Inf Retrieval (2009) 12:559–580
DOI 10.1007/s10791-008-9071-y

Preferential text classification: learning algorithms and evaluation measures

Fabio Aiolli · Riccardo Cardin · Fabrizio Sebastiani · Alessandro Sperduti

Received: 3 September 2007 / Accepted: 8 September 2008 / Published online: 9 October 2008

Abstract In many applicative contexts in which textual documents are labelled with thematic categories, a distinction is made between the primary categories of a document, which represent the topics that are central to it, and its secondary categories, which represent topics that the document only touches upon. We contend that this distinction, so far neglected in text categorization research, is important and deserves to be explicitly tackled. The contribution of this paper is threefold. First, we propose an evaluation measure for this *preferential text categorization* task, whereby different kinds of misclassifications involving either primary or secondary categories have a different impact on effectiveness. Second, we establish several baseline results for this task on a well-known benchmark for patent classification in which the distinction between primary and secondary categories is present; these results are obtained by reformulating the preferential text categorization task in terms of well established classification problems, such as single and/or multi-label multiclass classification; state-of-the-art learning technology such as SVMs and kernel-based methods are used. Third, we improve on these results by using a recently proposed class of algorithms explicitly devised for learning from training data expressed in preferential form, i.e., in the form "for document d_i, category c' is preferred to category c''"; this allows us to distinguish between primary and secondary categories not only in the

F. Aiolli · R. Cardin · A. Sperduti
Dipartimento di Matematica Pura e Applicata, Università di Padova,
Via Trieste, 63-35121 Padova, Italy
e-mail: aiolli@math.unipd.it

R. Cardin
e-mail: riccardo.cardin@gmail.com

A. Sperduti
e-mail: sperduti@math.unipd.it

F. Sebastiani (✉)
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

classification phase but also in the learning phase, thus differentiating their impact on the classifiers to be generated.

Keywords Preferential learning · Supervised learning · Text categorization · Text classification · Primary and secondary categories

1 Introduction

In many applicative contexts in which textual documents are labelled with thematic categories, a distinction is made between the primary and the secondary categories that are attached to a given document. The *primary* category (or categories) of a document represent the topic(s) that are central to the document, or that the document is mainly about. The *secondary* categories represent instead topics that are somehow touched upon, albeit peripherally, in the document, and do not represent, thematically speaking, the main thrust of the document. This distinction has been neglected in text categorization (TC) research. For instance, a systematic search through the literature shows that all authors who have tested their TC systems on the WIPO-alpha collection (Fall et al. 2003) have either considered the primary categories alone, thus basically using WIPO-alpha as a single-label classification dataset (Cai and Hofmann 2004; Fall et al. 2003; Hofmann et al. 2003; Tikk and Biró 2003; Tikk et al. 2004; Tsochantaridis et al. 2004; Vishwanathan et al. 2006), or collapsed primary and secondary categories, thus using it as a for multi-label classification dataset (Cai and Hofmann 2007; Rousu et al. 2006). The same happens for those who have worked on the OHSUMED collection (Hersh et al. 1994), who have all collapsed primary and secondary categories (Forman 2003; Lam and Ho 1998; Lewis et al. 1996; Ruiz and Srinivasan 2002; Yang et al. 2003).[1]

We instead contend that the distinction between primary and secondary categories is important, and deserves to be explicitly tackled by TC research. The main reason is that, in most contexts in which the distinction is made, misclassifications may be more or less serious, depending on whether they involve a primary or a secondary category.

For instance, when a patent application is submitted to the European Patent Office (EPO), a primary category from the International Patent Classification (IPC) scheme[2] is attached to the application, and that category determines the expert examiner who will be in charge of evaluating the application. Secondary categories are instead attached for the only purpose of identifying related prior art, since the appointed expert examiner will need to determine the novelty of the proposed invention against existing patents classified under either the primary or any of the secondary categories. For the purposes of EPO, failing to recognize the true primary category of a document is thus a more serious mistake than failing to recognize a true secondary category.[3]

[1] Both OHSUMED and WIPO-alpha make a distinction between primary and secondary categories. A third dataset in which this distinction is made is the ICCCFT dataset used in the 2007 "International Challenge on Classifying Clinical Free Text Using Natural Language Processing" (http://www.computationalmedicine.org/challenge/index.php). Other well-known TC datasets such as Reuters-21578 or RCV1 instead do not make this distinction.

[2] http://www.wipo.int/classifications/en/.

[3] Barrou Diallo, personal communication. Barrou Diallo is Head of the Research and Development Directorate at the European Patent Office.

Another instance is represented by the ACM Computing Reviews magazine,[4] which publishes reviews of articles and books related to computer science, each classified according to one primary and several secondary categories from the ACM Computing Classification System.[5] Here the primary category determines in which section of the magazine the review is going to be printed, while secondary categories, together with the primary category, are used for facilitating search (e.g., allowing a user to search only the reviews belonging to a particular category). Again, for the purposes of ACM Computing Reviews, getting the primary category of a document wrong is thus a more serious mistake than failing to recognize a true secondary category.[6]

This paper represents a first attempt at solving *preferential text categorization*, a task which we define as the attribution to a textual document d_i not of a subset $C_i \subseteq C$ of the set of categories C (as in standard *multi-label*—aka "*n*-of-*m*"—text categorization), but of a *partial ordering* among the set of categories C; this partial ordering specifies which category "applies more than" (or "is preferred to") which other category to the document. This is itself a variant of the so-called "category ranking problem", which consists in attributing to a textual document a *total* ordering among the categories (Crammer and Singer 2002). We will here discuss a special case of preferential TC, that is, the case in which each document is associated to a "three-layer" partial order, consisting of a top layer of one or more primary categories,[7] which are each preferred to each of a mid layer of secondary categories, which are each preferred to each of a bottom layer of "non-categories" (i.e., categories that do not apply at all to the document).

The original contribution of this paper is threefold. First, we propose an evaluation measure for preferential TC, in which different kinds of misclassifications involving either primary or secondary categories have a different impact on effectiveness.

Second, we establish baseline results for this task on the WIPO-alpha collection. These results (expressed in terms of the evaluation measure defined in the previous step) are obtained by decomposing the 3-layered preferential TC problem into simpler TC problems for which standard state-of-the-art learning tools can be used. In WIPO-alpha each document is associated to a single primary category and multiple secondary categories. Therefore, we first test a "poor man's baseline" consisting of generating binary classifiers for all categories involved by means of binary SVMs and picking, as the primary category of document d, the category which has received the highest score. We then establish an alternative, stronger baseline by decomposing our problem into (i) a *single-label* (aka 1-of-*m*) TC problem, aimed at determining the primary category, and for which we use a recent, top-performing type of multiclass SVMs (Platt et al. 1999); and (ii) a multi-label TC problem, aimed at determining which of the remaining $m - 1$ categories is a secondary category of the document, and for which we use plain binary SVMs (Joachims 1998).

The combined use of these two technologies allows us to attach primary and secondary categories to the test documents, but uses the categories attached to the training documents in a traditional way. That is, a training document d' for which category c is a primary category has the same impact on the classifier for c of another training document d'' for which c is a secondary category; we think this is unintuitive. Finally, we formulate the

[4] http://www.reviews.com/.

[5] http://www.acm.org/class/1998/.

[6] Carol Hutchins, personal communication. Carol Hutchins is Editor-in-Chief of ACM Computing Reviews.

[7] Whether one or several primary categories are possible will depend on the application. For instance, in the WIPO-alpha collection only one primary category per document is allowed, while in the OHSUMED collection several primary categories for the same document may exist.

problem as a (multivariate) ordinal regression task, where, given a document, each category is associated to one of the three possible ordered ranks, i.e. primary, secondary, non-category.

Our third contribution is thus an improvement on the above baseline results obtained by using a learning model, dubbed the *Generalized Preference Learning Model* (Aiolli and Sperduti 2005), that was explicitly devised for learning from training data expressed in preferential form, i.e., in the form "category c' is preferred to category c'' for document d". This model, which does not require any decomposition into subproblems, allows us to draw a fine distinction between primary and secondary categories not only in the testing phase but also in the learning phase, and to leverage on the different importance that primary and secondary categories attached to a training document have.

