Obtaining Rubric Weights For Assessments By More Than One Lecturer Using A Pairwise Learning Model

J. R. Quevedo[1] and E. Montañés[2]
[1]Artificial Intelligent Center, Oviedo University
(quevedo@aic.uniovi.es)
[2]Computer Science Department, Oviedo University
(montaneselena@uniovi.es)

Abstract. Specifying the criteria of a rubric to assess an activity, establishing the different quality levels of proficiency of development and defining weights for every criterion is not as easy as one a priori might think. Besides, the complexity of these tasks increases when they involve more than one lecturer. Reaching an agreement about the criteria and the levels of proficiency might be easier taking into account the abilities students must achieve according to the purpose of the subject. However, the disagreement about the weights of every criterion in an assessment rubric might easily appear. This paper focuses on the automatic weight adjustment for the criteria of a rubric. This fitting can be considered as a global perception that the whole group of lecturers have about the accuracy of solving an activity. Firstly, each lecturer makes a proposal of weights and then, from a set of pairs of students he/she globally expresses who of each pair has solved better the activity for which the rubric was designed. Secondly, an approach based on the pairwise learning is proposed in this work to obtain adequate weights for the criteria of a rubric. The system commits fewer errors than the lecturers and makes them improve and reconsider some aspects of the rubric.

1 Introduction

Lots of changes are involved within the new process of convergence to a European Space of Higher Education [12]. The subjects are designed as a set of abilities students must reach. The process encourages a careful inclusion and integration of several methodologies which must point in the right direction in order to guarantee students reach such abilities. Especial emphasis has been made over transversal abilities, such as group working highly demanded by companies from university graduates [1]. Finally, this new paradigm makes new assessment strategies arise to provide reliable information about the skills students in terms of the abilities must reach.

This paper focuses on developing a reliable mechanism to evaluate to what extent students acquire the abilities of a course when more than one lecturer with different perspective is involved in the assessment. Firstly, lecturers have to reach an agreement about what criteria are adequate to consider and then they have to establish minimum thresholds for them. A common strategy for this purpose consists in using evaluation rubrics, which have been increasingly gaining relevance in the last years. They are scoring guides that describe the requirements for various levels of proficiency in the development of certain activity. The purpose is to find out the conditions of success and their degree of fulfillment [11].

The contribution of this paper is a method which engages the variety of preferences every lecturer expresses to contrast and adjust weights to the criteria of a rubric. The challenge consists of including, reproducing and summing up as clearly as possible the global and different perception lecturers have about the achievement of the activities. The hypothesis of departure is that this method leads to reconsider some aspects in the design of a rubric. The approach is based on a pairwise learning system [7] based on Support Vector Machines (SVM). It feeds on information provided by making every lecturer decide between pairs of students who have solved an activity more satisfactorily. The fact that a system uses information of more than one expert (lecturer in this case) was successfully adopted before [2], [3].

2 Evaluation rubrics

Evaluation rubrics [9] can be defined as explicit summaries of the criteria for assessing a particular activity and the different levels of potential achievement for each criterion. There are many features a rubric should fulfill in order to be effective and helpful both for students and lecturers. Then, although it is not an easy task, lecturers must make an effort to carefully design them. One of the requirements must be to reflect the most significant elements related to success in a learning task. Commonly, this expresses the basic skills a student must reach. But they have to provide more information than just a list of goals. It must also enable both students and lecturers to accurately and consistently identify the level of competency of development. This makes students know beforehand what lecturers expect from them and what the characteristics a quality work is required to have. In relation to these conditions, rubrics have to encourage self-assessment of students to become more aware of their own learning process. In this sense, students do not have to wait for the correction of the lecturer to know if the development of the activity is adequate or not [10]. Besides, it helps lecturers to adjust the marks for the activities of the students more accurately and fairly, avoiding discriminatory treatment and adding transparency to the assessment process.

In any case, once the evaluation criteria have been established on the vertical axis of the rubric and the quality levels on the horizontal axis, the rubric must be completed in the sense referred to the weights of every criterion included in the final mark. Table 1 shows a general diagram of a rubric where Q means the quality levels, C corresponds to the criteria, W defines the weight, $N_{i,j}$ contains a description of what a student has been able to do in relation to the i criterion to reach the j level and w_i is the weight of the i criterion.

