**THE TERM PROJECT REPORT: An Exploration of Mood Classification in the Million Songs Dataset**

**Section 4:** **Term-Weighting Schemes**

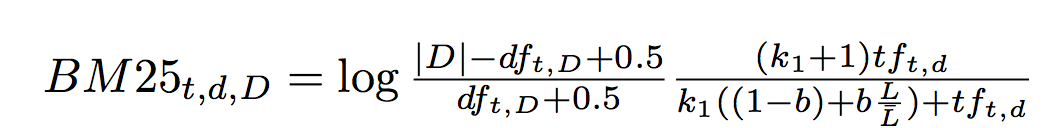
In this study, we represent each document ***d*** as a vector space model. To describe the documents as vectors, we use five term-weighting schemes in Information Retrieval such as *binary approach, term frequency****(tf)***, *term frequency-inverse document frequency****(tf-idf)***, *BM-25* and *delta tf-idf* schemes. We will give brief introductions to these five term weighting schemes.

First of all, *binary approach* is a weighting scheme such as each of the elements of vectors are defined as 1 if it exists or 0 if it does not exist.

Secondly, *term frequency* ***(tf)*** is a weighting scheme such as it counts how many times each term exists in a document. We give higher term frequency for frequently occurring words. Generally we use normalized term frequency as ntf=log(1+tf).

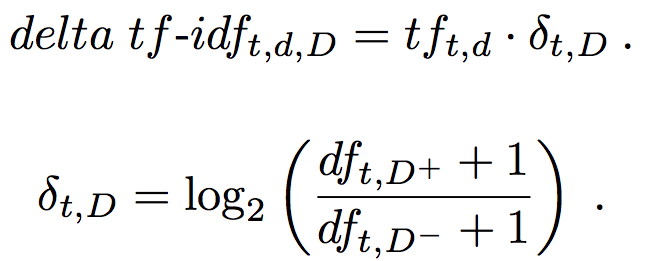
Next, *term frequency-inverse document frequency* ***(tf-idf)*** is another term-weighting scheme such that the formula for it is given as the product of term frequency and inverse document frequency for each term in a document D. And the idf value is given as the logarithm of N (the total number of documents) over df i (number of documents that contain the term i). As opposed to term frequency defined in the previous paragraph, inverse document frequency gives lower weights for frequent words and higher weights for the rare words.

Another term-weighting scheme is a sophisticated one which is *BM-25* which is frequently used in information retrieval. It is computed with the formula:



where L is the document length and Lbar is the average document length in the collection D. In any work, k1 and b are fixed numbers.

The last term-weighting scheme is *delta tf-idf* scheme. It is used for emotion classification with the given formula:

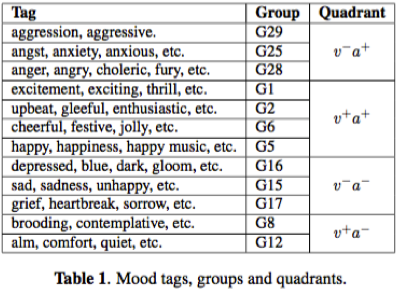


where df+ and df- are document frequencies for any term t in documents labeled as positive and negative respectively. This solution is given by Martineau and Finin to classify documents according to emotions.

**Section 6: Mood Classification**

Using the term weighting explained in the 4th section and granularities (mood quadrants, groups or tags) given in section 3, we can classify songs according to their music moods based on their lyrics with publicly available *Million Song Dataset*.

Experimental Methodology is given as follows: we started to classify songs according to their granularities given below:



We delete duplicates and non english lyrics. Then, we calculate the term distributions of documents and classes with respect to the term-weighting schemes(*a binary term weighting, the term-frequency* ***(tf)****, inverted document frequency* ***(tf-idf),*** *BM25,* and *delta tf-idf)* and use a stemmer to stem the words*.* Also as we mentioned above, we consider each document as a vector.With these calculations, we tried to determine moods of the songs. And classification has been done using the Weka machine learning framework (Weka is a popular suite of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) software written in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), developed at the [University of Waikato](https://en.wikipedia.org/wiki/University_of_Waikato), [New Zealand](https://en.wikipedia.org/wiki/New_Zealand). Because it’s developed in Java language and has files with jar extension, it’s commonly used and plugged in Java projects) with the LIBLinear L2-SVM classification algorithm (Liblinear is popular [open source](https://en.wikipedia.org/wiki/Open_source_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) library, developed at the [National Taiwan University](https://en.wikipedia.org/wiki/National_Taiwan_University) and written in [C++](https://en.wikipedia.org/wiki/C%2B%2B) though with a [C](https://en.wikipedia.org/wiki/C_(programming_language)) API. Liblinear implements [linear](https://en.wikipedia.org/wiki/Linear_classifier) SVMs(support vector machines) and [logistic regression](https://en.wikipedia.org/wiki/Logistic_regression) models trained using a [coordinate descent](https://en.wikipedia.org/wiki/Coordinate_descent) algorithm.). Also this classification’s performance evaluation is calculated by the accuracy function whose formula is given below:

Accuracy=(TP+TN)/(TP+TN+FP+FN) where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

And by making a Kruskal-Wallis[[1]](#footnote-1) test at the level 0.05 we evaluate whether there are significant statistical differences between the term evaluation scheme. And for the final step, we calculate the delta tf-idf values over training set documents and then we apply the results on test set documents.

And the results of the experiment shows that using the term weighting schemes(*a binary term weighting, the term-frequency* ***(tf)****, inverted document frequency* ***(tf-idf),*** *BM25,* and *delta tf-idf)* and mood granularities, we have very similar results in terms of classification accuracy.

1. [↑](#footnote-ref-1)