1.1 Outline of the paper

The paper is structured as follows. In Sect. 2 we propose an evaluation function for preferential TC. Section 3 introduces the learning algorithms by means of which we will tackle the preferential TC task, from the "standard" (both multiclass and binary) SVM technology that will allow us to obtain baseline results (Sect. 3.1), to more novel "preference learning" technology by means of which we improve on this baseline (Sect. 3.2). Section 4 reports on our experiments, by briefly reviewing the WIPO-alpha dataset we have used and the experiments we have conducted on it. Section 5 concludes.

2 Preferential text categorization and its evaluation

The de facto standard measure for the evaluation of a binary classifier for category c is

$$F_1(c) = \frac{2\pi(c)\rho(c)}{\pi(c) + \rho(c)} \tag{1}$$

which corresponds to the harmonic mean of precision ($\pi(c)$) and recall ($\rho(c)$) (Lewis 1995). Since multi-label TC resolves to binary TC, averaging $F_1(c)$ across the categories is the standard way of evaluating multi-label TC.

However, F_1 is too coarse for preferential TC, since it is the very notions of precision and recall that are too coarse for properly addressing this task. For instance, precision is the probability that, if a category has been assigned to a document, this assignment was correct. But what does, in preferential TC, "assigning a category" mean? Assigning it as a primary or as a secondary category are different things.

2.1 Why Kendall distance is unsuitable

It might be tempting to hypothesize that the "right" evaluation measure for preferential TC is one from the tradition of evaluating automatically produced rankings (*predicted rankings*) σ_P against a "true" ranking σ_T. The (*normalized*) Kendall distance with penalization p (noted $K^p(\sigma_P, \sigma_T)$—see Fagin et al. 2004, 2006) is nowadays considered the standard function for the evaluation of rankings in which "ties" (i.e., pairs of objects which have the same place in one of the two rankings) might occur; K^p and its variants are now heavily used in the evaluation of several IR-related tasks involving ranking, such as ranking system runs based on their results on a given test collection (Voorhees 1998), ranking topics by

estimated difficulty (Yom-Tov et al. 2005), or computing the similarity between features in terms of the document rankings that their use brings about (Geng et al. 2007). It is a non-symmetrical[8] distance (i.e., $K^p(\sigma_i, \sigma_j) \neq K^p(\sigma_j, \sigma_i)$) defined as

$$K^p = \frac{n_d + p \cdot n_u}{Z} \tag{2}$$

where n_d is the number of *swappings*, i.e., pairs of objects ordered one way in σ_T and the other way in σ_P; n_u is the number of *false ties*, i.e., pairs ordered not tied in σ_T and tied in σ_P; p is a penalization to be attributed to each false tie; and Z is a normalization factor (equal to the number of pairs that are ordered in the true ranking) whose aim is to make the range of τ_p coincide with the [0,1] interval. *True ties* (i.e., pairs tied in σ_T) are not considered by K^p. The penalization factor is typically set to $p = \frac{1}{2}$, which is equal to the probability that a ranking algorithm correctly orders the pair by random guessing, so that there is no advantage to be gained from either random guessing or assigning ties between objects. If $\sigma_P \equiv \sigma_T$, then $K^p(\sigma_P, \sigma_T) \equiv 0$; and if σ_P is exactly the reverse of σ_T, then $K^p(\sigma_P, \sigma_T) \equiv 1$. That is, lower values of K^p indicate better performance.

For preferential TC, one would use K^p on a document-by-document basis, i.e., to measure, given a document d, the difference between how the categories in C are ordered in the true ranking (that is: the ones that most apply to d placed on top of the ranking, and the ones that least apply to d at the bottom of the ranking), and how they are instead ordered in the predicted ranking.

However, a closer analysis reveals that this measure is unsuitable to dealing with situations in which a large number of objects to be ranked are allowed to fall into a much smaller number of "layers" *and* these layers are imbalanced in σ_T. This is the case for 3-layered preferential TC, in which the number of categories that we need to rank is $\gg 3$ (in WIPO-alpha we use 614 categories), and in which the first and second layers (primary and secondary categories, respectively) are much less populated than the third one (non-categories), given that the average document has one primary category, a handful of secondary categories, and hundreds of non-categories. In such cases, ties (in both σ_P and σ_T) obviously tend to be the norm rather than the exception, and a violation involving a tie in one of the overpopulated layers entails a large penalization, as in the following example.

Example 1 Suppose a WIPO-alpha document d_1 is such that its set of primary categories is $P(d_1) = \{c_1\}$, its set of secondary categories is $S(d_1) = \{c_2, c_3, c_4\}$, and its set of non-categories is $N(d_1) = \{c_5,\ldots,c_{614}\}$. Suppose that the only mistake a classifier Φ does is incorrectly deeming c_4 a non-category. This will bring about a cost of $610p$ (since this will generate 610 "false ties" between c_4 and $c_5,\ldots,c_{614}$), and $K^p = .249$. Instead, another classifier Φ' whose only mistake is to incorrectly deem c_1 a secondary category (arguably a more serious mistake) will only bring about a cost of $3p$ (since this will generate 3 "false ties" between c_1 and c_2, c_3, c_4), and $K^p = .001$. Since lower values of K^p are better, Φ' is incorrectly deemed a much better system than Φ.

A second reason why K^p is unsuitable to our case is that it is sensitive only to the relative order of objects, rather than to their having been placed in the correct layer, as in the following example.

[8] The asymmetric character of K^p is due to the fact that it caters for ties (since ties in the true ranking are treated differently from ties in the predicted ranking); "pure" Kendall distance (i.e., with no penalization p) is indeed symmetric, but it assumes that there are ties neither in the predicted nor in the true ranking.

Example 2 Suppose a WIPO-alpha document d_2 is such that $P(d_2) = \{c_1\}$, $S(d_2) = \{\}$, and $N(d_2) = \{c_2,\ldots,c_{614}\}$: a system Φ' which correctly deems c_1 the primary category and incorrectly deems all of $c_2,\ldots,c_{614}$ as secondary categories would bring about a perfect score, i.e., $K^p = 0$.

What Example 2 shows is that preferential TC is fundamentally different from (category) ranking. Guessing the perfect rank is not enough for preferential TC, since one (also) needs to guess, for each category, whether it is above or below the threshold that separates one layer from the other layer. This is not unlike binary classification, in which it is not enough, given a document, to rank the categories according to their estimated degree of relevance to the document. For this reason, none of the evaluation measures currently used (along K^p) in other ranking tasks in IR—such as mean average precision (MAP), precision at position n ($P@n$), normalized discounted cumulative gain (NDCG), etc.—are suitable for preferential TC.

2.2 3-Layered F_1

Below we thus propose an alternative evaluation function consisting of a weighted combination of different F_1 measures. First of all observe that, while a "standard" binary classifier for category c can be evaluated in terms of the standard 4-cell contingency matrix (on which $\pi(c)$ and $\rho(c)$ are based), n-layered preferential TC brings about an n^2-cell such matrix; for instance, in 3-layered preferential TC category c is associated to the 9-cell contingency matrix M^{PSN} of Table 1. From this latter matrix, the three 4-cell sub-matrices M^{PS}, M^{SN}, M^{PN} of Table 2 can be extracted, each of them detailing, with respect to category c, how many documents are correctly placed into or erroneously swapped between two layers t_i and t_j. For each category c we thus have a sub-matrix for first and second layer, one for first and third layer, and one for second and third layer. On each such sub-matrix, $\pi(c)$, $\rho(c)$, and $F_1(c)$ can be computed as usual, although their meaning is clearly changed. For instance, recall as computed on sub-matrix M^{SN} of category c (denoted $\rho^{SN}(c)$, where S and N stand for "secondary category" and "non-category") is computed as $\rho^{SN}(c) = \frac{SS}{SS+NS}$, i.e., as the fraction of documents for which c has been correctly deemed a secondary category, out of the total number of documents for which c is in fact a secondary category and has been deemed either a secondary category or a non-category. The meaning and definition of $\rho^{PS}(c)$, $\rho^{PN}(c)$, and those of $\pi^{PS}(c)$, $\pi^{SN}(c)$, $\pi^{PN}(c)$, $F_1^{PS}(c)$, $F_1^{SN}(c)$, and $F_1^{PN}(c)$, should now be obvious.

