# Preference Learning for Category-Ranking based Interactive Text Categorization

Fabio Aiolli, Fabrizio Sebastiani, Alessandro Sperduti

***Abstract*— *Category Ranking* is a variant of the multi-label classification problem, in which, rather than performing a (hard) assignment to an object of categories from a predefined set, we rank all categories according to their estimated "degree of suitability" to the object. Category ranking has many applications, all pertaining to "interactive" classification contexts in which the system, rather than taking a final categorization decision, is simply required to support a human expert who is in charge of taking this decision. Despite its high applicative potential in information retrieval applications, and in text categorization in particular, category ranking has mainly been tackled by standard text categorization methods. In this paper, we take a radically different stand to category ranking, i.e. one in which supervision is provided to the learner not in the standard form of labels attached to training documents, but in the form of *preferences* of type "category $c_1$ is to be preferred to category $c_2$ for document $d$". We apply to this problem a recently proposed, very general model for preferential learning, and show, through experiments performed on the standard Reuters-21578 benchmark, that this largely outperforms support vector machines, the learning method which has up to now proved the best-performing one in text categorization comparative experiments.**

## I. Introduction

*Category Ranking* (CR) is a variant of the multi-label classification problem, in which, rather than performing a (hard) assignment to a document $d$ of a (possibly empty) subset of *categories* (aka *classes*) from a predefined set $\mathcal{C} = \{c_1, \ldots, c_m\}$, we rank all categories in $\mathcal{C}$ according to their estimated "degree of suitability" to $d$.

Category ranking has many applications in Information Retrieval, all pertaining to *interactive* classification contexts. In such contexts, differently from *autonomous* Text Categorization (TC) systems [12], the system, rather than taking a final categorization decision, is simply required to support a human expert who is in charge of taking this decision. This is often the case in critical applications in which the categorization decision cannot be left to a machine. For instance, in patent classification [6], [9], [10], experts at international patent offices are presented with patent applications that they need to classify against a large, fine-grained, taxonomically organized set of classes of existing patents, in order to check the novelty of the proposed invention. These experts deem this classification operation simply too important to be left to a machine, and they want to be in charge of taking the final classification decision. However, a system that ranks the available classes in terms of their estimated suitability to the document to be classified, is extremely useful to these experts, since they can thus concentrate on the top-ranked categories, pretty much as a Web searcher concentrates on the top-ranked documents returned by a search engine following a query.

Despite its high applicative potential, CR has not received much attention from the Information Retrieval and TC communities. This can be due to two different reasons: (*i*) providing supervision in the form of rankings is more onerous for a user with respect to provide a *crisp* membership value to a document (i.e. being relevant or irrelevant); (*ii*) there are not ad hoc learning methods which can cope with this task in a principled way.

For what concerns this second point, to the best of our knowledge, there are only a few pioneering papers (e.g. [4] and [7]) that tackles this problem as such. Up to now, the dominant approach to category ranking has instead involved the application of standard methods for multi-label TC. By and large, this means training, for each category $c_i$, a binary classifier $\Phi_i$ that returns *confidence-rated predictions*, i.e. scores $\Phi_i(d) \in \mathbb{R}$ expressing the system's confidence in the fact that $d \in c_i$. Categories are then ranked based on the confidence scores returned by the respective classifiers when asked to classify $d$.

In this paper we take a radically different stand to CR, i.e. one in which supervision is provided to the learning device not in the "standard" form of labels attached to training documents, but in the form of *preferences*. These preferences can be of two different types:

1) *qualitative preferences*, expressing the relative suitability of two categories for a given document; e.g. "category $c_i$ is to be preferred to category $c_j$ for document $d$" (denoted $c_i \rhd_d c_j$);
2) *quantitative preferences*, expressing the degree of suitability of a category for a given document; e.g. "the degree of suitability of category $c_i$ for document $d$ is at least $\tau$" (denoted $c_i \rhd_d \tau$), or similarly "the degree of suitability of category $c_i$ for document $d$ is at most $\tau$" (denoted $\tau \rhd_d c_i$).

