Scientific Research Publishing

Open Journal of Social Sciences, 2016, 4, 9-18
http://www.scirp.org/journal/jss
ISSN Online: 2327-5960
ISSN Print: 2327-5952

Fine-Grained Detection of Programming Students' Frustration Using Keystrokes, Mouse Clicks and Interaction Logs

Fwa Hua Leong

School of Computing Science, Newcastle University, Newcastle Upon Tyne, UK
Email: h.l.fwa1@newcastle.ac.uk

How to cite this paper: Leong, F.H. (2016) Fine-Grained Detection of Programming Students' Frustration Using Keystrokes, Mouse Clicks and Interaction Logs. *Open Journal of Social Sciences*, **4**, 9-18.
http://dx.doi.org/10.4236/jss.2016.49002

Received: April 28, 2016
Accepted: September 19, 2016
Published: September 22, 2016

Abstract

Prolonged frustration leads to loss of confidence and eventual disinterest in the learning itself. The modelling of frustration in learning is thus important as it informs on the appropriate time to intervene to sustain the interest and motivation of students. To automatically detect learner's frustration in a naturalistic learning environment, the novel use of keystrokes, mouse clicks and interaction patterns of students captured within the context of a tutoring system was proposed. The modelling approach was described and a comparison was made between the proposed model using Bayesian Network and the baseline Naïve Bayes model. With the formulation of an overlapped sliding window mechanism, the granularity of detection was also investigated. The results confirm the hypothesis that a combination of keystrokes, mouse clicks and interaction logs can be used to accurately distinguish affective states of frustration and non-frustration amongst novice learners of computer programming in a granular fashion.

Keywords

Learning, Frustration, Detect, Keystrokes, Programming

1. Introduction

Effective tutoring by an adept teacher is a guided and interactive process where learner's engagement is constantly monitored to provide remedial feedback for sustained learning engagement [1]. This has led to accelerating research on the role of affect in the learning process. Pekrun *et al.* [2] examined academic emotions or emotions that occurred in academic context and concluded that students' engagement and performance correlated closely with academic emotions. Learning occurs when new in-

formation or knowledge is assimilated into the student's existing knowledge schema and this process of attending to and making sense of new knowledge is almost always associated with emotional experiences [3]. A few studies [4] [5] further reinforced that affect or emotions were infused into classroom life and played a critical role in social interaction (both peer to peer and student-teacher), cognitive processing and student engagement.

In a classroom environment, frustration occurs when students are involved in a learning activity that is deemed important but yet the obstacles inherent in the activity cannot be successfully handled [6]. Left unattended, this prolonged frustration will lead to loss of confidence and eventual disinterest in the learning itself. In the study of computer science, attracting and retaining students is a recurring concern in most faculties globally. A common gripe among novice programmers is that programming, an essential component of the computer science syllabus is difficult to master as it involves abstract concepts and tedious debugging and can lead to frustration. The modelling of frustration is thus important as it informs on the appropriate time to intervene to sustain the interest and motivation of students who may otherwise lose confidence and become disillusioned with the learning of the subject.

2. Background

A major challenge with the design of affect sensitive tutoring systems involves the development of computational systems to reliably detect the learner's emotions. Prior studies on the modelling of frustration and other affective states have focused on physical manifestations of the subject e.g. facial expressions [7] [8], eye gaze [9], posture [10] and physiological signals [11] [12]. Most of these sensors though yielding encouraging recognition results, suffer from various issues and constraints when adapted and deployed in a naturalistic learning environment. For example, eye gaze and facial cameras suffer from occlusion and lighting issues while physiological sensors may be overly obtrusive, cumbersome to setup and rarely available for most educational contexts.

On the other hand, keystrokes and mouse clicks are ubiquitous in all classrooms and yet relatively un-explored as a possible sensor for affect recognition. Keystrokes analyses are in fact well researched into as a form of biometrics for user authentication and identification [13] [14] but its potential in affect recognition and detection remains untapped [15]. Some studies [16] [17] employ logs on students' actions or interactions within the tutoring system to detect whether the students are off-task, disengaged or having difficulty with an on hand programming task. The results from these studies confirmed that interaction patterns can be used to detect affect but combining it with other sensors enhances the accuracy of detection [18].