We propose *3-layered*F_1 as a measure for evaluating 3-layered TC; this is defined as

Table 1 The 9-cell contingency matrix M^{PSN} for 3-layered preferential TC; P, S, and N stand for "primary category", "secondary category", and "non-category", respectively; PS stands for the number of documents for which c_i is a true primary category and a predicted secondary category; the interpretation of the other double-letter symbols is analogous

c		Predicted		
		P	S	N
True	P	PP	SP	NP
	S	PS	SS	NS
	N	PN	SN	NN

Table 2 The three 4-cell sub-matrices (M^{PS}, M^{SN}, M^{PN}) of the contingency matrix M^{PSN} of Table 1

c		predicted	
		P	S
true	P	PP	SP
	S	PS	SS

c		predicted	
		S	N
true	S	SS	NS
	N	SN	NN

c		predicted	
		P	N
true	P	PP	NP
	N	PN	NN

$$F_1^3(c) = \sum_{i \in \{PS,SN,PN\}} \alpha_i F_1^i(c) \tag{3}$$

i.e., as a linear combination of the three $F_1(c)$ functions computed on the M^{PS}, M^{SN}, M^{PN} sub-matrices. The α_i are to be set depending on the constraints of the application. We propose that their default values should be $\alpha_{PS} = .25$, $\alpha_{SN} = .25$, and $\alpha_{PN} = .50$; these values have the effect of considering erroneous swappings between first and second layer to bring about a cost as high as swappings between second and third layer, but only half as high as erroneous swappings between first and third layer. That is, cost is viewed simply in terms of the distance between the true and the predicted layer of a category. The definitions of $\pi^3(c)$ and $\rho^3(c)$ are completely analogous to that of $F_1^3(c)$.

Of course, several variants of these measures can be used, including ones in which the β parameter of any of the three component F_β measures is not necessarily set to 1; setting β at values different from 1 is well-known to have the effect of emphasizing precision at the expense of recall ($\beta < 1$), or vice versa ($\beta > 1$). For n-layered preferential TC we may analogously define $F_1^n(c)$, a function that depends on the computation of F_1 on $\frac{1}{2}n(n-1)$different submatrixes.

It can be easily seen that F_1^3 does not suffer from the problems from which K^p suffers, as described in Sect. 2.1. For instance, in Example 1 system Φ would receive a score of $F_1^3 = .95$ (resulting from $F_1^{PS} = 1$, $F_1^{SN} = .80$, $F_1^{PN} = 1$) while system Φ' would receive a score of $F_1^3 = .75$ (resulting from $F_1^{PS} = 0$, $F_1^{SN} = 1$, $F_1^{PN} = 1$), i.e., Φ would correctly be deemed a much better system than Φ'. Also, in Example 2, system Φ'' would receive a score of $F_1^3 = .75$ (resulting from $F_1^{PS} = 1$, $F_1^{SN} = 0$, $F_1^{PN} = 1$), correctly stating that Φ'' is far from being the perfect system.

3 Learning algorithms for preferential text categorization

3.1 Two baselines: binary and multiclass SVMs

We concentrate on the problem of predicting, for each document, a *single* primary category and several (possibly zero) secondary categories. We thus defer the problem of predicting several primary and several secondary categories for the same document to future work; anyway, this problem admits solutions similar to the ones presented here.

At first glance, the 3-layered classification problem of attributing a single primary category and a (possibly empty) set of secondary categories to a given test document d seems to have a very simple solution. In fact, one could build a binary classifier for each $c_i \in C$ (by using as positive examples of category c_i all the documents that have c_i either as a primary or as a secondary category) and use the real-valued scores output by each

classifier for d: the category for which the largest score has been obtained would be selected as the primary category, while the set of secondary categories could then be identified by optimizing a threshold for each individual category and selecting the categories whose associated classifier has returned a score above its associated threshold. We have indeed implemented this approach (by using standard binary SVMs); in Sect. 4 this is dubbed "Baseline1".

However, this simple approach has a main drawback, i.e., it does not use the distinction between primary and secondary categories in the training phase. In other words, a training document d' for which category c_i is a primary category has the same impact on the classifier for c_i of another training document d'' for which c_i is a secondary category; we think this is unintuitive.

A stronger approach (dubbed "Baseline2" in Sect. 4) consists in performing two different classification tasks, a first one (by means of a single-label classifier h_P) aimed at identifying the primary category of d, and a second one (by means of a multi-label classifier h_S consisting of m binary classifiers h_S^i, one for each category $c_i \in \{c_1,\ldots,c_m\}$) aimed at identifying, among the remaining categories, the secondary categories of d. The h_P classifier is trained by using, as positive examples of each c_i, only the training documents that have c_i as primary category. Each of the h_S^i is instead trained by using as positive examples only the training documents that have c_i as secondary category, and as negative examples only the training documents that have c_i as non-category (those that have c_i as primary category are discarded).

As an aside, we have also tested a slight variant of Baseline2 in which each of the h_S^i is trained by using as positive examples the training documents that have c_i*either as primary or* as secondary category; this corresponds to using the single-label classifier h_P of Baseline2 and the h_S^i binary classifiers of Baseline1. In our experiments this "Baseline2a" has given inferior results to Baseline2 (see Table 3), and we will thus not discuss this any further.

In our experiments, for generating h_P we use a "multiclass" (i.e., single-label) SVM based on combining binary classifiers, each of them generated through standard binary SVMs, into a Decision Directed Acyclic Graph (DDAG—see Platt et al. 1999, for details,

Table 3 Micro-averaged F_1^3 values obtained by the classifiers

	F_1^{PS}	F_1^{SN}	F_1^{PN}	F_1^3
Baseline1	.8514	.1802	.4823	.4991
Baseline2	**.8861**	.2002	.4642	.5041
Baseline2a	.8837	.1575	.4648	.4927
Ordinal regression (SD)	.7716 (.0378)	.1666 (.0315)	**.5395** (.0202)	.5042 (.0072)
GPLM Adatron[a] (SD)	.8433 (.0238)	.2138 (.0088)	.5129 (.0131)	.5206 (.0026)
GPLM Adatron committee of 3	.8601	.2225	.5155	.5284
GPLM Adatron committee of 5	.8533	.2252	.5190	.5291
GPLM Adatron committee of 10	.8487	.2260	.5220	.5297
GPLM Adatron committee of 15	.8490	**.2262**	.5221	**.5298**
GPLM Adatron committee of 20	.8512	.2252	.5212	.5297

[a] Obtained by averaging on 20 runs

SD, standard deviation

Boldface indicates the best performing system

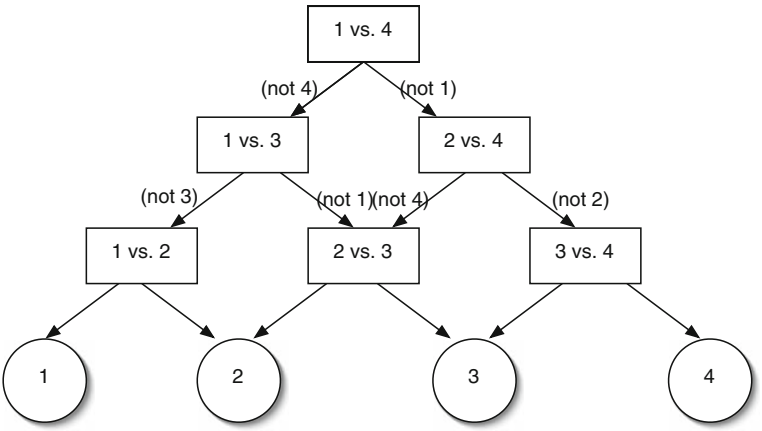

Fig. 1 Scheme of a DDAG-based multiclass SVM, exemplified for a set of four categories {1,2,3,4}. Each box represents a binary classifier c' versus c''. Each circle represents the category which is finally picked for d after a descent in the DDAG

see also Fig. 1). Briefly, a DDAG-based multiclass SVM for categories $\{c_1,\ldots,c_m\}$ generates a rooted, layered binary DAG with the structure of a "Pachinko machine": layer 1 (the root) is placed at the top, the j-th layer ($j \in \{1,\ldots, m-1\}$) contains j internal nodes, down until the m-th (bottom) layer, consisting of m leaves. The i-th node of layer $j < m$ points to the i-th and $(i+1)$-th node of the $(j+1)$-th layer. Each of the $\frac{1}{2}m(m-1)$ internal nodes is associated with a binary classifier, generated by a binary SVM, and in charge of deciding whether or not a category c' is less suitable than another category c'' for test document d. The less suitable category is discarded from consideration; through a recursive descent through the DDAG, categories are repeatedly discarded until only one is left for d. With this method, a total of $\frac{1}{2}m(m-1)$ binary classifiers need to be trained; however, this is less expensive than it might appear at first, since each classifier is trained only with a small subset of examples (namely, the training documents which have either of the two categories as primary category). This makes DDAG-based SVMs more efficient at classification time than other classes of multiclass SVMs, such as those presented in (Crammer and Singer 2001), since the fact that few training documents are used brings about a smaller number of support vectors, which ultimately means that the generated classifiers consist of sparser vectors, hence more efficient at classification time. Note that, while $\frac{1}{2}m(m-1)$ binary classifiers need to be generated at training time, only m need to be invoked at classification time for each test document.

