A Comparative Study on Term Weighting Schemes for Text Classification

Ahmad Mazyad, Fabien Teytaud, Cyril Fonlupt

▶ **To cite this version:**

Ahmad Mazyad, Fabien Teytaud, Cyril Fonlupt. A Comparative Study on Term Weighting Schemes for Text Classification. The Third International Conference on Machine Learning, Optimization and Big Data, Sep 2017, Volterra, Italy. pp.100-108. hal-01662131

HAL Id: hal-01662131
https://inria.hal.science/hal-01662131v1

Submitted on 12 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Comparative Study on Term Weighting Schemes for Text Classification

Ahmad Mazyad, Fabien Teytaud and Cyril Fonlupt

LISIC, Université du Littoral Côte d'Opale,
50 Rue Ferdinand Buisson, 62100 Calais - France

Abstract

Text Classification (or Text Categorization) is a popular machine learning task. It consists in assigning categories to documents. In this paper, we are interested in comparing state of the art classifiers and state of the art feature weights. Feature weight methods are classic tools that are used in text categorization. We extend previous studies by evaluating numerous term weighting schemes for state of the art classification methods. We aim at providing a complete survey on text classification for fair benchmark comparisons.

1 Introduction

Nowadays, at the time of the rapid growth of the internet, the volume of text documents becomes more and more important. Consequently, effective document retrieval may be a really hard task, especially without any organization. Text classification has become a state of the art solution to this problem. Over time, several classification methods appear [5], such as k-nearest neighbor [17], Naïve Bayes [9], decision trees [1], neural networks [10], boosting methods [12] and Support Vector Machines [2]. In this paper we are interested in finding a good term weighting method for state of the art classification algorithms. The paper is organized as follows: in Sect. 2 we present text classification and in particular state of the art term weighting method definitions. In Sect. 3 we present the state of the art classifiers used in our study. In Sect. 4, we compare the different term weighting methods applied to 3 famous text categorization benchmarks. Finally, we discuss and present future works in Sect. 5.

2 Text Classification

Text Classification (TC) aims at automatically assigning a set of predefined categories to a text document. Depending on the text corpus being classified, each document can be in one or multiple categories. This task is achieved by

using a classifier learned on a training set of labeled documents.
A fundamental step in learning a classifier is to represent text documents in a suitable format recognizable by this classifier. In Vector Space Model (VSM), each text document is represented as a vector of index terms in which each term is associated with a weight (score) that measures how informative/discriminative the correspondent term is. The method which assigns a weight to a term is called Term Weighting Scheme (TWS).

To the best of our knowledge, no complete survey exists on how effective TWS performs with different state of the art classifiers. In this paper, we focus on this comparison in order to have a fair and complete study.

One of the most famous TWS is $tf.idf$ proposed by Jones in [13] and stands for term frequency-inverse document frequency. $tf.idf$ is an unsupervised term weighting method. It is the product of the Term Frequency component (TF) by the Collection Frequency component (CF): Term Frequency (tf) and Inverse Document Frequency (idf) respectively. We use the logarithmically scaled tf defined as:

$$tf_{t,d} = 1 + \log(f_{t,d}) \ .$$

where $f_{t \in d}$ stands for the occurrence of term t in the document d. And idf defined as:

$$idf(t) = \log \frac{|D|}{|\{d' \in D | t \in d'\}|} \ .$$

where $|D|$ is the total number of documents and $|\{d' \in D | t \in d'\}|$ is the number of documents that contains the term t.

TC is a supervised learning task, such that document membership (class information) is known in advance. We call Supervised Term Weighting (STW), the term weighting that incorporates the class information. In that context, researchers proposed various supervised term weighting methods that replace the unsupervised collection frequency component idf by a supervised component. For instance, Chi-square (χ^2) is a test of independence between two variables and it was first used as a TWS in text categorization in [6, 4]. Gain ratio (gr) in [4], odds ratio (or) in [6], relevance frequency (rf) was proposed by Lan et al. in [8], inverse category frequency (icf) proposed by Wang et al.in [15], and term relevance ratio (trr) by Youngjoong in [18].