Table 1 General diagram of a rubric

C	*1*	*2*	...	*Q*	*W*
1st	$N_{1,1}$	$N_{1,2}$	...	$N_{1,q}$	$w_1(\%)$
…	...	...	...	...	...
pth	$N_{p,1}$	$N_{p,2}$	...	$N_{p,q}$	$w_q(\%)$

The weights in the rubric must be comparative. The assignment of a weight to each criterion to obtain the final mark shows the relevance a lecturer grants to it, to which one is assigned more and less relevance and their effect on the final mark.

3 Pairwise learning

In the design of a rubric the marks lecturers grant to two students according to his/her global and comparative perception between both are likely to disagree with the marks resulting from the predefined weights of the rubric. This could happen, for instance, when both students reach a certain level in some criteria but one of them is better in one criterion, for example, in originality or in the degree of knowledge. Moreover, the rank could be reversed. Besides, if lecturers were perfect experts, then it would be possible to assume that a mark in an evaluation scale with regard to certain criterion has the same meaning for all lecturers when evaluating all the students, although not for all lecturers [5]. Hence, this means that there are different scales with different levels. But, in any case a perfect expert would be coherent and would have great ability to discriminate, even the degree of the uniformity of the teaching team. Unfortunately, lecturers are not perfect experts. A common effect is known as the batch effect, which often modifies the marks so that a student obtains a higher/lower mark when he/she is assessed together with other students who are clearly worse/better.

Despite those discrepancies, the hypothesis that lecturers are able to decide who is better from two students in the execution of an activity can be regarded as reliable. Taking into account this principle, the goal is to obtain a global ranking of the resolutions of an activity from the partial ranking between members of a pair. If this is possible, and in fact it is, then it is not necessary for lecturers to agree on the marks, just to order them in pairs.

The students' marks in each criterion according to the weights previously defined by lecturers are available and that information will be expressed by judgement preferences. Fig. 1 shows a diagram of the whole process.

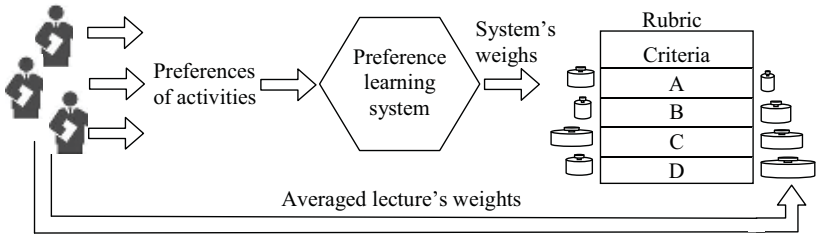


Fig. 1 Process of adjusting weights of the criteria of a rubric from preferences

A preference judgment [4] is an ordered pair whose first element has been preferred to the second. Then, a set of positive preferences (the first element is preferred to the second) and a set of negative ones (the second element is preferred to the first) are built. The challenge is to obtain a ranking as coherent as possible to the preferences expressed by the lecturers about the students. This ranking must be able to establish a correct order

to other unknown items which could be introduced to the system in the future. Then, the target is to obtain a function from the set of solutions of an activity proposed by the students so that it satisfies the following rule for the majority of unknown students.

$$u \text{ is preferred to } v \leftrightarrow f(u) > f(v) \tag{1}$$

For this purpose, one could think of applying a traditional regression method. However, these methods aim to find a function that minimizes the loss between the real and forecasted values, whereas in the preference learning model the target is to minimize the times the value that defines the ranking of a student is lower than the second one in each preference, taking into account that such student has been preferred to the other. Then, this guarantees that the marks are coherent with the preferences rather than produces an exact mark.

The problem could also be reduced to find just a linear function, since the weights of the criteria are the same, regardless of the marks the students obtain in such criteria. For instance, if a student has the mark 7 in a criterion, the weight of this criterion is the same if such student had had the mark 9. An advantage of the linear case is the intuitive interpretation of the function, since each criterion will be weighted using its corresponding value. For instance, a null value means that this criterion does not have any effect over the final mark. Hence, the rule shown in (1) is expressed as the rules shown in (2) for positive and negative preferences respectively:

$$\begin{aligned} &u \text{ is preferred to } v \leftrightarrow f(u) > f(v) \leftrightarrow f(u - v) > 0 \\ &v \text{ is preferred to } u \leftrightarrow f(u) < f(v) \leftrightarrow f(u - v) < 0 \end{aligned} \tag{2}$$

Therefore, the function will be positive if the first item is preferred to the second one and negative otherwise. This allows defining a training example as *u*-*v* with class +1 and *u*-*v* with class -1.