The multi-label classifier h_S is simply formed by m binary classifiers $\{h_S^i: c_i \in \{c_1,\ldots,c_m\}\}$ generated by binary SVMs; see Sect. 4.1 for details on how we have performed parameter optimization.

3.2 Moving further: the generalized preference learning model

The method proposed in the previous section has been obtained by decoupling the problems of finding the primary and the secondary categories of a document. In a sense, the overall problem has been simplified and reduced to two almost independent modules whose predictions are then combined. This approach would be reasonable if the primary

and secondary categories had been attached to the document independently of each other. Unfortunately, in many applicative domains this is not the case. For example, it is plausible to imagine that the set of secondary categories associated to WIPO-alpha patents depends on the primary category. Therefore, a classifier which aims at separating secondary categories from non-categories should have access to the information about the primary category of the document.

As a principled solution of the 3-layered classification problem, we propose the adoption of the generalized preference learning model (GPLM) (Aiolli 2005; Aiolli and Sperduti 2005), a recent framework which generalizes a large class of supervised learning problems by using the notion of preference between categories. The next sections show how this can be adapted to our needs.

3.2.1 The GPLM

The GPLM assumes 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. For each document d the relevance function thus induces a ranking among the categories. A *preference* is a constraint on categories and documents, that should be satisfied by the relevance function. Specifically, GPLM focuses on two types of preferences: (i) *qualitative preferences* $c_i \rhd_d c_j$ ("category c_i applies to document d more than c_j does"), which means that $r(d, c_i) > r(d, c_j)$; and (ii) *quantitative* preferences of type $c \rhd_d \tau$ ("the degree to which category c applies to document d is at least τ", where $\tau \in \mathbb{R}$), which means that $r(d, c) > \tau$.

In this learning framework, supervision for a training document is provided as a set of preferences (of either type). These preferences constitute constraints 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 consistent as possible with these constraints.

As a very simple example of how supervised problems can be modelled in the GPLM let us consider the (single-label) classification problem in which a classifier has to predict the primary category $P(d)$ for a test document d. This case can be modelled by stating, for each training document d', the set of preferences $\{P(d') \rhd_{d'} c_i\}_{c_i \neq P(d')}$. Note that, when classifying a test document d, its primary category will correspond to $\arg\max_{c \in C} r(d, c)$.

As a further example, a multi-label classification problem can instead be modelled by stating, for each training document d', the set of preferences

$$\{c_i \rhd_{d'} \tau\}_{c_i \in C(d')} \cup \{\tau \rhd_{d'} c_j\}_{c_j \in C \setminus C(d')}$$

where τ is a real-valued threshold to be optimized, C is the set of categories, and $C(d')$ is the set of categories to which d' belongs. In this case, the set of categories to which a test document d is deemed to belong are obtained by comparing their associated relevance value to τ: d is deemed to belong to a category c if and only if $r(d, c) > \tau$.

It should be stressed that in GPLM any set of preferences can be associated to a document d, so if no information about the relative ranking of two categories for d is available, no preference involving these two categories need be stated. This allows us to impose on the learner only constraints which are needed.

We may instantiate the GPLM by assuming that the relevance of a document to a category can be expressed in linear form, i.e.

$$r(d, c_i) = \mathbf{w}_i \cdot \mathbf{d} \tag{4}$$

where $\mathbf{d} \in \mathbb{R}^D$ is a (possibly weighted) vectorial representation of d (and D is the size of this vector) and $\mathbf{w}_i \in \mathbb{R}^D$ is a weight vector (containing parameters to be learnt) associated to category c_i. Interestingly, for this case it is possible to give effective algorithms which explicitly attempt to minimize the number of wrong predictions in the training set. In fact, following Eq. 4, qualitative and quantitative preferences can be conveniently reformulated as linear constraints. Specifically, let us consider the qualitative preference $\lambda_1 \equiv (c_i \rhd_d c_j)$. This preference imposes the constraint $r(d, c_i) > r(d, c_j)$ on the relevance function r, which using Eq. 4 can be rewritten as $\mathbf{w}_i \cdot \mathbf{d} > \mathbf{w}_j \cdot \mathbf{d}$, or $(\mathbf{w}_i \cdot \mathbf{d} - \mathbf{w}_j \cdot \mathbf{d}) > 0$. Similar transformations can be done for quantitative preferences.

A uniform treatment of quantitative and qualitative preferences can then be obtained by concatenating all the vectors $\mathbf{w}_i$ (for $i \in \{1,\ldots,m\}$) and all the thresholds $\tau_1,\ldots,\tau_q$ involved in the formulation of the problem, into a single vector $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_m, \tau_1, \ldots, \tau_q) \in \mathbb{R}^{mD+q}$.

In the qualitative case, assuming $i < j$ with no loss of generality, λ_1 can then be further rewritten as

$$\mathbf{w} \cdot \underbrace{(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{i-1}, \mathbf{d}, \underbrace{\mathbf{0},\ldots,\mathbf{0}}_{j-i-1}, -\mathbf{d}, \underbrace{\mathbf{0},\ldots,\mathbf{0}}_{m-j}, \underbrace{0,\ldots,0}_{q})}_{\psi(\lambda_1)} > 0 \tag{5}$$

where $\psi(\lambda_1) \in \mathbb{R}^{mD+q}$ is a representation of λ_1 and $\mathbf{0}$ stands for a vector of all 0's of length D.

In the quantitative case, the preference $\lambda_2 \equiv (c_j \rhd_d \tau_k)$ can similarly be expressed as

$$\mathbf{w} \cdot \underbrace{(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{j-1}, \mathbf{d}, \underbrace{\mathbf{0},\ldots,\mathbf{0}}_{m-j}, \underbrace{0,\ldots,0}_{k-1}, -1, \underbrace{0,\ldots,0}_{q-k})}_{\psi(\lambda_2)} > 0 \tag{6}$$

while preference $\lambda_3 \equiv (\tau_k \rhd_d c_j)$ is expressed as

$$\mathbf{w} \cdot \underbrace{(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{j-1}, -\mathbf{d}, \underbrace{\mathbf{0},\ldots,\mathbf{0}}_{m-j}, \underbrace{0,\ldots,0}_{k-1}, 1, \underbrace{0,\ldots,0}_{q-k})}_{\psi(\lambda_3)} > 0 \tag{7}$$

In general, the training data can then be reduced to a set of linear constraints of the form $\mathbf{w} \cdot \psi(\lambda) > 0$ where $\mathbf{w}$ is the vector of weights and thresholds and $\psi(\lambda)$ is a suitable representation of preference λ. As a consequence, any preference learning problem can be seen as a (homogeneous) linear problem in $\mathbb{R}^{mD+q}$. 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 (i.e., the set of the $\psi(\lambda)$'s is not linearly separable), i.e., such that there is no weight vector $\mathbf{w}$ able to fulfil all the constraints induced by the preferences in the training set. To deal with training errors we may minimize, consistently with the principles of Structural Risk Minimization (SRM) theory (Vapnik 1998), an objective function which is increasing in the number of unfulfilled preferences (the training error) while maximizing the *margin* $1/\|\mathbf{w}\|$ (where $\|\cdot\|$ denotes the 2-norm of a vector). To this end, let us consider the quantity $\rho(\lambda|\mathbf{w}) \equiv \mathbf{w} \cdot \psi(\lambda)$ as the degree of satisfaction of a preference λ given the hypothesis $\mathbf{w}$. This value is greater than zero when the hypothesis is *consistent* with the

preference and smaller than zero otherwise. Now, let us assume a training set $Tr = \{(d_i, \Lambda_i)\}_{i=1,\ldots,n}$, where Λ_i is the set of preferences associated to the i-th document. We aim at minimizing the number of preferences which are unfulfilled (and thus the wrong predictions) on the training set, while trying to maximize the margin $1/\|\mathbf{w}\|_2$. Let $L(\cdot)$ be a convex, always positive, and non-increasing function, such that $L(0) = 1$. It is not difficult to show that the function $L(\rho(\lambda|\mathbf{w}))$ is an upper bound to the error function on the preference λ. Thus, a fairly general approach is the one that attempts to minimize a (convex) function like

$$D(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 + \gamma \sum_{i=1}^{n} \sum_{\lambda \in \Lambda_i} L(\rho(\lambda|\mathbf{w})) \tag{8}$$

where γ is a parameter that determines the relative contributions of the regularization and the loss in the objective function. Note that, if we adopt the *hinge loss* $L(\rho) = [1-\rho]_+ = \max(0, 1-\rho)$, the optimization required is the same as required by a binary SVM with an extended training set consisting of all $\psi(\lambda) \in \mathbb{R}^{mD+q}$ for each preference, each one taken as a (positive) example (see Aiolli 2005 for details).