Note that training information of the standard form (i.e. labels attached to training documents) can be viewed in terms of preferences, by assuming that

1) whenever $d \in c_i$ and $d \notin c_j$, then $c_i \rhd_d c_j$;
2) whenever $d \in c_i$, then $c_i \rhd_d \tau$, and whenever $d \notin c_i$, then $\tau \rhd_d c_i$.

*Fabio Aiolli* and *Alessandro Sperduti* are with the Dipartimento di Matematica Pura ed Applicata, Universita di Padova, Via Trieste 63, 35121 Padova - Italy (e-mail: {aiolli,sperduti}@math.unipd.it); *Fabrizio Sebastiani* is with the Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi 1, 56124 Pisa - Italy (e-mail: fabrizio.sebastiani@isti.cnr.it)

We apply to this problem a recently proposed, very general model for learning from preferences, called the *Preference Learning Model* (PLM) [1], [2], [3]. While the PLM was especially devised for learning from information that is *naturally* expressed as preferences (e.g. when all we know about a document $d$ and two categories $c_i$ and $c_j$ is that $c_i$ is to be preferred to $c_j$ for $d$ *without knowing* whether $d$ actually belongs to $c_i$ and/or $c_j$), it can also be fruitfully applied to contexts (such as CR) in which supervision is naturally expressed in terms of labels attached to documents. In fact, the strength of the PLM is that it is able to set its internal parameters in a way that maximizes the effectiveness of the category ranking produced for the training examples. In this way, effectiveness measures specific to category ranking can be brought to bear, and this (hopefully) allows the PLM to outperform methods in which such effectiveness measures cannot be plugged in.

Indeed, the PLM approach shares some similarities with the approaches in [4] and [7]. However, [4] is based on an on-line setting and is far less flexible than PLM. Similarly, the framework in [7] can be seen as a particular case of PLM which does not take conjunctions of preferences into consideration. A feature that makes PLM far more attractive.

### A. Outline of the paper

The paper is structured as follows. In Section II we describe the PLM and its application to Information Retrieval tasks. In Section III we describe how both the multilabel text categorization and the category ranking tasks for Information Retrieval can be naturally modelled in the PLM framework thus providing us with a principled solution to these settings. Section IV reports on our experiments, by briefly reviewing the Reuters-21578 benchmark we have used and the experiments we have conducted on it. Finally, Section V concludes.

## II. The Preference Learning Model

In short, in PLM we assume the existence of a real-valued relevance function that for each document $d$ and category $c$ returns a score, $r(d, c)$ (the relevance value), which "measures" the degree to which category $c$ applies to document $d$ (and viceversa, how relevant the document is for that category). Thus the relevance function, for each document $d$, induces a ranking among categories. A preference is a constraint involving two categories that should be satisfied by the relevance function. Specifically, PLM focuses on two types of preferences: *qualitative* preferences $c_k \rhd_d c_s$, where a category $c_k$ is told to be preferred to a category $c_s$ ("category $c_k$ applies to document $d$ more than $c_s$ does"), i.e. $r(d, c_k) > r(d, c_s)$, and *quantitative* preferences of type $c \rhd_d \tau$ ("the degree to which category $c$ applies to document $d$ is at least $\tau$"), i.e. $r(d, c) > \tau$, and $\tau \rhd_d c$ ("the degree to which category $c$ applies to document $d$ is at most $\tau$"), i.e. $r(d, c) > \tau$, where $\tau \in \mathbb{R}$.

In this learning model, supervision for a document is given as a set of preferences (of any type). These preferences constitute contraints on the form of the relevance function which has to be learned. The aim of the learning process is to return a relevance function which is as much consistent with these constraints as possible.