Automatic affect detection is inherently challenging as affect is fuzzy and it is unlikely that two individuals with different personalities or life experiences react uniformly (e.g. by displaying the same facial expression or exhibiting the same bio-physiological attributes) when presented with the same academic problem even though

both are equal in terms of their cognitive abilities. A number of studies seek to overcome the fuzziness in the sensing of emotions by either using posed or induced emotions [19]. This however compromises the accuracy of affect detection when deployed in a naturalistic environment e.g. a computer laboratory as the documented gains would likely not be realized. In this paper, I hypothesized that a combination of keystrokes, mouse clicks and interaction logs can be used to accurately detect frustration of students who are learning computer programming in a naturalistic environment.

Expert human tutors can achieve learning gains of 2 sigma as they are adept at recognizing the affective state of the student and then dynamically adapting their tutoring responses to sustain the student's learning [20]. Thus, to enhance the learning of students, it is vital to respond to the affect of students in a sufficiently timely fashion. Early detection of frustration of students would permit the tutoring systems sufficient time to enact corrective scaffolding actions. It would be futile to intervene if the student has passed the point of no return and has already given up working on the problem totally. On the other hand, it may be impossible to deduce whether a student is frustrated with just a few seconds of captured keystrokes and mouse clicks data. An appropriate level of granularity for affect recognition would have to be established for effective tutoring intervention and this would be investigated in this paper as well.

The rest of the paper is structured as follows: Section II describes the data set used in this work. Section III describes the modelling approach which utilizes the interaction, mouse and keystroke features that are logged during the student's interaction within the tutoring system. Section IV discusses the results and the evaluation of the results and Section V concludes this work.

3. Modelling

3.1. Data

Data from trials conducted in a tertiary institution within Singapore in the year 2014 and 2015 were used in this work. The trials were conducted in computer labs where students were enrolled to work on programming exercises within a Java programming tutoring software. The Java tutoring software is web based and was developed by the author. For each exercise, the students were required to write Java codes and then compiling them online within the tutoring system. The generated output after program compilation would have to match the expected output before the exercise is deemed completed. There are a total of 12 exercises to be completed and the majority of the 24 students who participated in the trials have undergone one term (60 hours) of foundational Java programming course.

To create models that predict students' frustration in a timely and accurate manner, observations of students' interaction with the tutoring system must first be collected. A video of each student's tutoring session which was recorded with a web camera during the tutoring session was replayed by an observer to annotate instances where frustration was observed. Some examples of the frustration behaviours observed in the session include use of expletives, long sighing, excessive gesturing and roughly ruffling through

hair while visibly distressed. Throughout the tutoring session, students' actions, mouse clicks and keystrokes within the tutoring system were continuously captured and written into log files. The annotations by the observer were then temporally aligned with the captured logs. The data were then processed to extract the relevant features. Some of the features aggregated from the input logs include the mean and median key latencies, number of keys, wait time (duration longer than 1 second with no key inputs), back space and delete key latency and frequencies and the frequencies of mouse clicks. The interaction features include the number of compilations, number of errors encountered, number of exercises completed and the duration of time spent working on the exercises.

3.2. Bayesian Network

A Bayesian Network is a directed acyclic graph in which nodes represent domain variables and arcs represent conditional dependencies. It enables an effective representation and computation of the joint probability distribution over a set of random variables [21]. A property of Bayesian Network – each variable is independent of its non-descendants given its parents, is often used to reduce the number of parameters to characterize the joint probability distribution of the variables. This reduction provides for an efficient computation of the posterior probabilities given the evidence. The formula for a Bayesian Network consisting of n nodes with random variables ($x_1, x_2, \cdots, x_n$) is

$$p\left(x_1, x_2, \cdots, x_n\right) = \prod_{i=1}^{n} p\left(x_i \middle| x_{p_i}\right) \tag{1}$$

where $p\left(x_i \middle| x_{p_i}\right)$ is the local probability distribution associated with node i and p_i is the set of indices labelling the parents of node i [22].