3.2.2 GPLM mappings for 3-layered classification

We now propose GPLM models for a principled solution of the 3-layered classification task. In the following, we denote by d a document having the set $P(d) = \{c_p\}$ (a singleton) as the set of its primary categories, $S(d) = \{c_{s_1}, \ldots, c_{s_k}\}$ as the (possibly empty) set of its secondary categories, and $N(d) = \{c_{n_1}, \ldots, c_{n_l}\}$ as the set of its non-categories, such that $C = P(d) \cup S(d) \cup N(d)$.

3.2.2.1 GPLM: ordinal regression for 3-layered classification One could be tempted to interpret the 3-layered classification problem as a (multi-variate) ordinal regression (OR) problem, i.e., the problem of attributing a label (here called *rank*) from the ordered set $\{\mathsf{primary}, \mathsf{secondary}, \mathsf{non-category}\}$ to each pair consisting of a document and a category.

Now, we first present a GPLM mapping which can be demonstrated equivalent to the ordinal regression method of (Chu and Keerthi 2007) (see Aiolli 2005 for details). Then we discuss why, in our opinion, this setting does not exactly match the notion of preferential classification. Our experiments, which will be reported in the experimental section, support this claim.

Specifically, for ordinal regression a GPLM model is built by considering two thresholds τ_p and τ_s (see Fig. 2). For each training document the relevance function of a primary category should be above the threshold τ_p, while the relevance function for any other category (either secondary or non-category) should be below τ_p. On the other hand, the relevance function of any secondary category should be above the threshold τ_s, while for any non-category it should be below τ_s. Summarizing, the preference graph for a given training document is as in Fig. 2. As a simple example, consider the set of categories $C = \{c_1, c_2, c_3, c_4, c_5\}$ and a training document d such that $P(d) = \{c_1\}$, $S(d) = \{c_2, c_3\}$, and $N(d) = \{c_4, c_5\}$. The set of preferences we thus generate is

$$\Lambda = \{(c_1 \rhd_d \tau_p), (\tau_p \rhd_d c_2), (\tau_p \rhd_d c_3), (c_2 \rhd_d \tau_s), (c_3 \rhd_d \tau_s), (\tau_s \rhd_d c_4), (\tau_s \rhd_d c_5)\}$$

The standard classification procedure of ordinal regression would select as primary categories the categories that reach a relevance score above τ_p, and select as secondary

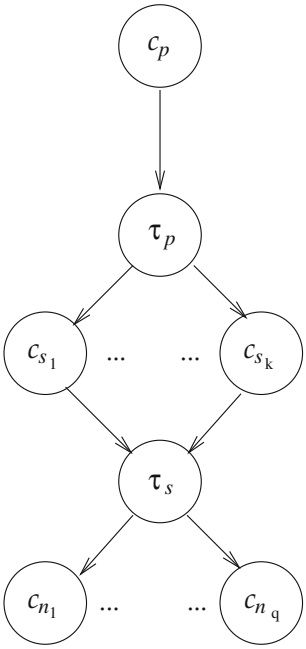

Fig. 2 GPLM mapping for ordinal-regression supervision

categories all the other categories that reach a relevance score above τ_s. However, a classification performed in this way could not guarantee that the predictions are consistent with the requirements of certain preferential tasks. For example, in WIPO-alpha, there is one and only one primary category for each document, and the predictions of ordinal regression as presented above do not necessarily have this property. To overcome this problem in WIPO-alpha, we slightly modify the standard ordinal regression classification procedure by selecting as primary category the category reaching the highest relevance score, and by selecting as secondary categories all the other categories that reach a relevance score above τ_s.

At this point we can discuss the OR-based preference model in more detail. In particular, in (multi-variate) ordinal regression it is assumed that, for each document, the rank of a category is independent from the rank of other categories. This assumption would be reasonable when discriminating between relevant categories (primary or secondary) and non-categories, since this is not a "competitive" decision, but is far less reasonable when one has to choose exactly one (the most relevant) among the relevant categories as the primary category for a document, since in this case we actually have a "competitive" decision. Thus, in this latter case, the choice of the primary category is strongly dependent on which are the relevant categories. This difference is reminiscent of the difference between single-label classification (which is competitive) and multi-label classification (which is not) in multi-class classification tasks. In other words, requiring the relevance score for the primary category to be higher than a given threshold seems an unnecessary constraint, which could eventually lead to deteriorate the overall performance.

3.2.2.2 GPLM: a mapping tailored to 3-layered classification In this section, a variant of the ordinal regression scheme, which seems more suitable for the task of 3-layered

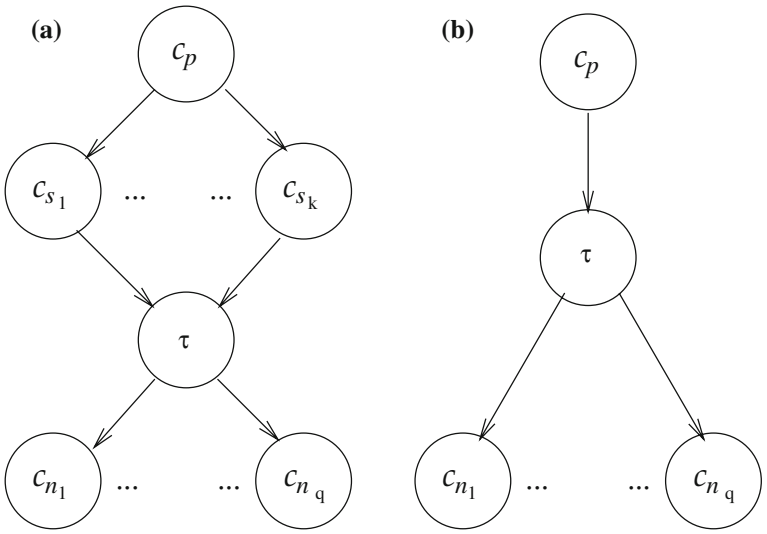

Fig. 3 GPLM mapping for supervision with **a** non-empty secondary category set and **b** empty secondary category set

classification, is presented. In this case, a GPLM model is built as follows. We interpret the primary category as the most relevant among relevant categories. This constraint is introduced by the insertion of a set of qualitative preferences between the primary and all the secondary categories. Moreover, given the multi-label nature of the problem of discerning the secondary categories from the remaining one, a single threshold τ on the relevance scores has to be added between the secondary categories and the non-categories. The categories reaching a relevance score above the threshold (apart from the one recognized as the primary category) will be predicted as secondary categories. See Fig. 3a for a graphical representation of this kind of preference model. Note that whenever $S(d) = \emptyset$, this means that the relevance values for categories in $C \setminus P(d)$ are all below the threshold. To cope with this situation, the qualitative preferences can be collapsed into a direct quantitative preference between the primary category and the threshold. See Fig. 3b for a graphical description of this kind of preference. As a simple example, consider the set of categories $C = \{c_1, c_2, c_3, c_4, c_5\}$ and a training document d such that $P(d) = \{c_1\}$, $S(d) = \{c_2, c_3\}$, and $N(d) = \{c_4, c_5\}$. The set of preferences we generate is

$$\Lambda = \{(c_1 \rhd_d c_2), (c_1 \rhd_d c_3), (c_2 \rhd_d \tau), (c_3 \rhd_d \tau), (\tau \rhd_d c_4), (\tau \rhd_d c_5)\}$$

Similarly, if d is instead such that $P(d) = \{c_1\}$, $S(d) = \emptyset$, $N(d) = \{c_2, c_3, c_4, c_5\}$, this will generate the set of preferences

$$\Lambda = \{(c_1 \rhd_d \tau), (\tau \rhd_d c_2), (\tau \rhd_d c_3), (\tau \rhd_d c_4), (\tau \rhd_d c_5)\}$$

3.2.3 GPLM by using an Adatron-like algorithm

A problem with the direct optimization of Eq. 8 is the huge number of preferences involved. This implies that the naïve solution (e.g., the one that involves training an SVM directly) is impractical. This is especially due to the fact that the number of examples in the extended training set is $O(ne)$, where n is the number of documents in the original training

set and e is the average number of preferences associated to each such documents. Note that in our application e is $O(|C|)$, and $|C|$ can be as high as many hundreds. As a consequence, keeping all the examples in memory may itself be an issue for a common SVM optimization package.