As a very simple example of how to model supervised problems in PLM, let consider the (single-label) multiclass problem where a classifier has to predict the most relevant category $c_p$ for a given document $d$. This case can be modelled by introducing for each training document $d$ a set of preferences $\{c_p \rhd_d c_i\}_{c_i \neq c_p}$. Note that, when testing a new document, the prediction is given by the category which maximizes the relevance with respect to the testing document, i.e. the unique category which satisfies all the associated preferences. As a further example, the (multi-label) multiclass problem can be easily modeled by considering preferences of the form

$$\{(c_i \rhd_d \tau)\}_{c_i \in Rel(d)} \cup \{(\tau \rhd_d c_j)\}_{c_j \in \mathcal{C} \setminus Rel(d)}$$

where $\tau$ is a real valued reference threshold, $\mathcal{C}$ is the set of categories, and $Rel(d) \subseteq \mathcal{C}$ is the set of relevant categories for document $d$. In this case, the set of relevant categories are obtained by comparing the associate relevance value against the (possibly category-dependent) threshold, that is, a category is considered relevant if and only if its relevance with respect to the document is above this threshold. It should be stressed that in PLM, any set of preferences can be associated to a document, so if there is no information about the relative ranking of two categories, no preference involving these two categories need to be inserted. This allows to impose on the learner only constraints which are strictly necessary thus using the available information only which alone might be sufficient to solve the problem.

In the simpler version of PLM, the relevance of a category for a given document is assumed to have a linear form:

$$r(d, c) = \mathbf{w}_c \cdot \phi(d) \tag{1}$$

where $\phi(d) \in \mathbb{R}^k$ is one of the standard vectorial representations for the documents ("bag-of-words", *tf-idf*, etc.), or any other feature mapping, and $\mathbf{w}_c$ are weight vectors (parameters to learn) associated to the different categories, i.e. $c \in \{c_1, \ldots, c_m\}$. Interestingly, for this case, it is possible to give effective algorithms which explicitly attempt to minimize the number of wrong predictions in a given training set. In fact, following equation (1), qualitative and quantitative preferences can be conveniently reformulated as linear constraints. Specifically, let consider the qualitative preference $p \equiv (c_i \rhd_d c_j)$. This preference imposes the constraint $r(d, c_i) > r(d, c_j)$ on the relevance function, which using equation (1) can be rewritten as $\mathbf{w}_{c_i} \cdot \phi(d) > \mathbf{w}_{c_j} \cdot \phi(d)$, or $\big(\mathbf{w}_{c_i} \cdot \phi(d) - \mathbf{w}_{c_j} \cdot \phi(d)\big) > 0$. Similar transformations can be done for quantitative preferences. A uniform treatment of the preferences can then be obtained by concatenating all the $\mathbf{w}_c$'s, $c \in \{c_1, \ldots, c_m\}$, and all the thresholds $\tau_1, \ldots, \tau_q$ involved in the formulation of the problem, i.e. $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_m, \tau_1, \ldots, \tau_q) \in \mathbb{R}^{mk+q}$. In the qualitative case, given the above preference $p$ and assuming $k < s$ with no loss in generality, we have

$$\mathbf{w}\cdot\underbrace{(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{i-1},\phi(d),\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{j-i-1},-\phi(d),\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{m-j},\underbrace{0,\ldots,0}_{q})}_{\psi(p)}>0,$$

where $\psi(p) \in \mathbb{R}^{mk+q}$ is the representation for $p$. In the quantitative case, the preference $p \equiv (c_i \rhd_d \tau_j)$ can similarly be expressed as

$$\mathbf{w}\cdot\underbrace{(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{i-1},\phi(d),\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{m-i},\underbrace{0,\ldots,0}_{j-1},-1,\underbrace{0,\ldots,0}_{q-j})}_{\psi(p)}>0,$$

while preference $p \equiv (\tau_j \rhd_d c_i)$ is expressed as

$$\mathbf{w}\cdot\underbrace{(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{i-1},-\phi(d),\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{m-i},\underbrace{0,\ldots,0}_{j-1},1,\underbrace{0,\ldots,0}_{q-j})}_{\psi(p)}>0.$$

In general, the supervision can be reduced into sets of particular linear constraints of the form $\mathbf{w}\cdot\psi(p) > 0$ where $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_m, \tau_1, \ldots, \tau_q)$ is the vector of weights augmented with the set of available thresholds and $\psi(p)$ is an opportune representation of the preference under consideration. As a consequence, any setting described by this theory can be seen as a (homogeneous) linear problem in an opportune augmented space. Specifically, any algorithm for linear optimization (e.g. perceptron or a linear programming package) can be used to solve it, provided the problem has a solution.