Thus, the general formula for the different TWS in this paper, could be defined as:

$$w_{t,d} = tf_{t,d} \times CF(t) \ .$$

Table 1 shows all the CF included in this study that are used in almost all TC works. All these TWS are used in classic machine learning tools for TC. We present some of these tools in the next section.

CF	Formula
idf	$\log(N/(w+y))$
χ^2	$N \times ((w \times z - x \times y)^2)/((w+y)(x+z)(w+x)(y+z))$
ig	$((w/N) \times \log(w \times N)/((w+x)(w+y))) + ((y/N) \times \log(y \times N)/((y+z)(w+y)))$ $+((x/N) \times \log(x \times N)/((w+x)(b+z))) + ((z/N) \times \log(z \times N)/((y+z)(b+z)))$
gr	$ig/((-(w+y)/N)(\log(w+y)/N) - ((x+z)/N)(\log(x+z)/N))$
or	$\log(2 + (w * z)/(x * y))$
rf	$\log(2 + (w/\max(1, x)))$
icf	$\log_2(C/C_i)$

Table 1: Collection Frequency Components. Given a term t and a category c, N stands for the total number of documents, $|C|$ is the total number of categories and $|C_t|$ is the number of categories where the term t occurs, w is the number of documents that contain t and belong to category c, x is the number of documents that contain t and do not belong to c, y is the number of documents that do not contain t and belong to cj, z is the number of documents that do not contain t and do not belong to c.

3 Classifiers

To study the effect of each Supervised Term Weighting (STW) on classification tasks, we use five known learning algorithms: Support Vector Machine (SVM), Passive-Aggressive (PA), Stochastic Gradient Descent (SGD), Nearest Centroid (NC), C4.5 (C4.5). However it's important to note that we are studying the effectiveness of STW rather than the performance of the learning algorithms.
SVM is a supervised machine learning algorithm used for both classification and regression. SVM has been proposed by Cortes and Vapnik in [2]. Joachims [7] was the first to use SVM for text categorization in which he shows the superiority of SVM over other traditional learning methods.
PA introduced in [3] is an online learning algorithm for large scale dataset. The algorithm watches a stream of instances. Once a new instance is received, the algorithm outputs a prediction. Later, the instance true label is uncovered and the algorithm updates its prediction function.
SGD [19] is another learning algorithm for large scale classification task. It is used to learn linear models such as linear SVM, by mimimizing its objective function.
NC [14] is a simple neighborhood-based classification algorithm. The algorithm computes a centroid for each class. It then outputs the label of the nearest centroid to the test instance as the predicted label.
C4.5 [11] is a supervised tree-based learning algorithm. In 2008, C4.5 has received a considerable amount of attention after being ranked first in the *Top 10 Algorithms in Data Mining* [16].

4 Results and Discussion

4.1 Experiments

Three widely-used datasets are used to evaluate the classifiers: Reuters, Oshumed and 20 Newsgroups.
Reuters-21578[1] is one of the most used test collection for TC research. We use the "ModApte" split which contains 90 categories.
The second dataset is extracted from the Oshumed[2] collection compiled by William Hersh.
The last test collection used in our experiment is the 20 Newsgroups. The dataset "20news-bydate"[3] is sorted by date and splitted into training set (about 60%) and test set (about 40%). Duplicates are removed. Newsgroup-identifying headers (Xref, Newsgroups, Path, Followup-To, Date) are also removed.
In all three test collections, we applied lower case transformation, word stemming and stop word removal. No additional preprocessing steps or feature selection is performed.
Reuters-21578 and Oshumed are multi-labelled datasets. 20Newsgroups is a multi-class dataset. In all cases, we transform the task into multiple binary single label tasks using the one-vs.-all transformation strategy aka one-vs.-rest.
Table 2 shows some statistics on the three collections.