Bayesian Network is used to model the occurrence of frustration in this paper as it allows one to see the effects and the degree that the cause (the existence of frustration) has on the effects (keystroke and mouse characteristics). The Bayes Net Toolbox by Kelvin Murphy [23] was used to develop and test the model in Matlab.

3.3. Sliding Window and Different Time Resolution

The features extracted from the students' interaction, keystrokes and mouse logs are aggregated into sliding window sizes of 30 seconds, 60 seconds, 90 seconds, 120 seconds, 150 seconds and 180 seconds with an overlap of 33.3% of the window size. If the time at which frustration is observed falls in the overlap area of 2 consecutive window slices, both window slices would be annotated as slices in which the student experiences frustration. Alternatively, if the time at which frustration is observed falls outside the overlap area, only the time window slice in which it occurs in will be annotated as the slice in which the student experienced frustration. This is illustrated in **Figure 1** below.

Establishing an optimal time window of affect detection is important as it informs on the affective state of students in a timely manner. A larger time window width would mean that students' frustration was left unattended for a long period of time and that increases the risk of losing the students totally. On the other hand, a shorter time

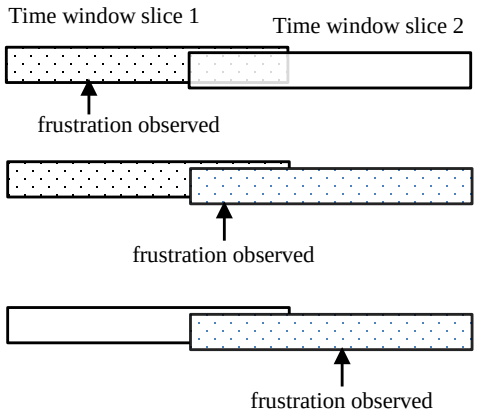

Figure 1. Overlapping time window slices for annotation of frustration.

window width would result in lower accuracy and performance as lesser data is accumulated within the short time window for analysis. Thus, there is a need to evaluate the optimal time window size for the model which balances timely detection of students' affective state with the accuracy of detection.

In cases where keys are sparse for a period, a larger time window width would ensure that more keys are accumulated in a time window which would in turn lead to a more accurate detection of students' affective state. The mean number of keys for each time window slice is shown is **Table 1**. The con of a larger time window width is that the model will need a longer period of time to establish students' affective state.

An observation noted during the trials is that when students are frustrated, the frustration may be manifested in their actions leading up to the observation and that it also persists for a period of time after the observation. To cater for this, the time window slices are overlapped such that if frustration is observed in the overlapped period, both the keystrokes in the time window slice leading up to the observation and the keystrokes in the time window slice after the observation will be included in the evaluation of the affective state (frustrated versus non-frustrated).

3.4. Data Pre-Processing

Innately, different students may type at different speeds even when they are in a neutral affective state. In order to mitigate the effect of the different typing speeds of students, we will have to equalize the keystroke latencies across students. To achieve this, the keystroke latencies for individual students are divided by their individual baseline latency. The baseline latency for each student is derived from the average of the latencies over a sliding window width of 10 keystrokes (after elimination of latencies more than 1 second).

All instances with no keystrokes recorded were eliminated from the data set. By eliminating the instances with no keystrokes, an enhanced classification performance is

Table 1. Mean number of keystrokes by time window sizes.

Time Window (in seconds)	Mean Number of Keystrokes
30	20.74
60	30.87
90	40.79
120	50.88
150	58.16
180	66.84

achieved as compared to the use of the entire data set. To illustrate, for the time window width of 120 seconds, the AUC figure obtained with the exclusion of instances with no keystrokes is 0.81 as compared to 0.54 when all data instances are included. As such, the model can only detect incidences of frustration at an acceptable level of accuracy when there are keystroke activities within the designated time window. To detect incidences of frustration during the period when the students are not typing, it may be necessary to complement the detection with other sensing modes e.g. facial expressions.

3.5. Discretization of Features

The extracted features were discretized using an unsupervised discretization technique—equal frequency binning. Discretization is the conversion of continuous random variables into discrete nominal variables and is a pre-processing step that is commonly utilized in modelling BNs when the continuous random variables do not fit into a Gaussian distribution [24] [25]. The supervised discretization technique—Class Attribute Interdependence Maximization [26] was applied but the results were less satisfactory as compared to that obtained using the Equal Frequency Binning technique. The Equal Frequency Binning discretization technique divides the data into m groups such that each group has the same number of values. The value m = 5 is used in this study.