To overcome this problem we propose using a version of the iterative Adatron algorithm (Friess et al. 1998) tailored to our needs. The original Adatron algorithm is capable of efficiently finding the solution of a hard-margin SVM, defined by

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to: } \mathbf{w}\cdot\psi(\lambda_i^j) \geq 1, \quad \forall i \in \{1,\ldots,n\},\ \forall j : \lambda_i^j \in \Lambda_i \end{aligned} \tag{9}$$

by optimizing the dual problem. More formally, let λ_i^j be the jth preference associated to the ith document, and let α_i^j be the associated dual variable; then the optimization problem, as instantiated in our setting, is

$$\begin{aligned} &\max_{\alpha} \sum_{i,j} \alpha_i^j - \frac{1}{2}\sum_{i',j'}\sum_{i'',j''} \alpha_{i'}^{j'}\alpha_{i''}^{j''} k_\epsilon(\psi(\lambda_{i'}^{j'}),\psi(\lambda_{i''}^{j''})) \\ &\text{subject to: } \alpha_i^j \geq 0, \quad \forall i \in \{1,\ldots,n\}, \forall j : \lambda_i^j \in \Lambda_i \end{aligned} \tag{10}$$

where k_ϵ is defined by

$$k_\epsilon(\mathbf{v}_i,\mathbf{v}_j) = \mathbf{v}_i\cdot\mathbf{v}_j + \epsilon\delta_{ij}$$

with δ_{ij} the "Kronecker function", which has value 1 if $i = j$ and 0 otherwise (Cristianini and Shawe-Taylor 2000). It can be shown (Herbrich et al. 2001) that, for all $\epsilon \geq 0$,k_ϵ is indeed a kernel, and the higher the value of ϵ, the more likely the training set will be linearly separable in feature space. Moreover, solving this formulation is equivalent to solving the *soft-margin* SVM with quadratic hinge loss (Vapnik 1998).

The Adatron algorithm requires the update of one single dual variable α_i^j at each iteration of the algorithm. Specifically, the update is made in such a way as to maximize the value of the objective function when all the other variables are fixed. At each iteration, the value of the $\mathbf{w}$ vector can be computed as

$$\mathbf{w} = \sum_{i,j} \alpha_i^j \psi(\lambda_i^j) \tag{11}$$

Note that, since all the terms of the sum are linear, we may directly keep $\mathbf{w}$ in memory.

It can be shown that, given the jth preference of the ith example, i.e., λ_i^j, the update to perform for the associated α_i^j in order to maximize the objective function can be given in closed form as

$$\Delta\alpha_i^j = \frac{1 - \mathbf{w}\cdot\psi(\lambda_i^j) + \epsilon\alpha_i^j}{||\psi(\lambda_i^j)||^2 + \epsilon}$$

In order to keep the solution admissible for the quadratic problem, when $\alpha_i^j + \Delta\alpha_i^j < 0$ the new value of α_i^j is simply put to 0. We can also note that, by construction of the preference representations given in Sect. 3.2.1, $\|\psi(\lambda_i^j)\|^2 = 2\|\mathbf{d}_i\|^2$ whenever λ_i^j is a qualitative preference and $\|\psi(\lambda_i^j)\|^2 = \|\mathbf{d}_i\|^2 + \|\tau\|^2$ whenever λ_i^j is a quantitative preference.

The complete Adatron-like algorithm, instantiated to our setting is given in the following. Recall that, with the GPLM construction discussed earlier, all the preferences can be seen as positive examples.

1. Let $Pr = \{\psi(\lambda_i^j)\}_{i,j}$ be the set of training preferences. Set $\alpha_i^j = 0$ for all i,j, and set $\mathbf{w} = 0$. Set ϵ to a positive value.
2. For all $i \in \{1,\ldots,n\}$ and for all $\lambda_i^j \in \Lambda_i$;
   - Compute the "degree of satisfaction" of $\psi(\lambda_i^j)$ by using the formula $z_i^j \leftarrow \mathbf{w} \cdot \psi(\lambda_i^j) + \epsilon\alpha_i^j$;
   - Compute the current update for the dual variable α_i^j as $\Delta\alpha_i^j \leftarrow \frac{1-z_i^j}{||\psi(\lambda_i^j)||^2+\epsilon}$;
   - If $\alpha_i^j + \Delta\alpha_i^j < 0$, set $\Delta\alpha_i^j \leftarrow -\alpha_i^j$;
   - Compute the new weight vector $\mathbf{w} \leftarrow \mathbf{w} + \Delta\alpha_i^j\psi(\lambda_i^j)$;
   - Update the dual variable $\alpha_i^j \leftarrow \alpha_i^j + \Delta\alpha_i^j$;
3. If the stopping criterion (see Sect. 4.1) in not satisfied, go back to Step 2;

Given that our Adatron-like algorithm is iterative, we can choose not to allow the training to converge, and consider instead the number of iterations as a further parameter which can be optimized by validating on a hold-out sample, similarly to what is done for the ϵ parameter. This seems reasonable because (i) the GPLM loss function that our Adatron-like algorithm minimizes is different from F_1^3, which means that minimizing the former is anyway suboptimal for our needs, and (ii) given the enormous number of preferences, this procedure can potentially bring about a huge speedup for training (in fact, as we will see empirically, this "early-stopping" strategy leads to stopping the algorithm after very few iterations).

The Adatron-like algorithm as presented above eventually converges to the same solution independently from the order of presentation of the preferences in Step 2. However, when the algorithm is stopped far before complete convergence, the produced models may still be immature. This implies that runs of the algorithm performed with different orders of presentation of the preferences can produce different models and, more importantly, they can generate different kinds of errors. In these cases, machine learning theory tells us that an improvement can be obtained by taking these diverse "good" classifiers and combining them into a single one, for example by averaging. Note that, since the classifiers $\mathbf{w}_k$ are linear, the computation of the average over K classifiers is obtained as $\hat{\mathbf{w}} = \frac{1}{K}\sum_k \mathbf{w}_k$.

4 Experiments

4.1 Experimental setting

We have evaluated our algorithms on the WIPO-alpha dataset, a large (3GB) collection published by the World Intellectual Property Organization (WIPO) in 2003. The dataset consists of 75,250 patents classified according to version 8 of the International Patent Classification scheme (IPC—see Footnote 1). Each document d has one primary category (known as the *main IPC symbol* of d), and a variable (possibly null) number of secondary categories (the *secondary IPC symbols* of d). The IPC scheme consists in a four-level hierarchy comprising 8 "sections" (1st level—the root coincides with level 0), 120 "classes" (2nd), 630 "subclasses" (3rd), and about 69,000 "groups" (4th). A typical category might be "D05C 1/00", which is to be read as Section D (Textiles; Paper), Class 05 ("Sewing; Embroidering; Tufting"), Sub-class C ("Embroidering; Tufting") and Group 1/00 ("Apparatus, devices, or tools for hand embroidering"). In order to avoid problems due to excessive sparsity, and consistently with previous literature (Fall et al. 2003), we

only consider categories at the subclass level; each of the 630 IPC subclasses is thus viewed as containing the union of the documents contained in its subordinate groups.