	Reuters	Oshumed	Newsgroups
# documents	7769/3019	6286/7643	11314/7532
# terms	26000	30198	101322
# categories	90	23	20
size of the smallest category	1/1	65/70	377/251
size of the largest category	2877/1087	1799/2153	600/399

Table 2: Statistics on the three test collections (train data/test data).

4.2 Evaluation

To assess the performance of STW, we use the standard F1 measure. The F1 score considers both precision (true positive over true positive plus false positive) p and recall (true positive over true positive plus false negative) r and can be formally defined as: $F1(p,r) = \frac{2rp}{r+p}$. We also report the precision and recall. The precision and recall results of the multiple binary tasks are averaged using the micro-(μ) and macro-(m) averaged measures.
Tables 3, 4, 5, 6, 7, 8, 9, 10, 11 show the μ/m-averaged precision, recall and f-score, for reuters, oshumed and 20newsgroups datasets, respectively. In these tables, the highest μ/m score over a column is underlined, and the best pair of

[1] `http://disi.unitn.it/moschitti/corpora.htm`
[2] `http://disi.unitn.it/moschitti/corpora.htm`
[3] `http://qwone.com/~jason/20Newsgroups/`

Table 3: μ/m-averaged precision results (%) on Reuters-21578 dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	91.50/62.69	94.37/56.75	94.40/56.64	39.22/30.28	82.17/57.17
tfchi2	91.35/63.23	94.37/56.75	94.46/55.54	39.22/30.28	82.18/56.23
tfgr	91.26/61.37	94.37/56.75	94.48/57.21	39.22/30.28	82.44/55.26
tficf	93.21/64.03	94.95/57.31	94.69/61.25	48.87/50.00	81.64/55.34
tfidf	**93.12/64.14**	95.17/56.95	94.45/58.85	63.40/47.57	81.82/56.65
tfig	91.56/62.63	94.37/56.75	94.48/56.68	39.22/30.28	82.45/58.53
tfor	91.73/63.42	94.37/56.75	94.47/56.62	39.22/30.28	82.07/56.24
tfrf	91.51/60.75	94.37/56.75	94.45/55.52	39.22/30.28	81.93/55.63

Table 4: μ/m-averaged recall results (%) on Reuters-21578 dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	82.27/42.74	78.85/33.51	79.73/35.08	**89.93/61.76**	81.62/53.79
tfchi2	81.76/42.64	78.85/33.51	79.62/34.82	**89.93/61.76**	81.41/52.91
tfgr	82.27/41.55	78.85/33.51	79.54/34.81	**89.93/61.76**	81.62/51.81
tficf	79.59/39.44	75.27/30.64	77.19/33.20	86.75/52.96	80.26/51.91
tfidf	82.02/41.81	78.37/33.60	80.02/36.29	87.55/55.60	80.80/53.65
tfig	82.61/41.93	78.85/33.51	79.51/34.81	**89.93/61.76**	81.70/53.51
tfor	82.10/42.59	78.85/33.51	79.46/34.66	**89.93/61.76**	81.68/53.32
tfrf	82.00/41.15	78.85/33.51	79.57/34.75	**89.93/61.76**	81.97/52.97

μ/m scores considering all classifiers and all TWS are bolded. The pair that have the highest average is choosen as the best.

4.3 Results

In tables 3, 4 and 5, we present the μ/m-averaged precision, recall, and f-score, respectively, for Reuters-21578 dataset. In tab.5, NC shows the lowest performance, considering both μ and m scores. PA have the highest μ-score (87.22%) and the second highest m-score (48.51%) preceded only by C4.5 with a m-score of 54.24%. Regarding TWS, even though, $tf.idf$ shows higher scores, the results are very close.

