Bayesian classification with discrete templates - Implementation and Testing

prepared by: P. Tsalmantza, C. A. L. Bailer-Jones
reference: GAIA-C8-TN-MPIA-PAT-010-1
issue: 1
revision: 0
date: 2012-08-22
status: Issued

Abstract

This document describes the results of the first tests on the performance of the discrete template classifier described in CBJ-061 on real and toy data.

Contents

1 Introduction

In this technical report we describe the implementation and testing of the Discrete Template Classifier (DTC) that was presented in CBJ-061. The present implementation is performed in the R programming language. In all the examples that follow we first split our data into a set of labeled data (i.e. of known classes, templates) and a set of unlabeled data for which we use the DTC in order to assign class probabilities to each object. As described in CBJ-061, DTC is a semi-supervised method that uses both the labeled and unlabeled data in order to classify the latter set of data. To do so the algorithm first uses equation (1) in CBJ-061 in order to calculate the likelihood of each unlabeled data point for each template.

The resulting likelihood models are used through equation (3) in equation (7) in the same document, to define the function that needs to be maximized in order to estimate the optimal set of parameters α_k and $\beta_j^{(k)}$. The optimization was performed using the function optim() in R with the conjugate gradients method. The function was run for 15000 iterations. Even though in most of the examples that follow, the method had not formally converged after 15000 iterations, the values of α_k and $\beta_j^{(k)}$ are practically unchanged in the last iterations and therefore it should be enough for our purposes. In most of the tests presented in this report, as an initialization for the parameter values we selected a normalized flat prior (i.e. we set all values of $\beta_j^{(k)}$ and α_k equal to 1 and we normalized them as described in equation (8) in CBJ-061).

Once the values of α_k and $\beta_j^{(k)}$ have been estimated, the DTC algorithm uses equation (4) in CBJ-061 in order to calculate the class probability for each source.

In order to test this implementation of the DTC algorithm we have performed a number of tests using i) SDSS QSOs-stars photometric data, ii) simulated Gaia BP/RP spectra for stars, galaxies and QSOs and iii) 2-dimensional toy data. In the sections that follow we present the results of the DTC on all these datasets.

2 SDSS QSOs-stars classification

2.1 Performance of DTC - Comparison with SVM

For the first tests on the implementation of the algorithm we have used a small photometric dataset of QSOs and point sources from SDSS DR6. The method was applied to the unlabeled data of 463 stars and 537 QSOs while the labeled data of 230 stars and 270 QSOs were used as our templates (figure 1). Initially the SDSS color u-g and the magnitude i were used for the classification.

Following the steps of the algorithm described in section 1 and using the normalized flat prior

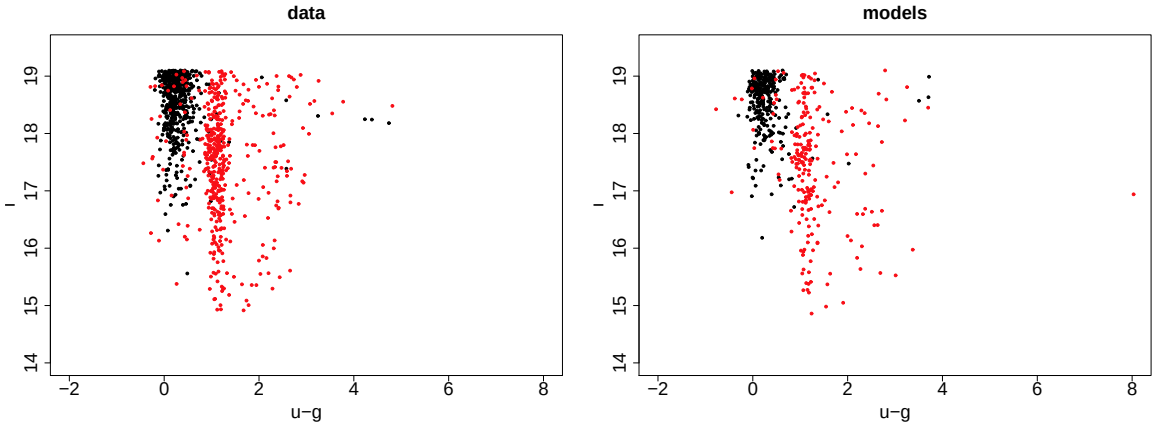

Figure 1: Color-magnitude diagram (CMD) for the unlabeled (left) and labeled (right) data sets used for the application of the method. The red and the black points represent the stellar and QSO sources respectively.

as the initial values of the parameters α_k and $\beta_j^{(k)}$, we calculated the class probability for each source in the unlabeled sample for both classes. The results for the present sample are presented in figure 2 and in table 1. The estimated values of α_k are equal to 0.55 and 0.45 for QSOs and stars respectively which corresponds approximately to the ratio of these classes in the labeled sample (0.54 QSOs and 0.46 stars).