Unfortunately, the set of preferences may generate a set of linear constraints that have no solution (not linearly separable by an hyperplane passing from the origin), i.e. there is no weight $\mathbf{w}$ able to fulfill all the constraints induced by the preferences in the training set. To deal with training errors, we may resort to the Structural Risk Minimization (SRM) theory [13]. This is made by considering the minimization of an objective function which aims at minimizing the number of unfulfilled preferences (the training error) while maximizing the margin (the inverse of the weights norm). See Section II-C.

### A. Evaluation and PLM

The mere consistency of supervision constraints is not necessarily the ultimate goal of a supervised learning setting. Rather, cost functions are often preferred measuring the disagreement between the current prediction and the target supervision. These functions may either depend on the particular structure of the prediction or other factors. For example, the evaluation of a non-perfect category ranking result can be better described as the number of categories which are misordered instead of simply as an error. For this, in [2] supervision is mapped into sets of preferences.

Specifically, supervision $S$ is described by a preference set, denoted $g[S]$, and a cost mapping

$$\mathcal{G} : g[S] \mapsto \{g_1, \ldots, g_{q_S}\}$$

is defined, where each preference set $g_i$ is a subset of $g[S]$. Once the cost mapping $\mathcal{G}$ is defined, the total cost suffered by an hypothesis for the supervision $S$ is defined as the number of preference sets which are not satisfied by the current hypothesis. More formally, we have

$$cost(g[S]) = \sum_{j=1}^{q_S} [\![ g_j ]\!]. \qquad (2)$$

where $[\![ g ]\!]$ is an operator which is equal to 1 when there are constraints in $g$ which are not satisfied by the current hypothesis $r_{\mathbf{w}}(\cdot,\cdot)$, and 0 otherwise.

### B. Cost Mapping Examples

In order to better understand the PLM setting defined above, in this section, we briefly present cost mapping examples for a simple category ranking problem, the *single-label* multi-class classification problem. This task can be considered a category-ranking problem when, for each document, we want only one of the categories (the most relevant) to be ranked over the others and returned. Let $\mathcal{C} = \{c_1, c_2, c_3\}$, and $d$ a document which has to be classified as $c_1$. In PLM, a natural cost mapping for this problem corresponds to have a preference set like $g[c_1] \mapsto \{\{c_1 \rhd_d c_2, c_1 \rhd_d c_3\}\}$. The same preference set can however be decomposed in two separate preference sets, thus obtaining $g[c_1] \mapsto \{\{c_1 \rhd_d c_2\}, \{c_1 \rhd_d c_3\}\}$. Note that, these mappings will induce different cost functions. For example, let the current hypothesis such that $r_{\mathbf{w}}(d, c_3) > r_{\mathbf{w}}(d, c_2) > r_{\mathbf{w}}(d, c_1)$ then we have a cost equal to 1 in the first case and a cost of 2 in the second. In fact, using the last definition of cost, two preference sets are not satisfied. Specifically, in the first case, we count an error when there is a category different from the correct one on the top. Viceversa, in the second case, we count the number of uncorrect categories which are ranked over the correct one. The two examples above give a rough idea of the flexibility of the preference learning model.

### C. Learning in the PLM

In earlier sections we have discussed how a cost function for general supervised learning problems can be modeled using preference sets. Now, we see how to give a general learning algorithm which is able to learn from preferences.

Supervised learning algorithms aim at minimizing the *true cost*, that is the expected value of the cost according to the true distribution of data, i.e. $R_t[\mathbf{w}] = E_{S\sim\mathcal{D}}[c(S|\mathbf{w})]$. The distribution $\mathcal{D}$ is typically unknown, while it is available a training set $\mathcal{S} = \{S_1, \ldots, S_n\}$ with supervision drawn $i.i.d.$ from $\mathcal{D}$. An empirical approximation of the true cost, also referred to as the *empirical cost*, is defined by