Tables 6, 7 and 8 shows the μ/m-averaged precision, recall, and f-score, respectively, for Oshumed dataset. Considering both precision and recall scores in 8, PA shows the best performance, followed by SGD, SVM. Strangely C4.5 shows the lowest performance.
Regarding TWS, $tf.or$ outperforms clearly all other methods except when used in conjunction NC. $tf.rf$, $tf.gr$ and $tf.ig$ have close results, and come second, followed by tf and $tf.idf$. $tf.icf$ performs poorly.
For these two datasets, we can note that, in comparison with the other algo-

Table 5: μ/m-averaged f-score results (%) on Reuters-21578 dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	86.64/48.48	85.91/39.74	86.45/41.15	54.61/34.75	81.90/53.63
tfchi2	86.29/48.51	85.91/39.74	86.41/40.80	54.61/34.75	81.79/53.24
tfgr	86.53/47.14	85.91/39.74	86.37/41.14	54.61/34.75	82.03/51.79
tficf	85.87/46.42	83.97/37.77	85.05/40.28	62.52/46.43	80.94/52.05
tfidf	**87.22/48.20**	85.95/40.32	86.64/42.73	73.55/47.05	81.31/53.36
tfig	86.86/47.76	85.91/39.74	86.35/40.97	54.61/34.75	**82.08/54.24**
tfor	86.65/48.48	85.91/39.74	86.32/40.85	54.61/34.75	81.87/52.82
tfrf	86.49/46.74	85.91/39.74	86.37/40.76	54.61/34.75	81.95/52.82

Table 6: μ/m-averaged precision results (%) on Oshumed dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	71.13/73.55	78.77/81.13	79.85/82.42	39.22/35.64	57.09/53.40
tfchi2	64.72/61.40	72.81/71.56	71.57/69.70	47.34/45.23	58.22/56.02
tfgr	76.17/78.21	**81.04/80.14**	80.67/79.39	58.65/58.85	56.72/52.89
tficf	74.27/75.04	80.80/81.07	77.92/77.81	69.32/68.58	55.63/53.09
tfidf	75.76/78.26	80.83/80.36	80.48/79.11	54.40/53.06	57.54/53.61
tfig	76.14/77.84	**81.04/80.14**	80.81/79.49	58.65/58.85	56.84/53.65
tfor	74.25/76.45	79.74/81.91	79.44/81.19	53.58/53.61	57.34/54.62
tfrf	74.08/76.38	80.29/83.20	80.39/82.11	52.24/52.12	57.64/54.63

Table 7: μ/m-averaged recall results (%) on Oshumed dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	52.91/44.32	46.21/35.96	47.27/37.76	**68.04/66.55**	56.08/51.73
tfchi2	56.50/51.22	54.83/48.33	50.77/45.94	64.82/65.07	56.70/52.42
tfgr	54.89/47.80	48.61/40.01	52.08/44.79	66.60/64.62	56.89/52.70
tficf	45.57/40.16	35.50/29.46	42.02/36.32	51.58/47.07	57.43/52.71
tfidf	53.55/45.80	46.82/37.43	50.19/42.00	67.71/65.54	56.35/52.50
tfig	54.76/47.84	48.61/40.01	51.96/44.73	66.60/64.62	57.36/53.67
tfor	58.15/53.84	54.68/48.36	56.53/51.26	66.14/66.29	55.89/51.46
tfrf	56.38/50.68	52.32/44.55	53.69/46.72	66.32/65.98	55.53/51.79