In order to check if usually the method leads to values of α_k that correspond to the class ratios in the labeled sample we performed a small test. In this test we used in the labeled sample only 5 QSOs and 5 stars and we set the initial values of α_k equal to 1 and for the initial $\beta_j^{(k)}$ 2 were set to zero and 3 were set equal to 1 for each class. These values were once again normalized based on equation (8) in CBJ-061. The same set of templates were also used as the unlabeled sample in our test. The resulting values of α_k were approximately 0.5 (0.503 and 0.497) and the resulting $\beta_j^{(k)}$ were all approximately equal to 0.2 as expected.

Table 1: Summary of the performance of the classification models. The rows and columns correspond to the true and predicted classes of the objects respectively.

	DTC		not optimized DTC		SVM	
	Q	S	Q	S	Q	S
Q	484	53	488	49	514	23
S	53	410	60	403	44	419

From table 1 we see that the method has a 89.4% success rate in separating the two classes of objects. In order to see if this performance is good enough we classify the same sample of unlabeled data using i) the same method but without optimizing for the values of α_k and $\beta_j^{(k)}$ (i.e. using the initial normalized flat prior for their values) and ii) the Support Vector Machine

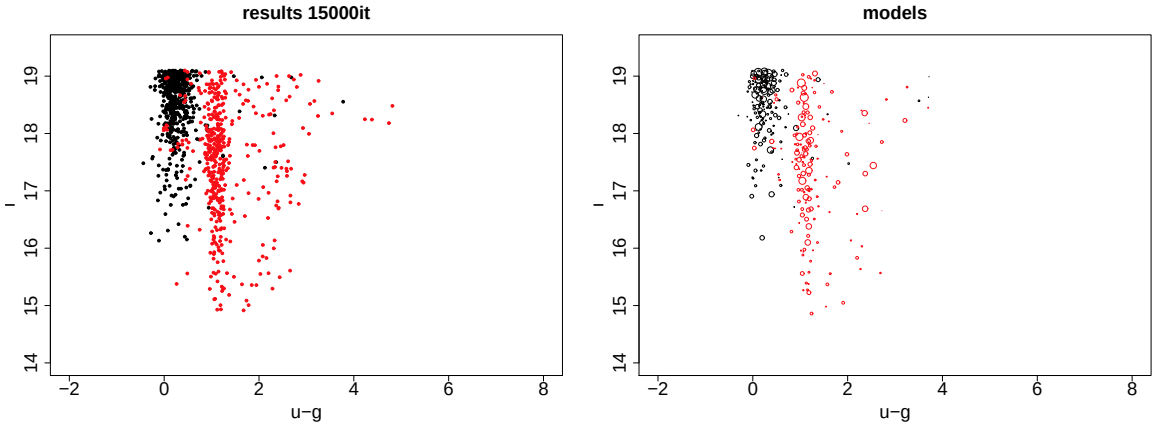

Figure 2: CMD of the results (left) and the labeled data (right) after the application of the method. The red and the black points represent the stellar and QSO sources respectively. In the right plot the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template).

(SVM) algorithm. The results of these two methods are presented in figure 3 and table 1.

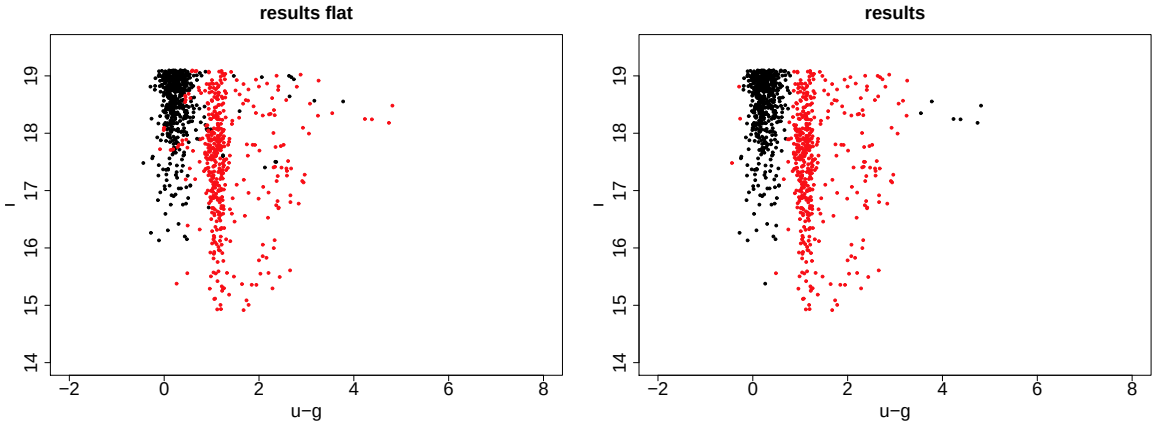

Figure 3: CMD for the results produced by (left) the DTC with the normalized flat prior and no optimization and (right) SVM. The red and the black points represent the stellar and QSO sources.