$$R_e[\mathcal{S}|\mathbf{w}] = \frac{1}{n}\sum_{i=1}^{n} c(S_i|\mathbf{w}).$$

Similarly, in PLM, the aim is to minimize costs, as defined in Eq. (2), induced by the cost mappings performed over the training set $\mathcal{S}$. Unfortunately, these functions are not continuous with respect to the parameters $\mathbf{w}$ and hence not easily treatable. To overcome this problem, consider the quantity $\rho(p|\mathbf{w}) = \mathbf{w} \cdot \psi(p)$ as a degree of satisfaction of a preference $p$ given the hypothesis $\mathbf{w}$. This value is greater than zero when the hypothesis is *consistent* with the preference and less than zero otherwise. Now, an approximation to the error is obtained by introducing the soft-margin loss, the continuous non-increasing function $l(\rho) = [1-\rho]_+ = \max(0, 1-\rho)$, which upper-bounds the indicator function $I(\rho)$ which is 1 when $\rho > 0$ and 0 otherwise. Specifically, this approximated cost will be

$$\tilde{c}(S|\mathbf{w}) = \sum_{g \in \mathcal{G}(g[S])} \max_{p \in g} [1 - \rho(p|\mathbf{w})]_+.$$

Given the assumptions above, one can notice that the function $\tilde{c}(S|\mathbf{w})$ upper-bounds the empirical cost over the whole training set and the general problem can be formulated as in the follows.

Given a set

$$\mathcal{V}(\mathcal{S}) = \bigcup_{S \in \mathcal{S}} g[S] = \{g_1, \ldots, g_N\}$$

of $N$ preference sets describing the supervision given to the algorithm, we want to find a set of parameters $\mathbf{w}$ in such a way to minimize the functional

$$\mathcal{Q}(\mathbf{w}) = \mathcal{R}(\mathbf{w}) + \gamma \mathcal{L}(\mathcal{V}(\mathcal{S})|\mathbf{w}) \qquad (3)$$

where $\mathcal{L}(\mathcal{V}(\mathcal{S})|\mathbf{w}) = \sum_{S \in \mathcal{S}} \tilde{c}(S|\mathbf{w})$ is related to the empirical cost, $\mathcal{R}(\mathbf{w})$ is a regularization term over the set of parameters, and $\gamma$ the trade-off parameter.

The use of a regularization term on a problem of this type has many different motivations, including the theory on regularization networks (see e.g. [5]). Moreover, we can see that by choosing a convex loss function and a convex regularization term (let say the quadratic term $\mathcal{R}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$) it warranties the convexity of the functional $\mathcal{Q}(\mathbf{w})$ in Eq. (3) and then the solution does not have the problem of local minima. In our particular case, we obtain the following constrained quadratic problem

$$\begin{array}{l} \min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^2 + C \sum_i^N \xi_i \\ \text{subject to:} \left\{ \begin{array}{ll} \mathbf{w} \cdot \psi(p) \geq 1 - \xi_i, & \forall i \in \{1,..,N\}, \forall p \in g_i \\ \xi_i \geq 0, & \forall i \in \{1,..,N\} \end{array} \right. \end{array} \qquad (4)$$

This formulation resembles the SVM formulation where we have a constraint for each preference. However, in this case, a single slack variable is present binding multiple constraints associated to the same preference set. Indeed, this is a generalization of SVM to more general cost functions which are defined by preferences.

Moreover, one can show that the solution $\mathbf{w}$ of the previous problem will take the (sparse) form:

$$\mathbf{w} = \sum_{i,r} \alpha_i^r \psi(p_i^r)$$

where $\alpha_i^r \in \mathbb{R}$, and $p_i^r$ is the $r$-th preference of the $i$-th example, Similarly to SVM, the final solution will have only a few $\alpha_i^r > 0$ (support vectors).

Since the solution is expressed by dot products only, then any kernel function can be used in place of dot products. Thus, this method constitutes a new kernel method which is able to solve any problem defined by preferential information (see [1] for details). Moreover, changing the cost function means to redefine the cost mapping but still keeping the same solver.