Table 8: μ/m-averaged f-score results (%) on Oshumed dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	60.68/53.95	58.25/47.02	59.39/48.78	49.76/44.48	56.58/52.42
tfchi2	60.33/55.51	62.55/55.27	59.40/52.01	54.72/51.83	57.45/53.88
tfgr	63.80/58.11	60.77/51.71	63.29/56.05	62.37/60.16	56.80/52.65
tficf	56.48/51.32	49.33/41.93	54.60/48.32	59.15/55.25	56.51/52.67
tfidf	62.75/56.42	59.30/49.08	61.83/53.41	60.33/57.43	56.94/52.88
tfig	63.71/58.12	60.77/51.71	63.25/56.02	62.37/60.16	57.10/53.47
tfor	**65.22/62.37**	64.87/58.78	66.05/60.57	59.20/57.43	56.60/52.76
tfrf	64.03/60.08	63.36/55.52	64.38/57.19	58.44/56.05	56.56/53.00

Table 9: μ/m-averaged precision results (%) on 20newsgroups dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	63.91/63.77	66.94/66.58	61.05/62.78	55.91/62.22	44.07/44.12
tfchi2	58.55/60.54	60.26/60.35	59.33/59.51	47.73/60.20	38.20/38.16
tfgr	68.43/68.37	69.69/69.41	**70.19/70.06**	62.85/71.44	43.07/43.37
tficf	68.14/68.23	69.15/69.23	69.24/68.85	59.43/71.87	49.19/51.77
tfidf	68.31/68.10	69.69/69.29	61.26/66.59	64.27/69.19	43.65/43.67
tfig	68.97/68.86	70.14/69.79	70.26/69.85	63.64/71.56	44.16/44.40
tfor	68.57/68.19	69.80/69.26	69.54/69.24	56.44/69.03	45.13/44.77
tfrf	56.00/55.42	57.73/56.95	56.57/55.36	36.56/46.22	42.22/42.58

Table 10: μ/m-averaged recall results (%) on 20newsgroups dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	63.91/62.87	66.94/65.81	61.05/59.69	55.91/55.17	44.07/43.01
tfchi2	58.55/57.05	60.26/58.74	59.33/57.82	47.73/47.05	38.20/37.17
tfgr	68.43/67.33	69.69/68.52	70.19/68.92	62.85/62.10	43.07/42.07
tficf	68.14/66.96	69.15/67.90	69.24/67.95	59.43/58.62	49.19/48.15
tfidf	68.31/67.20	69.69/68.48	61.26/59.95	64.27/63.32	43.65/42.73
tfig	68.97/67.86	70.14/68.93	**70.26/68.94**	63.64/62.78	44.16/43.15
tfor	68.57/67.42	69.80/68.52	69.54/68.26	56.44/55.81	45.13/44.05
tfrf	56.00/54.86	57.73/56.38	56.57/55.10	36.56/36.06	42.22/41.33

Table 11: μ/m-averaged f-score results (%) on 20newsgroups dataset using different weighting methods.

	PA	SVM	SGD	NC	C4.5
tf	63.91/63.06	66.94/65.85	61.05/60.38	55.91/56.97	44.07/43.18
tfchi2	58.55/56.89	60.26/58.18	59.33/57.21	47.73/50.62	38.20/36.91
tfgr	68.43/67.55	69.69/68.60	70.19/68.98	62.85/64.45	43.07/42.36
tficf	68.14/67.18	69.15/68.10	69.24/67.97	59.43/61.74	49.19/49.08
tfidf	68.31/67.37	69.69/68.49	61.26/62.38	64.27/64.90	43.65/42.86
tfig	68.97/68.05	70.14/68.96	**70.26/68.93**	63.64/65.09	44.16/43.39
tfor	68.57/67.52	69.80/68.51	69.54/68.30	56.44/59.19	45.13/43.97
tfrf	56.00/54.69	57.73/56.18	56.57/54.47	36.56/38.41	42.22/41.47