From table 1 we see that the performance of the optimized DTC method is marginally better than the one without the optimization (89.1% successful) but worse than the one of the SVM (93.3%). From figure 3 we see that the two classes have very hard boundaries in the case of the SVM classification. Therefore, SVM perform better in this example that the two classes are quite well separated.

2.2 Introducing duplicated information in the templates

In order to further test the behavior of the classifier we have performed some additional tests with the same datasets. In one of these tests we have artificially increased the number of stellar templates to see how this will affect the values of the α_k and $\beta_j^{(k)}$ parameters. More specifically, in our new template set every star is included 3 times. Therefore the stellar sample now consists of 690 instead of 230 stars while the number of QSOs remains at 270.

After applying the method to this dataset we see that the values of the α_k parameters are 0.55 and 0.45 for QSOs and stars respectively, values which are very close to the original class ratio (0.54 and 0.46). This indicates that the values of α_k that the method provides correspond to the "real" class ratios and they do not take into account redundant information. The results of the classification for this test are presented in figure 4 and table 2.

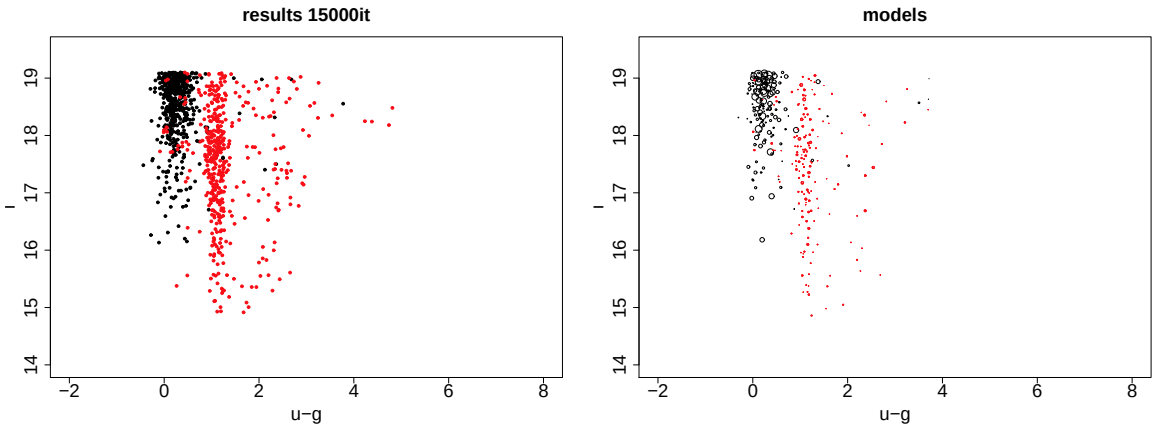

Figure 4: CMD of the results (left) and the labeled data (right) after the application of the method. The red and the black points represent the stellar and QSO sources. In the right plot the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template).

Table 2: Summary of the performance of the classification models. The rows and columns correspond to the true and predicted classes of the objects respectively.

	DTC	
	Q	S
Q	485	52
S	53	410

From the right plot of figure 4 we can see that the weight assigned by the optimization to each QSO template remains the same as in the case of figure 2, while for the stellar templates all the weights have become smaller but the relative contribution of each template in this class is kept

constant. This is because each weight has been divided by a factor of 3 in order to describe the three identical templates of each source that are now present in the labeled sample.

Finally, by comparing the results of the confusion matrices in tables 1 and 2 we see that the classification results were not affected by the increase of the training sample with duplicated information.

2.3 Changes in the class ratios

In section 2.2 we showed that the values of α_k that result from the DTC algorithm remained the same after changing the class ratio in the template sample using duplicated sources. In order to check that this is not the case when the class ratio really changed we performed another test with the same SDSS dataset used so far, but this time we artificially changed the class ratio in the two groups by changing the label in a subsample of sources. To do so we assigned the class of stars to every source with the magnitude i brighter than 18 mag. This lead to a sample of 268 stars and 232 QSOs. The results of the DTC show that this difference in the class ratio is also propagated in the α_k values after the optimization as it was expected (0.46 and 0.54 for QSOs and stars respectively). The results of this test are presented in figure 5.