4. Results

The model was tested using k-fold cross validation methodology with k = 5 folds (in each fold, 4 segments were used for training and 1 segment for testing). K-fold cross validation was employed to minimize over-fitting - the issue of the model having an excellent fit to the training data but yet not fitting well to future unseen data.

To evaluate the performance of the proposed model, it is compared against a baseline (Naïve Bayes) model. The Naïve Bayes model consists of only the class attribute (existence of frustration) as the parent node and assumes that all the other feature variables are conditionally independent given the class attribute. **Table 2** compares the performance of Bayesian Networks against Naïve Bayes models for a 120 seconds time window width. The results show that Bayesian Networks can better discriminate the existence of frustration as compared to Naïve Bayes as both AUC and accuracy of

Bayesian Networks are higher than that of Naïve Bayes by 32.79% and 32.73% respectively. This can be attributed to the fact that the constructed Bayesian Networks structure models the dependencies among the feature variables well.

The classification performance results for the various time window sizes are summarized in **Table 3** and the Receiver Operator Characteristics (ROC) curves for the various time window sizes are listed in **Figure 2**. The ROC graph is a 2 dimensional graph in which sensitivity is plotted against (1-specificity) and it depicts relative trade-offs between benefits (true positive) and cost (false positives) [27]. From **Table 3**, the Area

Table 2. Performance measures of Naïve Bayes and Bayesian Network models.

Performance Measures	Models	
	Naive Bayes	Bayesian Network
AUC	0.61	0.81 (+32.79%)
Accuracy	0.55	0.73 (+32.73%)
Sensitivity	0.79	0.81 (+2.53%)

Table 3. Performance measures for the various time window sizes.

Performance Measures	Time Window (in seconds)					
	30	60	90	120	150	180
AUC	0.70	0.74	0.77	0.81	0.80	0.81
Accuracy	0.79	0.82	0.75	0.73	0.73	0.72
Sensitivity	0.32	0.40	0.65	0.81	0.76	0.76
Specificity	0.83	0.90	0.78	0.68	0.71	0.70

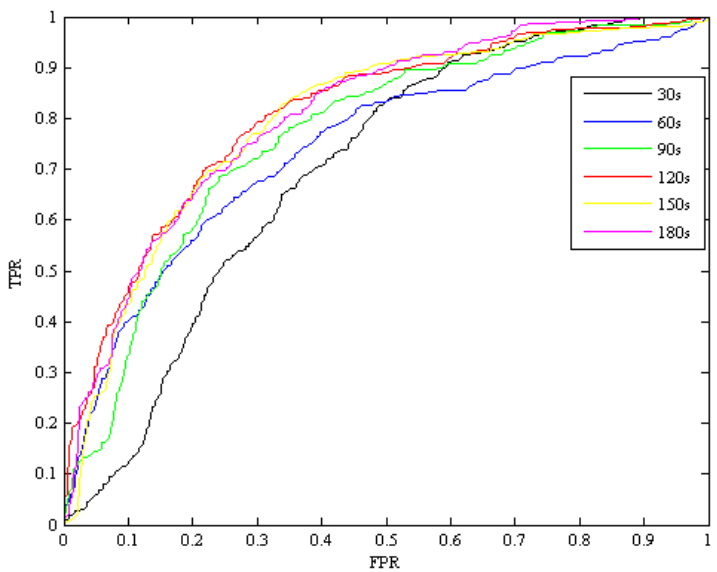

Figure 2. ROC curves for the various time window sizes.

