the training corpus

We can observe that the risk estimate globally decreases when the linear classifier is trained on more annotated data, which confirms that the proposed risk correlates nicely with the expected clusters given the chosen features. However, the minimum risk on the curve of the second task is not located at the rightmost x-value, but rather at about 300 sentences. This suggests that, on the second task, the risk optimization algorithm will fail to return the best performing parameters. This limitation may be addressed by choosing input features that are better correlated to the target clustering into entities.

5.3 Experiments with gradient descent

We now apply the optimization algorithm described in Figure 2.

Task 1: predicate identification For task 1, the risk is estimated only with the closed-form Eq. 4. We start from initial weights trained on 10 sentences and then apply the gradient descent algorithm for 10,000 iterations. The results obtained are shown in Table 1 (top half).

Task 1			
System	F1	precision	recall
MATE trained on 10 sent.	64.8%	72.1%	58.9%
MATE trained on 500 sent.	87.2%	92.0%	82.9%
Weakly supervised	**73.1**%	63.1%	**87.1**%
Task 2			
System	F1	precision	recall
Stanford trained on 20 sent.	77.4%	89.8%	68%
Stanford trained on 520 sent.	87.5%	90.3%	84.7%
Graph-based Semi.Sup.(MAD)	64.7%	2.37%	4.57%
Weakly sup. closed-form risk	**83.5**%	88.9%	**78.7**%
Weakly sup. numerical integration	**83.6**%	88.7%	**79**%

Table 1. Performances of the proposed weakly supervised system in both tasks.

We can observe that, after the unsupervised optimization iterations, the F1-measure of predicate identification increases, but does not reach the F1 of the supervised model that is trained on 500 sentences. This is due to the existence of weights that give a lower risk than the supervised weights. These results may thus be improved by considering better features or including other constraints during the search.

An analysis of the resulting sentences shows that the inferred classifier has learnt a linguistic property that was not captured by the initial models trained on 10 sentences: verbs without subject can also be predicate. Hence, Figure 5 shows an example of a sentence in the corpus with 2 predicates: the verbs *suffisent* (are enough) and *faire* (to do). The initial MATE model only identified the main verb, but missed its complement because it does not have any explicit subject.

We have noted that this is a typical mistake realized by the initial model, which results from the fact that the model has only been trained on 10 sentences without any occurrence of a similar pattern. But after unsupervised training, the same model is able to capture also the predicate without subject. This results from a better clustering of similar patterns when it is computed in an unsupervised way on a larger corpus.

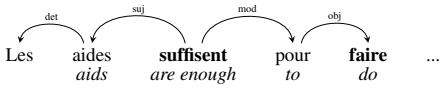


Fig. 5. Example of sentence with 2 predicates that are identified by the weakly-supervised model but not by the initial model

Task 2: Entity Recognition For task 2, experiments are realized both with the closed-form risk estimation in Eq. 4 and numerical integration. For numerical integration, we have made preliminary experiments with both the trapezoidal and Monte Carlo methods [7], and have chosen the former because it was more efficient in our experimental setup. The final performance figures are shown in Table 1 (bottom part), while Figure 6 shows the convergence of optimization with the closed-form risk estimate. According to our experiments, both the closed-form risk and numerical integration reach the same performances (the differences shown in Table 1 are not statistically significant): after $2,000$ iterations, the F1-measure is 83.5%. Therefore, when evaluated on a test set of $167,249$ words and $10,693$ sentences, both methods outperform the supervised linear classifier trained on 20 sentences.

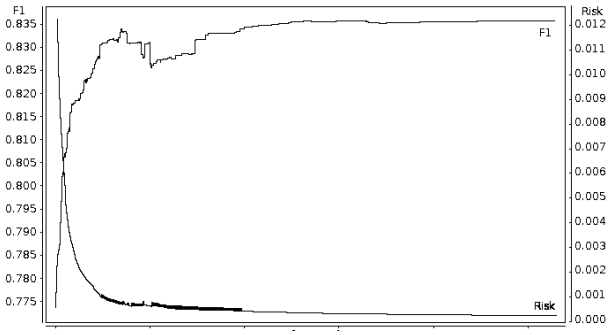


Fig. 6. F1 and $\hat{R}(\theta)$ (from Eq. 4) for entity detection, in function of the number of iterations (step 2 of Figure 2), up to 6340 outer iterations.