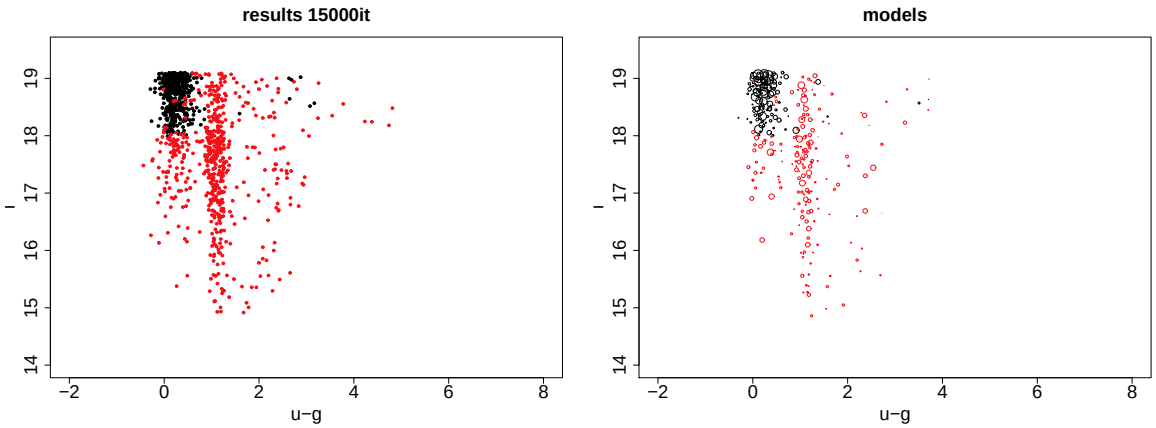

Figure 5: CMD of the results (left) and the labeled data in which we have assigned the class stars to all the sources with i<18 mag and (right) the models after the application of the method. The red and the black points represent the stellar and QSO sources. In the right plot the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template).

2.4 Classification using SDSS colors

Finally, in order to see if the method performs better when using more information, we have repeated the classification of QSOs and stellar sources but this time instead of using 1 color and 1 magnitude, we use the data of 2 colors (g-r vs u-g) for the classification (figure 6). The results

can be seen in figure 6 and table 3 and as it is obvious they are better than before (94% correct classification results).