Under the Curve (AUC) and accuracy figures do not differ much across the different time window sizes. The accuracy is defined as the number of correctly identified instances divided by the total number of instances. In situations where the class distributions are highly skewed as they are in our case, accuracy is not a good performance metric for classification. AUC and sensitivity would be more adequate as the performance metric for model evaluation instead. For our model, the best AUC figure is in the 120 seconds time window and accuracy peaks in the 60 seconds time window. For sensitivity, the 120 seconds time window offers the best performance. Sensitivity measures the true positive rate or the proportion of positives that are correctly identified as such while specificity measures the proportion of negatives that are correctly identified as such. A high sensitivity would mean that most of the instances when students are frustrated are detected by the model. A lower specificity is of a lesser concern as for our fail soft scenario of identifying students who are frustrated and responding with strategies to sustain and motivate them in their learning, the repercussions of wrongly labelling students as frustrated when they are not are negligible.

For the time window of 30 seconds, both the AUC of 0.70 and the ROC characteristics depict reasonable performance for frustration detection. By changing the classifier's threshold, we can move to another point on the ROC with a higher sensitivity at the cost of a lower specificity. The concern however is that with the shortened time window, there will be more "holes" – periods with no keystrokes which are eliminated from the data set. It may thus be necessary to make up for these periods of non-detection with other sensing modes.

5. Conclusions

In this paper, I propose the novel use of keystrokes and mouse clicks as sensors and Bayesian Network as the model for naturalistic affect detection in the context of a Java tutoring system. The results showed that keystrokes and mouse clicks characteristics together with the interaction patterns of students can be used in a Bayesian Network model to distinguish between instances of frustration and non-frustration with a high AUC (0.81) and sensitivity (0.81) measure. Comparing the proposed Bayesian Network model to the baseline model using Naïve Bayes, the Bayesian Network model achieved a differential of 32.8% over Naïve Bayes. This confirms the classification performance of the proposed model as compared to baseline. In addition, the risk of overfitting to the data is mitigated with the use of cross validation.

Establishing the appropriate granularity of affect detection is critical as it informs on the affective state of students on a timely fashion so that appropriate remedial actions can be initiated by the tutoring system when students are frustrated with a learning task. In this paper, various detection time window widths are investigated with the formulation of an overlapped sliding window mechanism.

6. Future Extensions to Study

The elimination of time window with no keystrokes constrains the detection of frustra-

tion to only periods where keys are depressed. A possible extension to this study would be to investigate the use of other sensing modes such as facial expressions and eye gaze during the period of time when students are thinking but are not yet ready to attempt the exercises. With the incorporation of other sensing modes, it may also be possible to further reduce the detection time window width for optimal detection accuracy. A reduced time window of detection would allow for the initiation of more timely remedial actions to tackle students' frustration.

References

[1] Merrill, D.C., Reiser, B.J., Ranney, M. and Trafton, J.G. (1992) Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems. *The Journal of the Learning Sciences*, **2**, 277-305. http://dx.doi.org/10.1207/s15327809jls0203_2

[2] Pekrun, R., Goetz, T., Titz, W. and Perry, R.P. (2002) Academic Emotions in Students' Self-Regulated Learning and Achievement: A Program of Qualitative and Quantitative Research. *Educational Psychologist*, **37**, 91-105. http://dx.doi.org/10.1207/S15326985EP3702_4

[3] Stein, N.L., Levine, L.J., Stein, N., Leventhal, B. and Trabasso, T. (1990) Making Sense out of Emotion: The Representation and Use of Goal-Structured Knowledge. *Psychological and Biological Approaches to Emotion*, 45-73.

[4] Kort, B., Reilly, R. and Picard, R.W. (2001) An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy-Building a Learning Companion. *IEEE International Conference on Advanced Learning Technologies,* IEEE Computer Society, 0043. http://dx.doi.org/10.1109/ICALT.2001.943850

[5] Linnenbrink-Garcia, L. and Pekrun, R. (2011) Students' Emotions and Academic Engagement: Introduction to the Special Issue. *Contemporary Educational Psychology*, **36**, 1-3. http://dx.doi.org/10.1016/j.cedpsych.2010.11.004

[6] Pekrun, R. (2006) The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. *Educational Psychology Review*, **18**, 315-341. http://dx.doi.org/10.1016/j.cedpsych.2010.11.004

[7] Bartlett, M.S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I. and Movellan, J. (2006) Fully Automatic Facial Action Recognition in Spontaneous Behavior. *7th International Conference on Automatic Face and Gesture Recognition*, 2006, *FGR* 2006, 223-230.