Once the data set is defined, it is possible to learn a ranking function from the binary classification yield by a binary classifier that separate the classes depending on the sign returned. Among the different methods available to obtain such function, one might a priori think of using a machine learning system based on a decision tree. But even if the performance of that kind of models is high, it may not be as useful as expected, since such methods tend to use the minimum possible number of features. From the point of view of the weights of the criteria, this means that many criteria will have weights equal to zero. However, SVM maximize the margin between the model and the nearest examples, called support vectors and this makes them use all relevant features [8]. Imagine that most of the students obtain the same degree of fulfillment in two criteria. These criteria are redundant since it will be easy to obtain the degree of one of them in function of the other. In this case, a decision tree would choose one of the criteria since both have the same relevance, whereas SVM would give similar weights, which is the desirable situation in this case. Figure 2 compares a possible model obtained by the two approaches, where a positive example is labeled with class +1 and a negative one with class -.

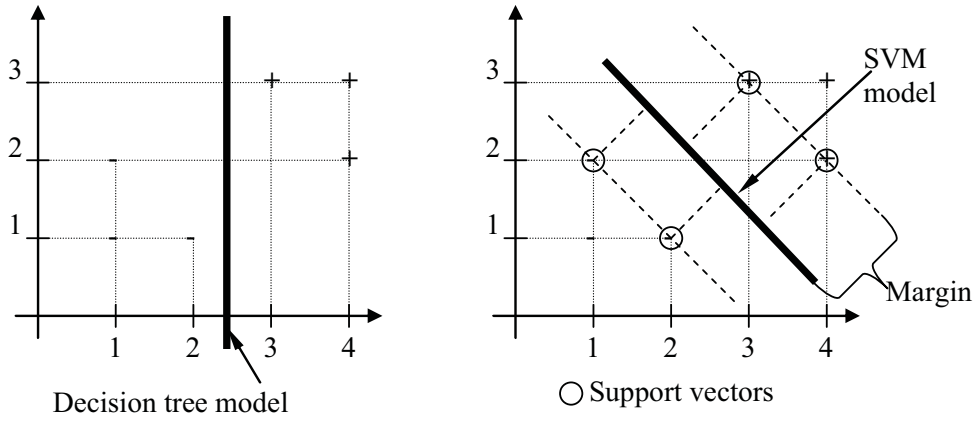


Figure 2. Representation of the model generated by two different binary separators. The decision tree model tries to use the minimum number of features, whereas SVM tries to generate a model that uses all relevant features.

Particularly, SVM [13], [14], [15] is a good choice for performing pairwise learning to aggregate the knowledge extracted from a set of different experts (lectures in this case) [2]. The linear learned function passes through the origin of coordinates and it is then expressed by

$$f(u) = \langle w, u \rangle = \sum_{j=1}^{p} w_j u_j \tag{3}$$

where w is the vector that will defined the separation hyperplane, u is the representation of a student that will be the vector of marks in the criteria, $\langle w, u \rangle$ denotes the scalar product of vectors w and z and p is the number of criteria. From a student u the function provides a value that quantifies the preference of such student against other students. The weights of the criteria are obtained from the vector that defines the separation hyperplane. In fact, it is only necessary to normalize and multiply it by the maximum mark a student could reach in the activity. Unlike SVM, a complex process would have been necessary to obtain the weights from the rules produced by decision tree method

4 Experiments

This section describes the data set and some experimental settings. It also includes a discussion of the results.

4.1 Data set and experimental settings

The data set comes from a core and compulsory course of second year of the Computer Science degree related to database design. Several activities were proposed during the year to perform continuous evaluation. Particularly, the lecturers of the subject agree in defining 10 criteria for Activity 1, 9 criteria for Activities 2 and 4 and 8 criteria for

Activity 9. The number of students for the experiments was 44 and the number of lecturers was 3.