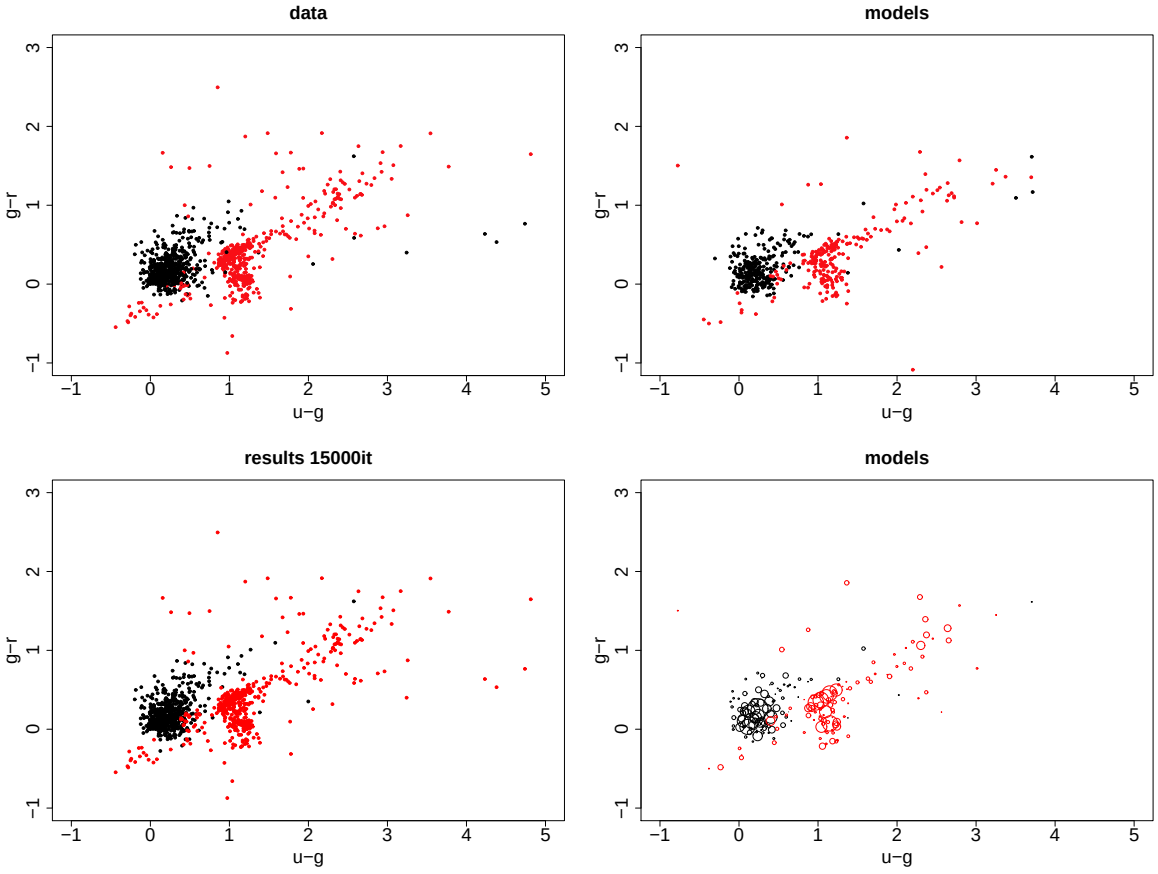

Figure 6: Color-color diagram of the unlabeled (top left) and labeled (top right) data sets used for the application of the method, as well as for its results (bottom left) and (right) for the models after the application of the method. The red and the black points represent the stellar and QSO sources. In the bottom right plot the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template).

3 Tests on Gaia simulated spectra

3.1 Classification using BP/RP spectra - comparison with SVM

Once the first DTC tests on 2D data were performed, we decided to use the method for Gaia classification purposes. In order to do so we have selected a random sample of simulated Gaia spectra of stars, galaxies and QSOs. For the classification we have used the 60 BP and the 60 RP pixels of each spectrum. The spectra are derived from the semi-empirical libraries of stars,

Table 3: Summary of the performance of the classification models. The rows and columns correspond to the true and predicted classes of the objects respectively.

	DTC	
	Q	S
Q	498	39
S	21	442

galaxies and QSOs. These spectra were simulated during cycle 6 and correspond to G magnitude equal to 18 mag. For the labeled sample we have selected 200 spectra from each class (in total 600 spectra) and for the unlabeled one 800 objects from each class (in total 2400 spectra). Following once again the procedure described in section 1, we first estimated the likelihoods and then we defined the function to be optimized. During this process we found that 163 spectra from our unlabeled sample were so far from any template that was used that led practically to minus infinite values of the sum used in equation (7) in CBJ-061. For this reason we have excluded these 163 sources and have performed the classification for the remaining 2237 spectra. The final data selected for the two samples are presented in figure 7. For visualization purposes the data are plotted in the two dimensional space of the 2 first Principal Components (PCs), even though the method was applied using the whole Gaia BP/RP spectra and not just the 2 PCs.

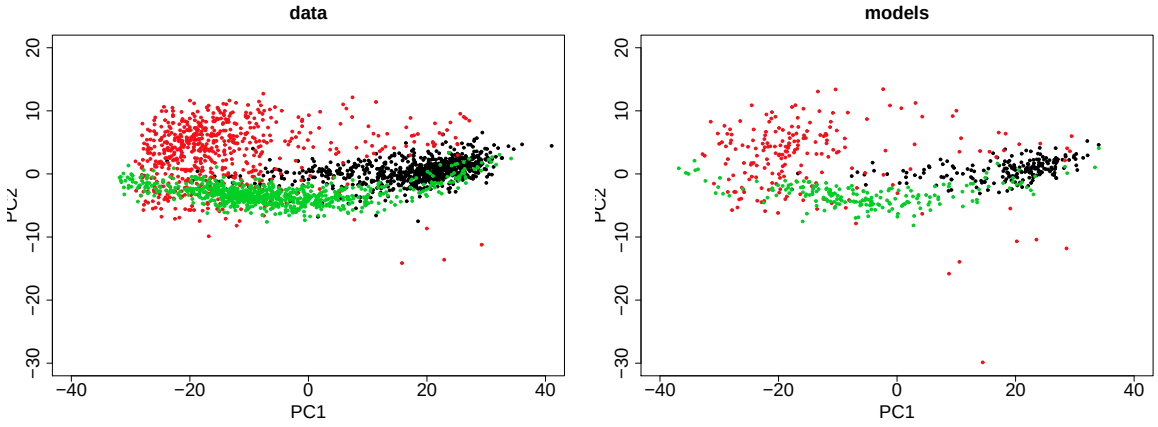

Figure 7: The first vs the second Principal Component for the unlabeled (left) and labeled (right) data sets used for the application of the method. The black, the red and the green points represent the galaxies, QSOs and stars in our sample.

The results of the DTC for the classification of this sample are presented in figure 8 and table 4. From this table we can see that DTC managed to classify 95.1% of the sources correctly.