## III. From categorization to ranking

A well-known baseline approach to category ranking, when categorical supervision is available, is to train a classifier independently for each class by using the supervision, and then to rank categories based on the confidence of the output of different classifiers. In PLM this approach can be modelled by introducing a preference $(c \rhd_d \tau$ whenever $d$ is member of class $c$ and a preference $\tau \rhd_d c$ when this is not the case. Moreover, the very common cost function used for multi-label categorization, computed as the number of document-category pairs which are not correctly classified, can be obtained in PLM by defining a mapping which takes all the preferences $c \rhd_d \tau$ and $\tau \rhd_d c$ independently.

The (bipartite) category ranking task, however, is slightly different since in this case it is required to produce a full order such that some classes are ranked over the remaining classes. One can note that if a set of examples is correctly categorized (thus having cost zero in the previous cost mapping), the produced ranking is correct. The other way around is not necessarily true. In fact, given a correct ranking, it could be impossible to find an optimal threshold determining the correct target categorization.

In PLM, however, considering all the documents for which we have categorical supervision in the training set, a complete preference set can be built as the union of all the preferences derivable by them through transitivity closure. This set highlights additional information which is not directly expressed in the original categorical form. For example, we have new relations/preferences like $c_r \rhd_d c_s$ whenever $d$ belongs to $c_r$ and $d$ does not belong to $c_s$. This highlights that categorization information subsumes information over document and category rankings which is not self-evident when single documents or single classes are treated independently. An interesting point now is how to use this additional information to improve over current methods for categorization and category ranking.

In this paper, we focus our exposition on the experimental comparison of the SVM baseline approach versus an approach that uses category ranking information only by considering independently those preferences which are defined over different documents and not considering the threshold $\tau$. Specifically, the following cost mappings will be considered. The *disagreement mapping* (DIS), which considers each preference $c_r \rhd_d c_s$, whenever $d \in c_r$ and $d \notin c_s$, independently. As an alternative cost mapping, the *domination mapping* (DOM) will be considered. The basic idea underpinning the use of the domination mapping is that for each input document we prescribe that the score assigned by the predictor to any positive class should be higher than the score obtained by any negative class, and the cost will be non-zero whenever any negative class gets a score above the score of the considered positive class. Costs suffered by positive classes are cumulated. A dual cost mapping, which we consider for completeness, is the *dominated mapping* (DME), where the roles of positive and negative examples described above are exchanged, while keeping the score of positive examples to be higher. It should be clear that using either the *domination mapping* or the *dominated mapping* does not exploit all the possible ranking information we have available. In fact, in this way we give more emphasis to supervision concerning positive, or negative, classes for single documents, respectively. Interestingly, we may try to compound these two cost mappings in a joint *domination-dominated mapping* (DOM-DME) which simply cumulates the cost obtained for the two separately.

It is worthwhile to recall that all these cost mappings are plugged into the same algorithm. A single learning algorithm is able to cope with all these cost functions in a very modular way.

## IV. Experiments

### A. Experimental setting

In our experiments we have used the "Reuters-21578, Distribution 1.0" corpus, currently the most widely used benchmark in text categorization research[1]. Reuters-21578 consists of a set of 12,902 news stories, partitioned (according to the "ModApté" split we have adopted) into a training set of 9,603 documents and a test set of 3,299 documents. We have discarded the categories that have no training examples, leaving us with 115 categories with at least one training example. We have also discarded all the (training and test) documents that have no label (originally, these documents were meant to be considered legitimate negative examples for all categories); note in fact that any CR system would perform equally well on test documents of this type under any reasonable evaluation metric (namely, it would return an ordered list of 115 false positive labels). This leaves us with a training set $\mathcal{S}$ consisting of 7,775 documents and a test set $\mathcal{T}$ of 3,019 documents. The average number of categories per document is 1.08, ranging from 1 to 16; the number of positive examples per category ranges from 1 to 3964.

In all the experiments discussed in this section, stop words have been removed using the stop list provided in [11, pages 117–118]. Punctuation has been removed, letters have been converted to lowercase, numbers have been removed, and stemming has been performed by means of Porter's stemmer.