Two kind of judgement preferences were taken into account. The first group includes those preferences where one of the students of the pair has better mark in all criteria. In this case, the value of the preference is obvious, that is, it will be positive if such student is the first student of the pair and negative otherwise. All the preferences satisfying such condition were included in the data set. The preferences of the second kind are those whose students have better marks in some criteria and worse in others. In this case, just a large enough random sample of pairs of students (60 preferences) satisfying such condition was considered. Then, the sample was equally split in as many subsets as lecturers of the subject. Each sample set was presented to a lecturer who has had a look at the resolutions of every pair and has expressed his/her opinion about who solved the activity better within each pair. Obviously, lecturers have not take into account any weight of the criteria; they just express their own general impression about which one is better. Notice that taking into account all possible pairs means to compare about $(n^2-n)/2$ where n is the number of students (not all of them since the preferences of the first kind were removed from this group), what would be unapproachable.

The experiments were performed using LibSVM [6] with default parameters together with the Spider Matlab toolbox [16]. The default parameters consist of choosing a linear kernel and of setting the trade-off between training error and margin to be 1.

4.2 Discussion of results

Five different experiments were compared for each activity. The first three consisted of checking in what extent the preferences of each of the three lecturers are coherent with the marks computed according to their own weights previously fixed. This is shown in the first three rows of Tables 2-5. The fourth experiment consisted of checking in what extent the preferences of all the lecturers together are coherent with the marks computed according to the weights obtained as the average of the weights of the three lecturers. This is shown in the fourth row of Tables 2-5. Finally, the fifth experiment consisted of checking in what extent the preferences of all the lecturers together are coherent with the marks computed according to the weights the learning process produces from the preference data. This is shown in the last row of Tables 2-5. Notice that the errors committed when the averages of the weights among the lecturers are considered are not necessary the averaged errors committed by each lecturer on their own. Besides, the number of preferences considered when the averages of the weights are computed and when the learning process is applied is the sum of the preferences of all the lecturers.

Table 2 shows that lecturers commit some errors when they express their global impression with regard to their own weights of the criteria in Activity 1. Particularly, they disagree between 5% and 15% of the preferences, whereas the system is able to accurately reproduce a summary of all them (0% of error). Notice that using the averaged weights does not lead to an improvement. It seems that this activity presents great difficulties when defining a set of weights, since the weights of the system in general are quite different from those previously defined by the lecturers.

Table 2. Weights of the lecturers, weights averaged and weights of the system for Activity 1

	Criteria weights										Pairwise	
Lectures	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Nº**	**Errors**
1	2.5	2.5	1	1	1	1	0.25	0.25	0.25	0.25	20	5%
2	2	2	1	1	1	1	1	0.5	0.25	0.25	20	15%
3	2	3	1	1	1	0.5	0.5	0.5	0.25	0.25	20	15%
Average	2.17	2.5	1	1	1	0.83	0.58	0.42	0.25	0.25	60	11.66%
System	1.28	1.90	1.28	0.48	0.19	0.95	0.48	0.96	1.52	0.95	60	0%

In case of Activity 2 presented in Table 3, lecturers seem to agree among them about the weights, but there are slightly high differences between the marks of these weights and their own preferences, since they commit between 20% and 35% of errors. In this case the proposed system produces 8.33% of error against 20% if the average of weights is used. The differences between the weights produced by the system and those of the lecturer are useful for the lecturers as a feedback to make them think about the relevance of the criteria.

Table 3. Weights of the lecturers, weights averaged and weights of the system for Activity 2

	Criteria weights									Pairwise	
Lectures	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Nº**	**Errors**
1	0.5	0.5	0.5	0.75	0.5	0.75	0.5	0.5	0.5	20	35%
2	0.5	1	0.5	0.5	0.5	0.75	0.5	0.25	0.5	20	20%
3	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	20	20%
Average	0.67	0.67	0.5	0.58	0.5	0.67	0.5	0.42	0.5	60	20%
System	0.85	0.85	0.8	0.45	0.23	0.28	1.1	0.23	0.23	60	8.33%

The results of Activity 3 shown in Table 4 are quite similar to those of Activity 2. Again the system is able to engage the information of the lecturer team to reduce the error.