Among the 614 IPC subclasses that are attached to at least one WIPO-alpha training document, either as a primary or a secondary category, only 451 appear as primary category for some training document; other 163 appear in the training set only in the role of secondary category. This poses an interesting problem in evaluation: should we test the system on the first 451 categories only, or should we also consider the latter 163? Our parameters for predicting secondary categories have sometimes (e.g., in Baseline1) been *collectively* optimized on all 614 categories, so using only the first 451 would penalize these approaches. However, the classifier for primary categories in Baseline2 is a 1-of-451 classifier. If we evaluated on all the 614 categories, it would necessarily misclassify a test document whose true primary category is one of the remaining 163 categories; and we cannot ask a system to predict a concept for which we provide no training data. As a consequence, we consider the first 451 categories the potential primary categories, and all the 614 categories the potential secondary categories, consistently with what our training set tells us. This means that, if a system predicts as primary category a category not in the set of 451 potential primary categories (this can be the case for both Baseline1 and our GPLM system), we simply treat this as a predicted secondary category, and at the same time treat as a predicted primary category the top-scoring predicted secondary category which is in the set of the 451 potential primary categories.

WIPO-alpha comes partitioned into a training set *Tr* of 46,324 documents and a test set *Te* of 28,926 documents. Each category appears as *primary* category in at least 20 and at most 2,000 training documents, and of at least 10 and at most 1,000 test documents; each category appears as *secondary* category in at least 1 and at most 1,857 training documents, and of at least 1 and at most 1,621 test documents. The percentage of documents which are associated to at least one secondary category is 34% in the training set and 33% in the test set; most documents have thus no secondary categories attached. Categories that are attached as secondary categories to at least ten documents are 62% of the total set of categories for the training set and 39,7% for the test set.

In our experiments we use the entire WIPO-alpha set of 75,250 documents. Each document includes a title, a list of inventors, a list of applicant companies or individuals, an abstract, a claims section, and a long description. Similarly to Fall et al. (2003) we have only used the title, the abstract, and the first 300 words of the "long description."[9] Preprocessing has been obtained by performing stop word removal, punctuation removal, down-casing, number removal, and Porter stemming. Vectorial representations have been generated for each document by the well-known "ltc" variant of cosine-normalized *tfidf* weighting.

For both baselines we have used "soft-margin" SVMs (in Thorsten Joachims' SVM-light implementation[10]) with a linear kernel; this has thus required us to optimize the c parameter, which determines the tradeoff between the complexity of the generated model and its training error.

In training the baseline classifiers, the ordinal regression classifier, and our GPLM-based classifier, we have performed thorough parameter optimization. For Baseline1, the

[9] Fall et al. (2003) choose this setting due to the fact that the long description of the patent is a *very* long text that describes the invention at a level of detail largely irrelevant to the purposes of classification. Among other things, using the entire long description would bring about a feature set with more than half a million features.

[10] http://www.svmlight.joachims.org/.

validation process was performed by selecting a unique value of c for all the 614 binary SVMs; this is due to the fact that the overall prediction calculated by Baseline1 depends upon all the classifiers associated to the classes. It follows that, allowing for different choices of c over individual SVMs, let say q different values, would have required an exponential number, namely q^{614}, of different classifiers to train.

For Baseline2, validation was performed independently on the single-label (h_P) and on the multi-label (h_S) classifiers. For producing h_P we have generated a DDAG with $\frac{1}{2}(451 * (451-1)) = 101,475$ binary classifiers, each from 1,027 positive training examples per category on average. Since 101,475 is a very high number, we have chosen to optimize c only once for all binary classifiers. In order to do this, we have divided the training set into a validation set (Va) and a "true" training set ($Tr - Va$), obtained by attributing 70% of the positive training examples of each category to $Tr - Va$ and the remaining 30% to Va. The classifiers generated from $Tr - Va$ were organized into a DDAG-based single-label classifier, which was then evaluated on Va in terms of *accuracy* (the percentage of documents that have been correctly classified).[11] As values of c we have tested 10^i with $i \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$; the best result we obtained on Va (48.62%) was for $i = 1$.

For the multi-label classifier h_S of Baseline2 the validation process was performed by selecting a possibly different value of c for each of the 614 binary SVMs. We fixed the threshold to 0, so that all the categories whose associated classifier attributes a score higher than 0 to d are attached to d as secondary categories. For each category c_i, 70% of the training documents were used as "true" training examples and the other 30% as validation examples. The values of c we tested are the same as those used for optimizing h_P. The classifiers generated from the $Tr - Va$ were evaluated on Va in terms of F_1.

Validation for the ordinal regression and the GPLM classifiers was instead aimed at optimizing the ϵ parameter, along with the number of iterations, in our Adatron-like algorithm. The training set, which generates 28,859,852 preferences when using the approach described in Sect. 3.2.2, was divided into a true training set Tr and a validation set Va as for the h_S classifier. Model selection was performed by testing all values of (i) the parameter ϵ in $\epsilon \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ and (ii) the number of iterations t in the range $\{1,\ldots,30\}$, and choosing the best-performing values.

The overall validation procedure for both the individual GPLM models and the averaged model can be summarized as follows:

1. For $t \in \{1,\ldots,30\}$, for $\epsilon \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$, and for $k = 1,\ldots,K$ do
   (a) Fix an order o_k of presentation of the preferences;
   (b) Run t iterations of the Adatron algorithm with parameter ϵ over the set $Tr - Va$, apply the resulting classifiers on Va and compute the value of F_1^3, denoting it by $V_k(t, \epsilon)$);
2. For each pair $((t, \epsilon)$compute the average performance $\hat{V}(t, \epsilon) = \frac{1}{K}\sum_{k=1}^{K} V_k(t, \epsilon)$;
3. Return $(\hat{t}, \hat{\epsilon}) = \arg\max_{(t,\epsilon)} \hat{V}(t, \epsilon)$;

Once determined the "optimal" parameters $(\hat{t}, \hat{\epsilon})$ as described above, $K = 20$ runs of $\hat{t}$ iterations of the Adatron algorithm with parameter $\hat{\epsilon}$ are executed over the whole training set. As proposed in Sect. 3.2.3, the obtained models are also averaged to form a classifier committee. In our case, we obtained $\hat{t} = 3$ and $\hat{\epsilon} = 0.01$.

[11] For single-label classification it is well-known that micro-averaged precision, micro-averaged recall, micro-averaged F_1, and accuracy have the same value.

Finally, note that we do not compare our results to the ones previously obtained by other researchers on WIPO-alpha, since

- these authors work on tasks different from preferential classification, i.e., single-label classification in Cai and Hofmann (2004), Fall et al. (2003), Hofmann et al. (2003), Rousu et al. (2006), Tikk and Biró (2003), Tsochantaridis et al. (2004), Vishwanathan et al. (2006) and multi-label classification in Cai and Hofmann (2007), Tikk et al. (2004), and
- only few of them (Fall et al. 2003; Rousu et al. 2006; Tikk and Biró 2003) test their systems on the full set of 75,250 WIPO-alpha documents. For instance, Altun et al. (2007), Hofmann et al. (2003), Seeger (2007), Shahbaba (2007), Tikk et al. (2004), Tsochantaridis et al. (2004), Vishwanathan et al. (2006) only use 1,710 documents, while Cai and Hofmann (2004, 2007) only use 9,406.

4.2 Results

The results obtained for the different classifiers are summarized in Table 3.

The first three rows report the performance of the two baseline classifiers and the modified version of the Baseline2 classifier as described in Sect. 3.1. It can be observed that the fist two have almost identical F_1^3. Both baselines are good in telling apart primary from secondary categories. This is especially true for Baseline2, and it can be explained by recalling that the single-label classifier that selects the primary category is trained in such a way that both secondary and non-categories are considered as not relevant. Thus, since the number of secondary categories is much smaller that the number of non-categories, it is more likely to wrongly predict as primary a non-category instead of a secondary category. As a consequence of this, there will only be few cases in which a secondary category is deemed a primary category, thus improving F_1^{PS}, while there it will be more frequently the case that a non-category is predicted as a primary category, thus worsening F_1^{PN}. Also Baseline1 is not very good in telling apart secondary categories from non-categories (F_1^{SN}).

The fourth row reports the performance of the ordinal regression classifier, which turns out to have the best separation between primary categories and non-categories (F_1^{PN}) but a low performance on separating primary and secondary categories (F_1^{PS}). These results seem coherent with the analysis we have given in Sect. 3.2.2.1 as the separation between primary categories and non-categories is over-constrained by the ordinal regression model. The overall performance (F_1^3) is roughly equal to that of the baseline classifiers.