We have measured effectiveness in terms of *normalized microaveraged precision as a function of rank* ($\hat{\pi}^m(r)$), an adaptation of the *microaveraged precision* ($\pi^m$) measure to category ranking. Let us introduce this measure in steps. Precision wrt document $d_j$ (denoted by $\pi_j$) is defined as the proportion of true positive labels for $d_j$ out of the total of (true and false) positive labels of $d_j$. Microaveraged precision (denoted by $\pi^m$) is obtained by averaging $\pi_j$ values across all the test documents $d_j$, i.e. $\pi^m = \frac{1}{|\mathcal{T}|}\sum_{j=1}\pi_j$, and encodes the basic intuition that each document has the same importance[2]. Microaveraged precision can be evaluated at each rank position $r$ (this is denoted by $\pi^m(r)$). However, as such, this measure has problems, due to the fact that, when $r$ is higher than the number $n_j$ of the true labels of $d_j$, a perfect classifier (i.e. the one that has ranked the $n_i$ labels of $d_j$ at the first $n_j$ rank positions) would not achieve the theoretical maximum precision value of 1, since the remaining $r - n_j$ labels would be false positives anyway. As a result, as our final effectiveness measure we adopt $\hat{\pi}^m(r)$, which we define as the $\pi^m(r)$ of our classifier, normalized by the $\pi^m(r)$ of the 'ideal' ranker.

Effectiveness is thus plotted on a graph (see Figure1) in which the $x$ axis is the rank position, ranging between 1 and 115 (the number of categories in our benchmark); the higher the plot, the better. Note that all CR systems have the same effectiveness value for $x = 115$; this corresponds to the notion that, after scanning the ranked list of labels down to the bottom, the user has encountered all true labels and all false labels irrespectively of the category-ranking system used.

Note that, unlike in standard multilabel TC, we do not use $F_1$ (the harmonic mean of precision and recall) or variants thereof. In particular, we dispense with explicitly considering recall, since precision, when computed at a fixed rank position $r$, already "contains a recall component" (i.e. at a fixed rank $r$, an improvement of precision *strictly entails* an improvement in recall).

### B. Experimental Methodology

The aim of the experimental work was to compare the performance obtainable with different PLM mappings and to compare the PLM setting as a whole against the standard multilabel SVM. For each document, the ranking produced by PLM is the one induced by its relevance function, while, for the SVM case, the signed scores obtained as outputs by the multilabel SVM, have been used instead.

In order to fairly evaluate the different techniques, we performed model-selection by cross-validation. Specifically, the training set was split in 5 different folders. Then, each folder has been used as a test sample for the model trained with examples from the remaining folders only, for parameters $\gamma = 10^z$, $z = \{-2, \ldots, +2\}$. Each model has been evaluated in this phase by means of its own cost function. Finally, a complete training session has been performed for each method over the whole training set using the corresponding optimal parameter. The results reported in the graph refer to the evaluation of the obtained model over the test sample.

From the analysis of results one can evince that the multilabel SVM based method is largely worse than preference-

[1] The Reuters-21578 corpus is freely available for experimentation purposes from `www.daviddlewis.com/resources/testcollections/reuters21578/`

[2] The alternative notion of *macroaveraged precision* encodes the notion that documents count proportionally to the number of categories by which they are labelled.