Table 4. Weights of the lecturers, weights averaged and weights of the system for Activity 3

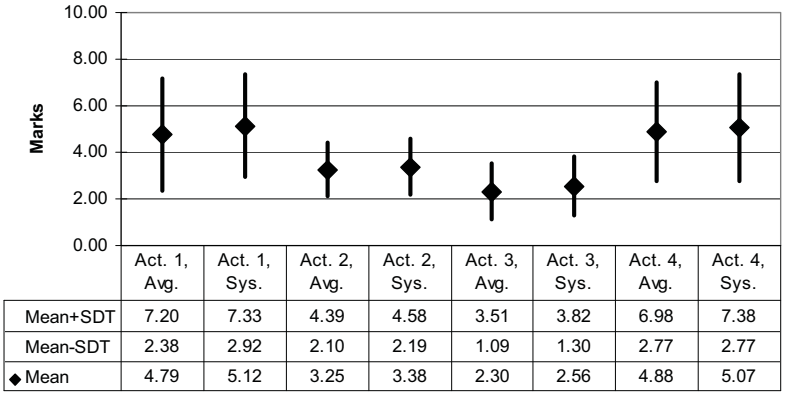
	Criteria weights								Pairwise	
Lectures	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Nº**	**Errors**
1	0.75	0.75	0.5	0.75	0.75	0.5	0.5	0.5	20	15%
2	0.25	0.5	0.5	0.75	0.5	0.75	1	0.75	20	30%
3	1	0.5	0.25	1	0.25	0.75	1	0.25	20	25%
Average	0.67	0.58	0.42	0.83	0.5	0.67	0.83	0.5	60	21.66%
System	1.1	1.2	0.44	0.44	0.37	0.73	0.22	0.48	60	5%

Looking at Table 5 for Activity 4, criteria 8 and 9 is quite interesting. In this case lecturer 1 does not take into account criterion 8 and lecturer 2 does not take into account criterion 9, but lecturer 3 grants equal weight to both criteria. This is a conflictive case and the system according to the preferences of the lecturers agrees with lecturer 1 about the criteria 8 and with lecturer 3 about criteria 9. This proves that the system try to sum up the preferences of all lecturers, although it produces a bit more error than lecturer 1.

Table 5. Weights of the lecturers, weight averaged and weights of the system for Activity 4

	Criteria weights									Pairwise	
Lectures	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Nº**	**Errors**
1	1.5	1	2	1	0.5	1	2	0	1	20	0%
2	0.5	2	2	1	0.5	1	2	1	0	20	25%
3	1	2	1.5	1	0.5	1	2	0.5	0.5	20	25%
Average	1	1.67	1.83	1	0.5	1	2	0.5	0.5	60	16,66%
System	1.94	1.40	1.71	1.29	0.89	0.63	1.65	0	0.49	60	5%

In general, the weights produced by the system differ from those granted by the lecturer before. Let us notice that the percentage of errors committed by the learning system are considerable lower that the rest ways of considering the weights. This means that this system is able to quite accurately reproduce the whole preferences of the lecturers. This also means that lecturers are not perfect experts because their own way of setting weights are not so coherent with their own preferences. Hence, the weights produced by the system make lecturers check their own incoherencies in order to change all or some weights which leads to establish a more accurate rubric. In fact, it helps to reach a consensus of the assessment process to encourage transparency and avoiding discriminatory treatment.



	Act. 1, Avg.	Act. 1, Sys.	Act. 2, Avg.	Act. 2, Sys.	Act. 3, Avg.	Act. 3, Sys.	Act. 4, Avg.	Act. 4, Sys.
Mean+SDT	7.20	7.33	4.39	4.58	3.51	3.82	6.98	7.38
Mean-SDT	2.38	2.92	2.10	2.19	1.09	1.30	2.77	2.77
◆ Mean	4.79	5.12	3.25	3.38	2.30	2.56	4.88	5.07

Figure 3. The averaged marks over the students when they are obtained from the weights averaged over the lecturers and from the weights the system grants

Applying the weighs the system produces would benefit some students and damages others. But, the question is that if there would be a global benefit or damage. Figure 3 shows the averaged marks of each activity together with the dispersion with regard to the use of averaged weights and to the use of the weights yield by the system.