In order to have a better understanding of the performance of the DTC on this data set, we have repeated the classification: i) without optimizing the α_k and β_j^k values, ii) using SVMs and

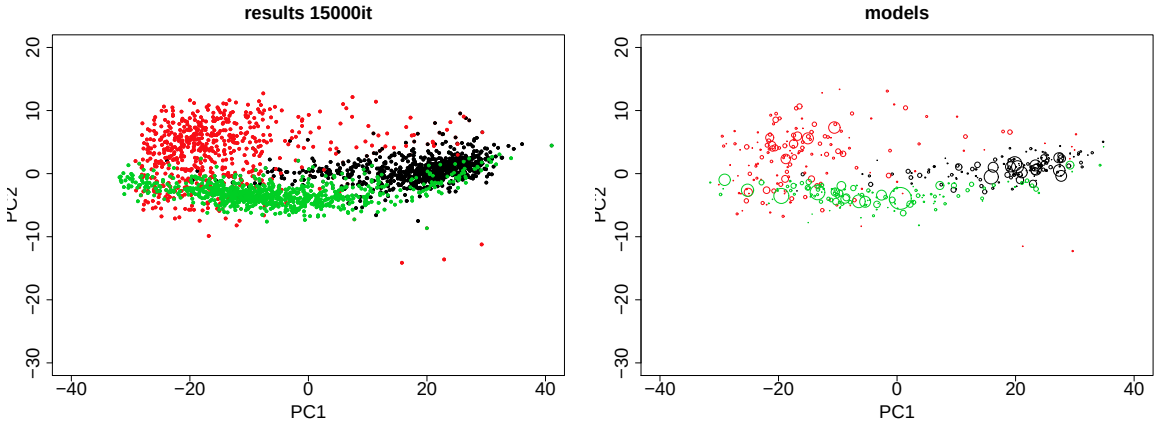

Figure 8: PC1 vs PC2 for the results (left) and the labeled data (right) after the application of the method. The black, the red and the green points represent the galaxies, QSOs and stars in our sample. In the right plot the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template).

Table 4: Summary of the performance of the classification models. The rows and columns correspond to the true and predicted classes of the objects. The results are based on 4 classifiers: The DTC with optimized α_k and $\beta_j^{(k)}$, the DTC without optimization, the SVM and the probabilities output of SVM.

	DTC			not optimized DTC			SVM			SVM prob		
	G	Q	S	G	Q	S	G	Q	S	G	Q	S
G	747	5	47	748	4	47	748	8	43	732	15	52
Q	25	601	27	25	601	27	10	628	15	3	634	16
S	6	0	779	4	0	781	7	8	770	5	12	768

obtaining their output and iii) using the probabilities that can be provided as an output of the SVM in R. In the third case the probability model for classification fits a logistic distribution using maximum likelihood to the decision values of all binary classifiers, and computes the a-posteriori class probabilities for the multi-class problem using quadratic optimization. The results for all these tests are presented in table 4, while the results of the standard SVM are visualized in the plane of the 2 first PCs in figure 9. By comparing the results for each classifier we see that once again the standard SVM method is giving the best performance (95.8% correct classifications) but only slightly better than the one obtained by the DTC. The results are slightly worse when the SVM output probabilities are used (95.4% correct classifications). What was a little suprising in this example is that the results of the DTC without optimization of the parameters are a little better than in the case where the coefficients have been optimized (95.2% correct classifications).

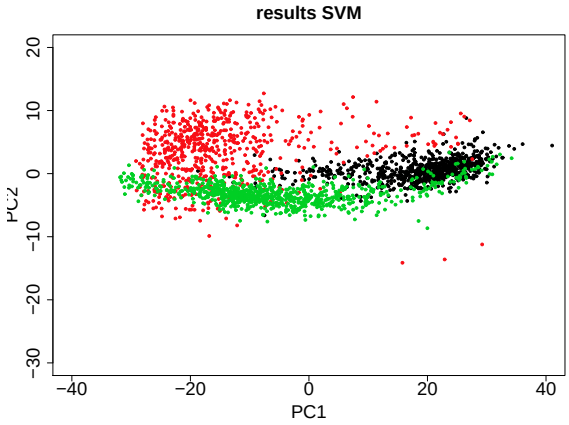

Figure 9: PC1 vs PC2 for the results of the SVM when applied to the unlabeled data. The black, the red and the green points represent the galaxies, QSOs and stars in our sample.