-	Baseline (Sup. on 20 sents)		Proposed Model	
Word	Class	Prob.	Class	Prob.
Fabrice	Entity	0.94	Entity	**0.99**
Drouelle	NO	0.53	**Entity**	**0.79**
Floch-Prigent	NO	0.58	**Entity**	**0.69**
Iran	Entity	0.66	Entity	**0.82**
F16	NO	0.73	**Entity**	**0.91**

Table 2. Excerpt of examples correctly classified by the weakly supervised approach for entity recognition, improving the baseline (i.e. the Stanford linear classifier trained on 20 sentences). The last column shows the output probability of the winning class.

In general the proposed model is prone to detect person names that are undetected by the baseline (i.e., the Stanford linear classifier trained on 20 sentences). Table 2 shows two examples of family names (e.g., Drouelle and Floch-Prigent) that are correctly recognized by our model but ignored by the baseline. Our model also correctly detects entities other than person names, such as the aircraft F16, which are not captured by the initial model. Note also that for the first name *Fabrice* and the country *Iran*, the unsupervised model correctly augments their probabilities (where the probabilities correspond to the normalized scores $f_\theta(X)$ given by the model) to belong to the class entity.

5.4 Closed-form vs. Numerical integration

This comparison is realized for task 2 to assess the relative gain in terms of computational costs when using closed-form integration. Figure 7 shows three curves, in function of the chosen setting used for numerical integration (number of trapezoids):

- The downward curve is the error: $|\hat{R}(\theta) - \hat{R}_{num}(\theta)|^{1/2}$ between the risk estimated respectively with the closed-form Eq. 4 and trapezoidal integration. We use the root-square of the approximation error to better view the details, because the trapezoidal method is known to converge in $O(n^{-2})$
- The horizontal line is the computation time (in seconds) required to compute the risk with Eq. 4; it obviously does not depend on the number of trapezoids.
- The rising line is the computation time required to compute the risk with numerical integration: it increases linearly as a function of the number of parameters used for numerical integration.

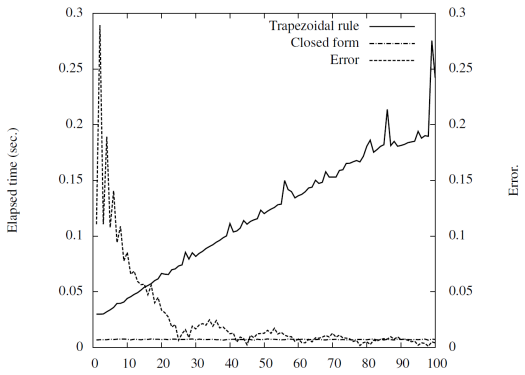


Fig. 7. Computational cost and approximated error of the trapezoidal rule with regard to the number of trapezoids (i.e., segments) used for approximating the integrals of the risk.

We can observe that increasing the number of trapezoids also increases the accuracy of numerical integration, and that the approximation error becomes smaller than 10% of the risk value for 20 trapezoids and more. This corresponds to a computational cost that is about 6 times higher for numerical integration than for closed-form estimate.

We can conclude that the advantage of the proposed closed-form solution is twofold: it reduces the required computation time by at least a factor of 6 while providing a better estimate of the risk, as compared to the original algorithm. Note however that we have not observed in our experiments any impact of this exact solution in terms of F1-measure.

6 Related Work

A number of previous works have already proposed approaches to train discriminative models without or with few labels. Please refer, e.g., to [9, 6] for a general and theoretical view on this topic. For NLP tasks several approaches have also been proposed. Hence, the traditional self- and co-training paradigm can be used to leverage supervised classifiers with unsupervised data [12, 8]. [4] exploit the Generalized Expectation objective function, which penalizes the mismatch between model predictions and linguistic expectation constraints. In contrast, our proposal does not use any manually defined prototype nor linguistic constraint.