At sight of Figure 3, one can observe that the mean and the deviation hardly vary between using average weights and the weights of the system. This allows concluding that the global benefit or damage will be the same. The advantage is that the marks of the students will be more accurately with regard to the global impression of the lecturer team. Notice that this process is internal among the lecturers and can be transparent for the students. Hence, it is not necessary to provide information to the student about the way of defining the weights.

5 Conclusions and Future Work

This work proposes a method based on preference learning to improve and adjust the weights granted to the criteria of an evaluation rubric according to the global impression of lecturers about pairs of activities solved by students when more than a lecturer is involved in the assessment process. The system proposed allows summing up the preferences of all the lecturers at the same time, and in fact, it reduces the errors between their own preferences and the original weights granted by every lecturer alone. Initially, lecturers give higher weight than the system yields from their preferences or vice versa. The tendency, unconsciously or not, of mixing criteria or taking into account other abilities such as transversal ones or those related to the attitude may be the cause of these disagreements. The results suggest lecturers must think about going more in depth into the design of the rubrics and about establishing more accurately the criteria and their relevance alone and together with their colleagues. Also, the weights the system grants benefit or damage the students the same with regard to consider the averages of the weights of all lecturers.

A proposal for future work is to find out if either grouping or breaking down the criteria makes lecturers improve the design of the rubrics.

Acknowledgements This research has been partially supported by the MICINN grants TIN2008-06247 and TIN2007-61273. The support of the University of Oviedo to the project entitled *La minería de datos como mecanismo de ayuda para la toma de decisiones en la actividad docente dentro del marco del Espacio Europeo de Educación Superior* is also gratefully acknowledged.

References

[1] Alan, J. Learning Outcomes in Higher Education. *Students in Higher Education,* 1996, 21(1), p. 93-108.

[2] Bahamonde, A., Bayón, G. F., Díez, J., Quevedo, J. R., Luaces, O., del Coz, J. J., Alonso, J., Goyache, F. Feature Subset Selection for Learning Preferences: A Case Study.

Proceedings of the 21st International Conference on Machine Learning, ICML 2004, Banff, Canada, 2004, p. 49-56.

[3] Bahamonde, A., Díez, J., Quevedo, J.R., Luaces, O., Del Coz, J. J. How To Learn Consumer Preferences From The Analysis Of Sensory Data By Means Of Support Vector Machines (SVM). Trends in Food Science & Technology, 18 (1), 2007, pp. 20-28.

[4] Branting, K., Broos, P. Automated Acquisition Of User Preferences. International Journal of Human-Computer Studies, 1997, p. 55–77.

[5] Cohen, W.W., Schapire, R.E., Singer, Y. Learning To Order Things. En Michael I. Jordan, Michael J. Kearns, y Sara A. Solla, editors, Advances in Neural Information Processing Systems, The MIT Press, 1998, 10.

[6] Fan R.E., Chen P.H., Lin. C.J. Working Set Selection Using Second Order Information For Training SVM. Journal of Machine Learning Research, 2005, 6, p. 1889-1918.

[7] Joachims, T. Optimizing Search Engines Using Clickthrough Data. In Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26, 2002.

[8] Joachims, T. Text Categorization With Support Vector Machines: Learning With Many Relevant Features Proceedings of ECML-98, 10th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE, 1998, 1398, p. 137-142.

[9] Moskal, B. M. Scoring Rubrics: What, When, And How? Practical Assessment, Research, and Evaluation, 2000, 7, 3.

[10] National Research Council. National Science Education Standards. Washington (DC): National Academy Press, 1996.

[11] Nitko, A.J. Educational Assessment Of Students. Upper Saddle River, NJ: Merrill, 2001.

[12] Realising the European Higher Education Area, Conference of European Ministers responsible for Higher Education, Berlín 2003. http://www.bologna-berlin2003.de/

[13] Schölkopf, B., Smola, A. J. Learning With Kernels. MIT Press, 2002.

[14] Shawe-Taylor, J. Cristianini, N. Kernel Methods For Pattern Analysis. Cambridge

University Press, 2004.

[15] Vapnik, V. Statistical Learning Theory. John Wiley, 1998.

[16] Weston J., Elisseeff A., BakIr G., Sinz F. Spider: Library Of Objects In Matlab. Software available at: http://www.kyb.tuebingen.mpg.de/bs/people/spider/