In general we should point out that the DTC method is very sensitive to the initialization used when optimizing the coefficients (i.e. the classification results are quite different for different initializations). Since the problem is not convex it is impossible to define the initialization that would lead to the global maximum of equation (7). The only way around this problem is to start with many different initializations and then select the one that leads to the larger maxima. The only problem with this solution is the very long time required in order to run the method. In table 5 that follows we present the dependence of the time required from the method to run on the amount of objects used for the labeled and the unlabeled dataset. The required time seems to be almost independent of the number of data points for each source which seems to be negligible at least when moving from 2 data points to 120 as in the examples described so far.

Table 5: Time required by TDC on 1 core on an Intel Xeon X5570 2.93GHz processor.

Number of objects in the training set	Number of objects in the testing set	Time for 3 iterations (sec)
500	100	1
500	500	5
500	1000	8
500	10000	84
100	500	1
1000	500	18

In an attempt to compare better the SVM and the DTC, we have compared the estimated probabilities for each class by both methods (figure 10). The comparison is done when the whole spectrum is used for the classification and not just the 2 PCs. We see that, even though the estimated probabilities are generally equal to 1, SVM seems to have a more continuous distribution

from 0 to 1 than DTC. The latter seems to classify everything with a probability either very close to 0 or to 1.

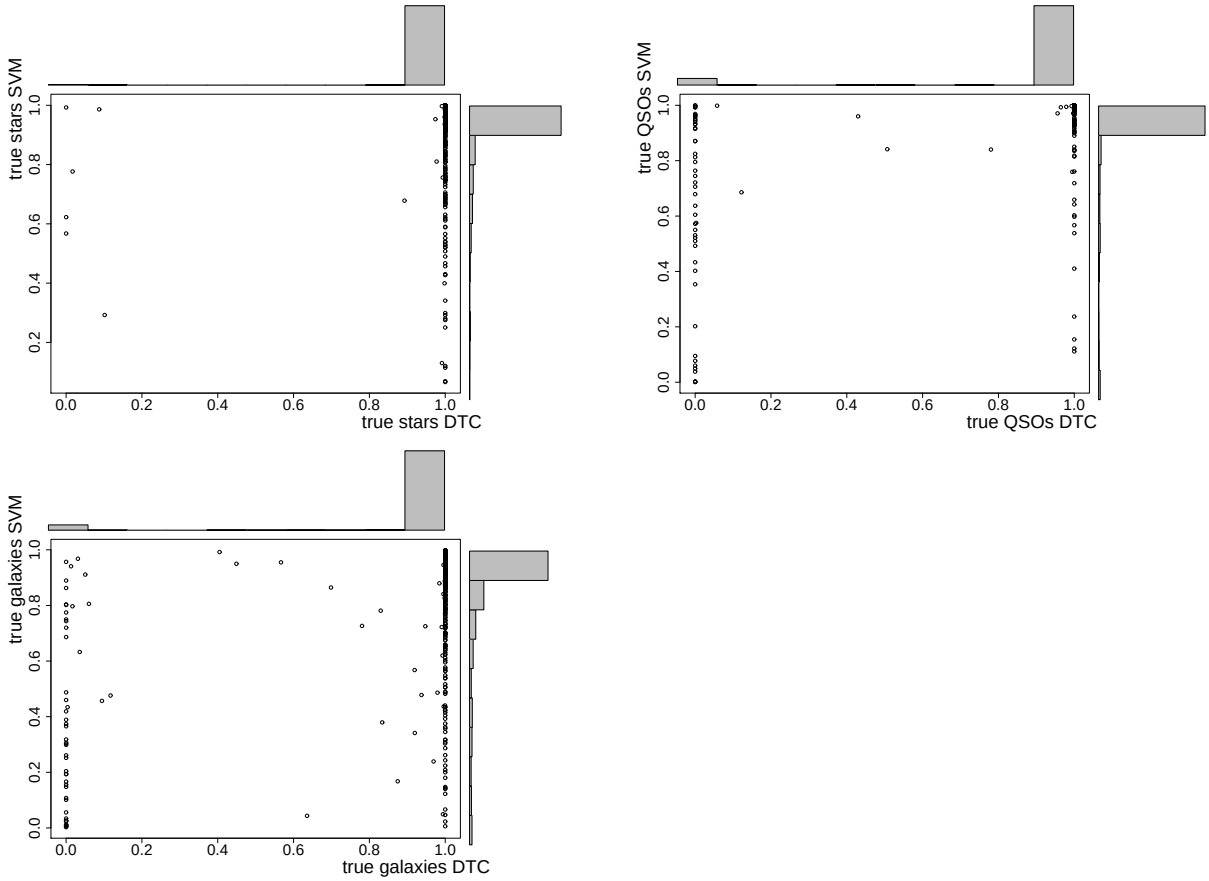

Figure 10: The probability distribution for the true stars (top left), QSOs (top right) and galaxies (bottom) as they were estimated by SVM (y-axis) and DTC (x-axis) to be of their true class.