Another interesting approach is *Unsearn* [3], which predicts the latent structure Y and then a corresponding "observation" $\hat{X}$, with a loss function that measures how well $\hat{X}$ predicts X. This method is very powerful and generalizes the EM algorithm, but its performances heavily depend on the quality of the chosen features set for discriminating between the target classes. A related principle is termed "Minimum Imputed Risk" in [11] and applied to machine translation. Our proposed approach also depends on the chosen features, but in a less crucial way thanks to both new assumptions, respectively the known label priors and discrimination of classes based on individual Gaussian distributions of scores. Another interesting generalization of EM used to train log-linear models without labels is *Contrastive Estimation*, where the objective function is modified to locally remove probability mass from implicit negative evidence in the neighborhood of the observations and transfer this mass onto the observed examples [15].

Comparatively, the main advantage of our proposed approach comes from the fact that the algorithm optimizes the standard classifier risk, without any modification nor constraint. The objective function (and related optimal parameters) is thus the same as in classical supervised training.

7 Conclusion

This work investigates the applicability of a novel framework to train linear classifiers without labels to the NLP domain. It is validated on two binary tasks, namely predicate and entity recognition. We show that convergence can only be obtained in a reasonable amount of time when initializing the classifier with good-enough values, which are obtained with supervised training on a very small amount of annotated data. We also show that the main assumption of the approach, i.e., gaussianity of the class-conditional distributions of the linear scores, is fulfilled in both our tasks. We finally propose and derive a closed-form expression of the risk estimator for a binary linear classifier, which reduces the algorithmic complexity of the proposed implementation. An interesting extension of the current approach would be to further consider some penalization term for

non-Gaussian conditional distributions of the linear scores, in order to guarantee that this assumption is preserved during the whole optimization process. We also plan to work on the optimization algorithm to further reduce its complexity and thus make it applicable to a wider range of applications. This shall also involve generalizing the risk derivation to multiclass classifiers.

References

1. Balasubramanian, K., Donmez, P., Lebanon, G.: Unsupervised supervised learning II: Margin-based classification without labels. Journal of Machine Learning Research 12, 3119–3145 (2011)
2. Björkelund, A., Hafdell, L., Nugues, P.: Multilingual semantic role labeling. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task. pp. 43–48. CoNLL '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009), http://dl.acm.org/citation.cfm?id=1596409.1596416
3. Daumé III, H.: Unsupervised search-based structured prediction. In: Proc. of ICML. Montreal, Canada (2009)
4. Druck, G., Mann, G., McCallum, A.: Semi-supervised learning of dependency parsers using generalized expectation criteria. In: Proc. of ACL. pp. 360–368. Suntec, Singapore (Aug 2009)
5. Galliano, S., Gravier, G., Chaubard, L.: The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In: Proc. of INTERSPEECH. pp. 2583–2586 (2009)
6. Goldberg, A.B.: New directions in semi-supervised learning. Ph.D. thesis, Univ. of Wisconsin-Madison (2010)
7. Gould, H., Tobochnik, J.: An introduction to computer simulation methods: applications to physical systems. No. v. 1-2 in Addison-Wesley series in physics, Addison-Wesley (1988), http://books.google.fr/books?id=JfpQAAAAMAAJ
8. Kaljahi, R.S.Z.: Adapting self-training for semantic role labeling. In: Proc. Student Research Workshop, ACL. pp. 91–96. Uppsala, Sweden (Jul 2010)
9. Kapoor, A.: Learning Discriminative Models with Incomplete Data. Ph.D. thesis, Massachusetts Institute of Technology (Feb 2006)
10. Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named entity recognition with character-level models. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. pp. 180–183. CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003), http://dx.doi.org/10.3115/1119176.1119204
11. Li, Z., Wang, Z., Eisner, J., Khudanpur, S., Roark, B.: Minimum imputed-risk: Unsupervised discriminative training for machine translation. In: Proc. of EMNLP. pp. 920–929 (2011)
12. Liu, X., Li, K., Zhou, M., Xiong, Z.: Enhancing semantic role labeling for tweets using self-training. In: Proc. AAAI. pp. 896–901 (2011)
13. van der Plas, L., Samardžić, T., Merlo, P.: Cross-lingual validity of propbank in the manual annotation of french. In: Proc. of the Fourth Linguistic Annotation Workshop, ACL. pp. 113–117. Uppsala, Sweden (Jul 2010)
14. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In: Proc. Workshop EACL SIGDAT. Dublin (1995)
15. Smith, N.A., Eisner, J.: Unsupervised search-based structured prediction. In: Proc. of ACL (2005)