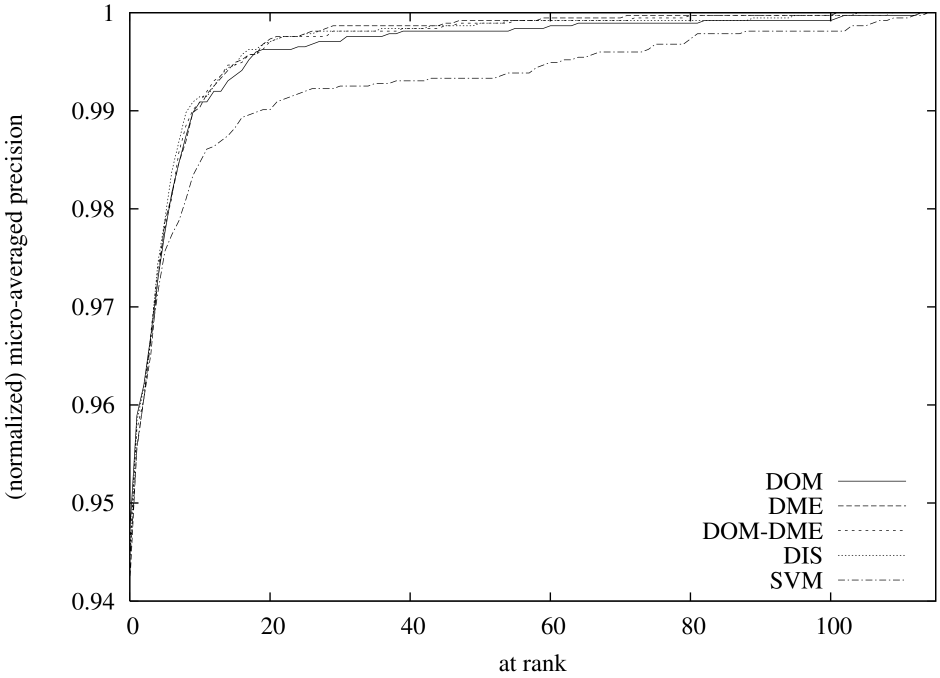

Fig. 1. Normalized micro-averaged ”precision at rank” obtained for the baseline multi-label SVM, and different PLM mappings: disagreement (DIS), domination (DOM), dominated (DME) and their combination (DOM-DME). For each position, the proportion of true positives is reported.

based methods on the category ranking problem under consideration. Specifically, we may observe that DIS and DOM-DME have a similar behavior along the whole range of available ranks, while DOM seems to slightly predominate over lower ranks and DME dominates over higher ranks.

## V. Conclusion

In this paper we have shown how the Preference Learning Model, a general framework for learning from training information expressed in preferential form, can be applied to the task of category ranking, and can outperform learning methods that are current top performers in the text categorization task. This is achieved by exploiting the ability of the PLM to explicitly maximize effectiveness functions that are specific of ranking tasks, i.e. optimize its internal parameters so that these functions, rather than “generic” effectiveness functions aimed at standard multilabel text categorization, are maximized. The model allows to codify cost functions as preferences and naturally plug them into the same training shell. Furthermore, it gives a tool for comparing different methods and cost functions on a same learning problem.

We are currently extending this new paradigm to tasks in which, unlike in the present setting, supervision *naturally* comes in preferential form. In text categorization, this is the case e.g. of applications, such as classifying medical articles in the OHSUMED collection [8] or classifying patents in the WIPO-alpha collection [6], in which training documents are labelled with “primary” and “secondary” categories.

## References

[1] Fabio Aiolli. *Large Margin Multiclass Learning: Models and Algorithms*. PhD thesis, Dept. of Computer Science, Univ. of Pisa, 2004.

[2] Fabio Aiolli. A preference model for structured supervised learning tasks. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM’05)*, pages 557–560, Houston, US, 2005.

[3] Fabio Aiolli and Alessandro Sperduti. Learning preferences for multiclass problems. In *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS’04)*, pages 17–24, Vancouver, CA, 2004.

[4] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 151–158, Tampere, FI, 2002.

[5] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[6] C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. Automated categorization in the International Patent Classification. *SIGIR Forum*, 37(1):10–25, 2003.

[7] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification for multiclass classification and ranking. In *Advances in Neural Information Processing Systems*, pages 785–792, Cambridge, MA, 2002. MIT Press.

[8] William Hersh, Christopher Buckley, T.J. Leone, and David Hickman. OHSUMED: an interactive retrieval evaluation and new large text collection for research. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 192–201, Dublin, IE, 1994.

[9] Marc Krier and Francesco Zaccà. Automatic categorization applications at the european patent office. *World Patent Information*, 24:187–196, 2002.

[10] Leah S. Larkey. A patent search and classification system. In *Proceedings of DL-99, 4th ACM Conference on Digital Libraries*, pages 179–187, Berkeley, US, 1999.

[11] David D. Lewis. *Representation and learning in information retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.

[12] David D. Lewis. Evaluating and optmizing autonomous text classification systems. In *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, US, 1995.

[13] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.