[8] Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N. and Lester, J.C. (2013) Automatically Recognizing Facial Expression: Predicting Engagement and Frustration. *Proceedings of the* 6*th International Conference on Educational Data Mining*.

[9] Kapoor, A., Mota, S. and Picard, R.W. (2001) Towards a Learning Companion that Recognizes Affect. *AAAI Fall Symposium*, 2-4.

[10] D'Mello, S. and Graesser, A. (2009) Automatic Detection of Learner's Affect from Gross Body Language. *Applied Artificial Intelligence,* **23**, 123-150. http://dx.doi.org/10.1016/j.cedpsych.2010.11.004

[11] Haag, A., Goronzy, S., Schaich, P. and Williams, J. (2004) Emotion Recognition Using Bio-Sensors: First Steps towards an Automatic System. *ADS* Springer, 36-48. http://dx.doi.org/10.1007/978-3-540-24842-2_4

[12] Healey, J.A. and Picard, R.W. (2005) Detecting Stress during Real-World Driving Tasks Using Physiological Sensors. *IEEE Transactions on Intelligent Transportation Systems*, **6**, 156-166. http://dx.doi.org/10.1007/978-3-540-24842-2_4

[13] Dowland, P., Furnell, S. and Papadaki, M. (2002) Keystroke Analysis as a Method of Advanced User Authentication and Response. In: *Security in the Information Society*, Springer, 215-226. http://dx.doi.org/10.1007/978-0-387-35586-3_17

[14] Joyce, R. and Gupta, G. (1990) Identity Authentication Based on Keystroke Latencies. *Communications of the ACM*, **33**, 168-176. http://dx.doi.org/10.1145/75577.75582

[15] Leong, F.H. (2015) Automatic Detection of Frustration of Novice Programmers from Contextual and Keystroke Logs. 2015 10*th International Conference on Computer Science & Education* (*ICCSE*), 373-377. http://dx.doi.org/10.1145/75577.75582

[16] Baker, R.S. (2007) Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1059-1068. http://dx.doi.org/10.1145/1240624.1240785

[17] Carter, J. and Dewan, P. (2010) Design, Implementation, and Evaluation of an Approach for Determining when Programmers Are Having Difficulty. *Proceedings of the* 16*th ACM International Conference on Supporting Group Work*, ACM, 215-224. http://dx.doi.org/10.1145/1880071.1880109

[18] Conati, C. and Maclaren, H. (2009) Empirically Building and Evaluating a Probabilistic Model of User Affect. *User Modeling and User-Adapted Interaction*, **19**, 267-303. http://dx.doi.org/10.1007/s11257-009-9062-8

[19] Scheirer, J., Fernandez, R., Klein, J. and Picard, R.W. (2002) Frustrating the User on Purpose: A Step toward Building an Affective Computer. *Interacting with Computers*, **14**, 93-118. http://dx.doi.org/10.1007/s11257-009-9062-8

[20] Bloom, B.S. (1984) The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, 4-16. http://dx.doi.org/10.3102/0013189X013006004

[21] Pearl, J. (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Reasoning. Morgan Kaufmann Publishers, Los Altos.

[22] Jordan, M.I. (1998) *Learning in Graphical Models. Proceedings of the NATO Advanced Study Institute*, Ettore Mairona Center, Erice, 27 September-7 October 1996, Springer Science & Business Media. http://dx.doi.org/10.1007/978-94-011-5014-9

[23] Murphy, K. (2001) The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, **33**, 1024-1034.

[24] John, G.H. and Langley, P. (1995) Estimating Continuous Distributions in Bayesian Classifiers. *Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 338-345.

[25] Kotsiantis, S. and Kanellopoulos, D. (2006) Discretization Techniques: A Recent Survey. *GESTS International Transactions on Computer Science and Engineering*, **32**, 47-58.

[26] Kurgan, L. and Cios, K.J. (2004) CAIM Discretization Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, **16**, 145-153. http://dx.doi.org/10.1109/TKDE.2004.1269594

[27] Fawcett, T. (2006) An Introduction to ROC Analysis. *Pattern Recognition Letters*, **27**, 861-874. http://dx.doi.org/10.1016/j.patrec.2005.10.010