The fourth row reports the average performance of the GPLM Adatron over 20 different runs (which differ for the order of presentation of the preferences). Standard deviation is also reported to show that there is not a severe dependence of the performance on the order of presentation of the preferences. With respect to the baselines and the ordinal regression classifier, there is a clear improvement on F_1^{SN}, while F_1^{PS} decreases. Overall, however, there is a significant improvement in F_1^3. The remaining rows report the performances obtained by committees using different number of members. The members are the classifiers obtained by the 20 runs described above. The committee with i members is composed of the classifiers obtained by the first i runs. It can be noted that F_1^3 increases with the number of members in the committee, however adding more members to a committee of ten does not give a significant improvement.

5 Conclusions

We have addressed the problem of how to learn a classifier that distinguishes between the primary and the secondary categories of a document, and argued that this task deserves to be explicitly tackled by TC research.

The first problem in dealing with this novel task is how to evaluate the performance of a classifier. We observed that already known evaluation measures defined either on standard categorization tasks or on ranking tasks, have drawbacks. We have thus proposed an F_1-based evaluation measure in which different kinds of misclassifications involving either primary or secondary categories have a different impact.

Then, by using state-of-the-art learning technology such as multiclass SVMs (for detecting the unique primary category) and binary SVMs (for detecting the secondary categories), we have established strong baseline results for this task on a patent classification dataset in which the distinction between primary and secondary categories is present. In addition, we have provided a slightly better solution based on an ordinal regression model implemented by a recently proposed class of learning algorithms explicitly geared to learning from training data expressed in preferential form, i.e., in the form "for document d_i, category c' is preferred to category c''".

Finally, we have shown that it is possible to improve on the baselines and on the ordinal regression classifier, by defining a preferential model according to the true nature of the problem, i.e., where the primary category is in competition with the secondary categories. Thanks to this approach we have been able to give proper treatment to primary and secondary categories not only in the testing phase but also in the learning phase. The proposed learner is incremental, improves its performance rapidly with the learning iterations, and generates a single model per class.

In the future we plan to investigate preferential text classification further by tailoring the training loss function used in the GPLM to the specific measure eventually used for evaluating the results of classification ($F^3{}_1$, in our case). Note in fact (see Sect. 3.2.1) that we have not made any attempt at selecting a training loss function that closely fits the chosen evaluation measure, since in our case the former only measures the number of unfulfilled preferences, without distinguishing *which kind* of preferences are unfulfilled. Aiming at a better such fit might turn out to improve effectiveness since, as reflected in the different values that the α_i parameters of Equation 3 may typically be given, different types of misclassification may have a different impact on the computed performance.

Acknowledgements We thank Tiziano Fagni for indexing the WIPO-alpha collection and Andrea Esuli for useful discussions on Kendall distance. Thanks also to Lijuan Cai, Shantanu Godbole, Juho Rousu, Sunita Sarawagi, Domonkos Tikk, and S. Vishwanathan for clarifying the details of their experiments. This work has been partially supported by the project "Tecniche di classificazione automatica per brevetti", funded by the University of Padova.

References

Aiolli, F. (2005). A preference model for structured supervised learning tasks. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)* (pp. 557–560). Houston, USA.

Aiolli, F., & Sperduti, A. (2005). Learning preferences for multiclass problems. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems 17* (Vol. 17, pp. 17–24). Cambridge, MA: MIT Press.

Altun, Y., Hofmann, T., & Tsochantaridis, I. (2007). Support vector machine learning for interdependent and structured output spaces. In: G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, & S. V. N. Vishwanathan (Eds.), *Predicting structured data* (pp. 85–104). Cambridge, MA: The MIT Press.

Cai, L., & Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM'04)* (pp. 78–87). Washington, DC.

Cai, L., & Hofmann, T. (2007). Exploiting known taxonomies in learning overlapping concepts. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)* (pp. 714–719). Hyderabad, India

Chu, W., & Keerthi, S. S. (2007). Support vector ordinal regression. *Neural Computation, 19*(3), 792–815.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research, 2*, 265–292.

Crammer, K., & Singer, Y. (2002). A new family of online algorithms for category ranking. In *Proceedings of the 25th ACM International Conference on Research and Development in Information Retrieval (SIGIR'02)* (pp. 151–158). Tampere, FI

Cristianini, N., & Shawe-Taylor J. (2000). *An introduction to support vector machines*. Cambridge, UK: Cambridge University Press.

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., & Vee, E. (2004). Comparing and aggregating rankings with ties. In *Proceedings of ACM International Conference on Principles of Database Systems (PODS'04)* (pp. 47–58). Paris, France

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., & Vee, E. (2006). Comparing partial rankings. *SIAM Journal on Discrete Mathematics, 20*(3), 628–648.

Fall, C. J., Törcsvári, A., Benzineb, K., & Karetka, G. (2003). Automated categorization in the international patent classification. *SIGIR Forum, 37*(1), 10–25.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research, 3*, 1289–1305.

Friess, T.-T., Cristianini, N., & Campbell, C. (1998). The Kernel-Adatron: A fast and simple learning procedure for support vector machines. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)* (pp. 188–196). Madison, USA.

Geng, X., Liu, T.-Y., Qin, T., & Li, H. (2007). Feature selection for ranking. In *Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval (SIGIR'07)* (pp. 407–414). Amsterdam, The Netherlands.

Herbrich, R., Graepel, T., & Campbell, C. (2001). Bayes point machines. *Journal of Machine Learning Research, 1*, 245–279.

Hersh, W., Buckley, C., Leone, T., & Hickman, D. (1994). Ohsumed: An interactive retrieval evaluation and new large text collection for research. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR'94)* (pp. 192–201). Dublin, Ireland.

Hofmann, T., Cai, L., & Ciaramita, M. (2003). Learning with taxonomies: Classifying documents and words. In *Proceedings of the NIPS'03 Workshop on Syntax, Semantics, and Statistics*. Vancouver, Canada.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)* (pp. 137–142). Chemnitz, Germany.

Lam, W., & Ho, C. Y. (1998). Using a generalized instance set for automatic text categorization. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)* (pp. 81–89). Melbourne, Australia.

Lewis, D. D. (1995). Evaluating and optmizing autonomous text classification systems. In *Proceedings of 18th ACM International Conference on Research and Development in Information Retrieval (SIGIR'95)* (pp. 246–254). Seattle, USA.

Lewis, D. D., Schapire, R. E., Callan, J. P., & Papka, R. (1996). Training algorithms for linear text classifiers. In *Proceedings of the 19th ACM International Conference on Research and Development in Information Retrieval (SIGIR'96)* (pp. 298–306). Zürich, Switzerland.

Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (1999). Large-margin DAGs for multiclass classification. In *Proceedings of the 11th International Conference on Neural Information Processing Systems (NIPS'99)* (pp. 533–547). Denver, USA.

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research, 7*, 1601–1626.

Ruiz, M., & Srinivasan, P. (2002). Hierarchical text classification using neural networks. *Information Retrieval, 5*(1), 87–118.

Seeger, M. W. (2007). Cross-validation optimization for large scale hierarchical classification Kernel methods. In B. Schölkopf, J. Platt, & Hoffman, T. (Eds.), *Advances in neural information processing systems* (Vol. 19, pp. 1233–1240). Cambridge, MA: MIT Press.

Shahbaba, B. (2007). Improving classification models when a class hierarchy is available. Ph.D. thesis, Graduate Department of Public Health Sciences, University of Toronto, Toronto, Canada.

Tikk, D., & Biró, G. (2003). Experiment with a hierarchical text categorization method on the WIPO-alpha patent collection. In *Proceedings of the 4th International Symposium on Uncertainty Modeling and Analysis (ISUMA'03)* (pp. 104–109). College Park, USA.

Tikk, D., Biró, G., & Yang, J. (2004). Experiments with a hierarchical text categorizer. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'04)* (pp. 1191–1196). Budapest, Hungary.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)* (pp. 104–111). Banff, Canada.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Vishwanathan, S., Schraudolph, N., & Smola, A. (2006). Step size adaptation in reproducing kernel Hilbert space. *Journal of Machine Learning Research, 7*, 1107–1133.

Voorhees, E. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)* (pp. 315–323). Melbourne, Australia.

Yang, Y., Zhang, J., & Kisiel, B. (2003). A scalability analysis of classifiers in text categorization. In *Proceedings of the 26th ACM International Conference on Research and Development in Information Retrieval (SIGIR'03)* (pp. 96–103). Toronto, Canada.

Yom-Tov, E., Fine, S., Carmel, D., & Darlow, A. (2005). Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th ACM International Conference on Research and Development in Information Retrieval (SIGIR'05)* (pp. 512–519). Salvador de Bahia, Brazil.