**Title:** ML-KNN: A lazy learning approach to multi-label learning

**Authors:** Min-Ling Zhang, Zhi-Hua Zhou∗

**Year:** 2006

**Published by:** Pattern Recognition

**General Description**

As its name implied, ML-KNN is derived from the popular K-nearest neighbor (KNN) algorithm [7]. Firstly, for each test instance, its KNNs in the training set are identified. Then, according to statistical information gained from the label sets of these neighboring instances, i.e. the number of neigh- boring instances belonging to each possible class, maximum a posteriori (MAP) principle is utilized to determine the la- bel set for the test instance.

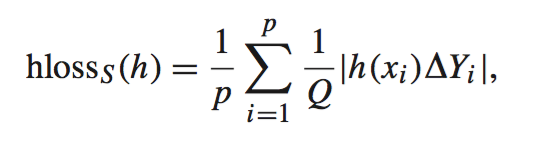
**Preliminaries**

It is supposed that, given an instance xi and its associated label set Yi, a successful learning system will tend to output larger values for labels in Yi than those not inYi,i.e.f(xi,y1)>f(xi,y2)for any y1 ∈Yi and y2∈/Yi.

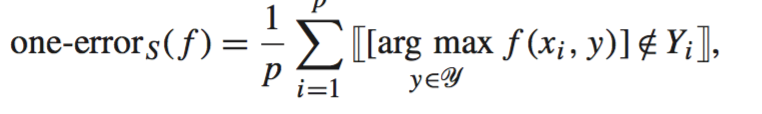
The real-valued function f (·, ·) can be transformed to a ranking function rankf(·,·),which maps the outputs off(xi,y) for any y ∈ Y to {1,2,...,Q} such that if f(xi,y1) > f(xi,y2) then rankf (xi,y1) < rankf (xi,y2).

**Metrics**

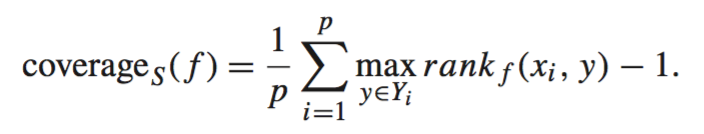
***Hamming loss*:** evaluates how many times an instance label pair is misclassified, i.e. a label not belonging to the instance is predicted or a label belonging to the instance is not predicted. The performance is perfect when hlossS (h) = 0; the *smaller* the value of hlossS (h), the better the performance:



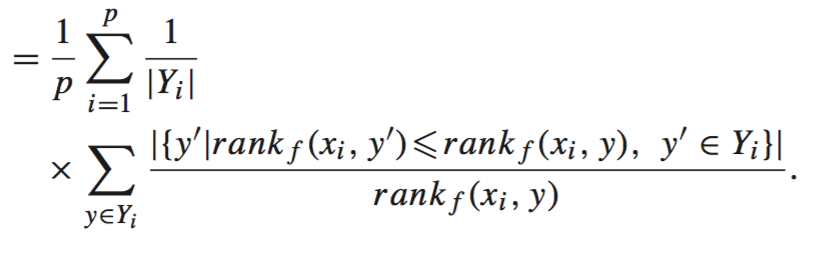
***One-error*:** evaluates how many times the top-ranked label is not in the set of proper labels of the instance. The performance is perfect when one-errorS (f ) = 0; the *smaller* the value of one-errorS (f ), the better the performance:



***Coverage*:** evaluates how far we need, on the average, to go down the list of labels in order to cover all the proper labels of the instance. It is loosely related to precision at the level of perfect recall. The *smaller* the value of coverageS(f), the better the performance:

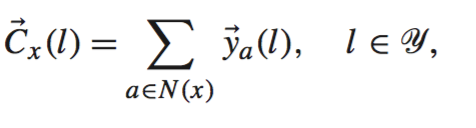


***Average precision*:** evaluates the average fraction of labels ranked above a particular label y ∈ Y which actually are in *Y*. It is originally used in information retrieval (IR) systems to evaluate the document ranking performance for query retrieval [10]. The performance is perfect when avgprecS (f ) = 1; the *bigger* the value of avgprecS (f ), the better the performance:

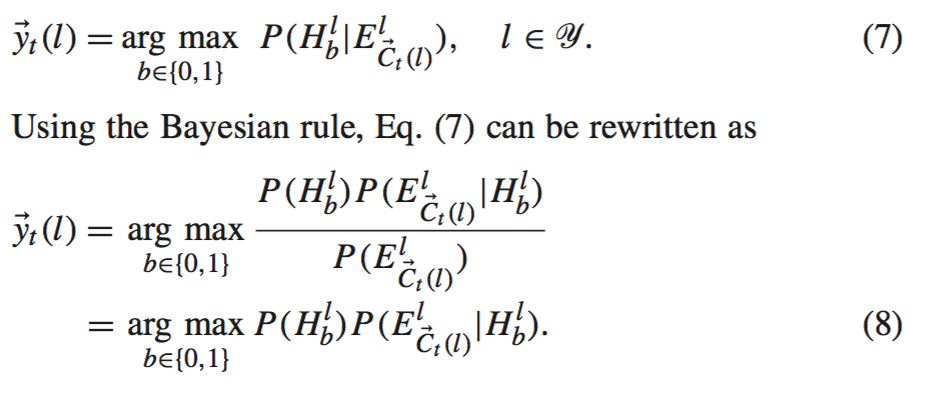


**ML-Knn:**

Given an instance *x* and its associated label set Y ⊆ Y, suppose KNNs are considered in the ML-KNN method. Let y⃗x be the category vector for *x*, where its *l*th component y⃗x(l)(l∈Y)takes the value of 1 if l∈Y and 0 otherwise.In addition, let N(x) denote the set of KNNs of *x* identified in the training set. Thus, based on the label sets of these neighbors, a *membership counting* vector can be defined as:



For each test instance *t*, ML-KNN firstly identifies its KNNs N(t) in the training set. Let H1l be the event that *t* has label *l*, while H0l be the event that *t* has not label *l*. Furthermore, let Ejl (j ∈ {0, 1, . . . , K}) denote the event that, among the KNNs of *t*, there are exactly *j* instances which have label *l*. Therefore, based on the membership counting vector C⃗t , the category vec- tor y⃗t is determined using the following MAP principle:

****

**Dataset**

Web pages linked from the “yahoo.com” domain, where it consists of 14 top-level cate- gories (i.e. “Arts & Humanities”, “Business & Economy”, etc.) and each category is classified into a number of second-level subcategories. By focusing on the second-level categories, they used 11 out of the 14 *independent* text categorization problems. For each problem, the training set contains 2000 documents while the test set contains 3000 documents.

In this paper, these data sets are used to further evaluate the performance of each multi-label learning algorithm. The simple term selection method based on *document frequency* (the number of documents containing a specific term) is used to reduce the dimensionality of each data set. Actually, only 2% words with highest document frequency are retained in the final vocabulary.7 Note that other term selection methods such as *information gain* and *mutual information* could also be adopted. After term selection, each document in the data set is described as a feature vector using the “*Bag-of-Words*” representation, i.e. each dimension of the feature vector corresponds to the number of times a word in the vocabulary appearing in this document.