rithms, NC have a very high recall scores 4, 7. However, NC reports the lowest precision scores 3, 6.
Scores for Newsgroups dataset are presented in tables 9, 10 and 11. $tf.ig$ and $tf.gr$ record the best scores (70%/69%) in conjunction with both SVM and SGD. Overall, in this dataset, SVM performs the best, followed by SGD and PA. C4.5 records very low scores. As for TWS, $tf.ig$ and $tf.gr$ give the best results, followed closely by $tf.or$, $tf.idf$ and $tf.icf$. $tf.rf$ shows the lowest scores. In constrast to the high recall scores and low precision scores registered by NC algorithm on Reuters-21578 and Oshumed datasets, NC registered approximately equal results on both precision and recall.
Concerning C4.5, we can note that precision and recall results are approximately equal on the three datasets.
Overall, in our study, we find that $tf.or$ is the best TWS. $tf.idf$, $tf.gr$ and $tf.ig$ are also good choices for weighting features. $tf.\chi^2$, $tf.icf$ and $tf.rf$ are the worst methods.

5 Conclusion

The aim of this paper is to give an insight into the different TWS available for TC. These schemes are used in conjunction with five classifers tested on Reutrers-21578, Oshumed and 20newsgroups datasets. Our work aims at extending previous surveys and establishing a clean and fair basis for TC benchmarks.

To sum up, we find that the superiority of supervised term weighting methods over unsupervised methods is still not clear. Even though, in our experiment, $tf.or$ gives better results than $tf.idf$, we find no consistent superiority. In addition, $tf.idf$ is shown to be superior that the three supervised methods $tf.rf$, $tf.\chi^2$ and $tf.icf$. We find also that, alongside with $tf.or$ which gave the best results, $tf.idf$, $tf.gr$ and $tf.ig$ are good choices for weighting features.

References

[1] Apte, C., Damerau, F., Weiss, S., et al.: Text mining with decision rules and decision trees. Citeseer (1998)

[2] Cortes, C., Vapnik, V.: Support-vector networks. vol. 20, pp. 273–297. Springer (1995)

[3] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7(Mar), 551–585 (2006)

[4] Debole, F., Sebastiani, F.: Supervised term weighting for automated text categorization. In: Text mining and its applications, pp. 81–97. Springer (2004)

[5] Deng, Z.H., Tang, S.W., Yang, D.Q., Li, M.Z.L.Y., Xie, K.Q.: A comparative study on feature weight in text categorization. In: Asia-Pacific Web Conference. pp. 588–597. Springer (2004)

[6] Deng, Z.H., Tang, S.W., Yang, D.Q., Li, M.Z.L.Y., Xie, K.Q.: A comparative study on feature weight in text categorization. In: Advanced Web Technologies and Applications, pp. 588–597. Springer (2004)

[7] Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. pp. 137–142. Springer (1998)

[8] Lan, M., Tan, C.L., Su, J., Lu, Y.: Supervised and traditional term weighting methods for automatic text categorization. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31(4), 721–735 (2009)

[9] McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. vol. 752, pp. 41–48. Citeseer (1998)

[10] Ng, H.T., Goh, W.B., Low, K.L.: Feature selection, perceptron learning, and a usability case study for text categorization. In: ACM SIGIR Forum. vol. 31, pp. 67–73. ACM (1997)

[11] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)

[12] Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine learning 39(2-3), 135–168 (2000)

[13] Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of documentation 28(1), 11–21 (1972)

[14] Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proceedings of the National Academy of Sciences 99(10), 6567–6572 (2002)

[15] Wang, D., Zhang, H.: Inverse category frequency based supervised term weighting scheme for text categorization. preprint arXiv:1012.2609v4 (2013)

[16] Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al.: Top 10 algorithms in data mining. Knowledge and information systems 14(1), 1–37 (2008)

[17] Yang, Y.: Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 13–22. Springer-Verlag New York, Inc. (1994)

[18] Youngjoong, K.: A study of term weighting schemes using class information for text classification. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. pp. 1029–1030. ACM (2012)

[19] Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: ICML 2004: PROCEEDINGS OF THE TWENTY-FIRST INTERNATIONAL CONFERENCE ON MACHINE LEARNING. OMNIPRESS. pp. 919–926 (2004)