As a final step in this analysis, we have compared the completeness and contamination for different probability thresholds for all the classes of the two different classifiers. The completeness is defined as the ratio between the true positives divided by the total number of true sources for each class while the contamination as the ratio between the false positives divided by the total number of sources that are classified into this class. Once again the data of the whole spectrum is used for the classification. The results are presented in figure 11.

3.2 Classification using Principal Components

Since the visualization of the results in this example is not straight forward related to the data, we have also applied the method for the same data but this time instead of using the whole BP/RP spectra we use only the 2 first PCs for each source. The results of this test are presented in figure 12 and table 6.

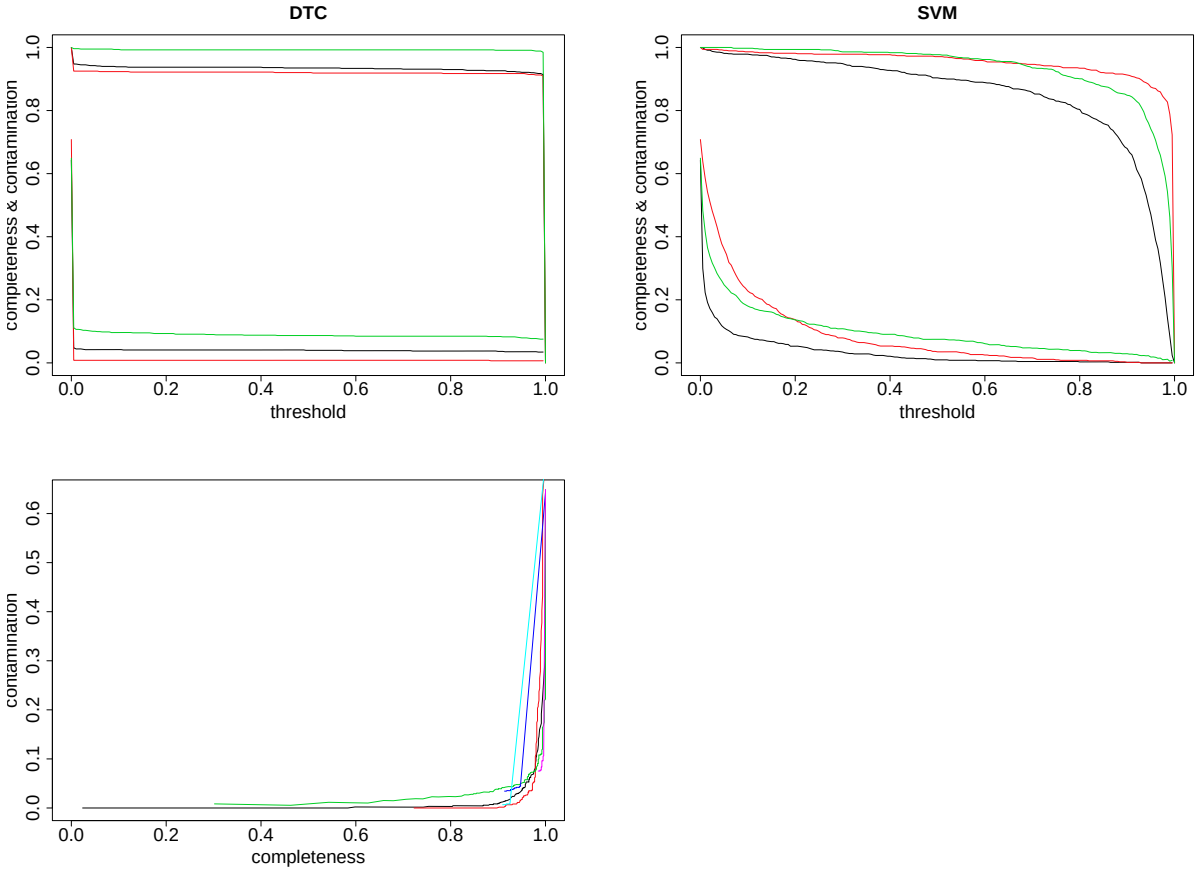

Figure 11: The completeness (upper lines) and contamination (lower lines) vs the probability threshold (Top panel), for the results of the (left) DTC and the (right) SVM. The results for galaxies, QSOs and stars are presented with black, red and green respectively. In the bottom panel plot the contamination vs the completeness for all the classes and both classifiers is presented. The color code is the same for the case of SVM as in the top panel plots. The blue, light blue and magenta lines correspond to galaxies, QSOs and stars for the case of the DTC results.

From these figures and table we can see that the performance was degraded a lot (74.8% correct classifications) compared to when using the whole spectrum as expected (section 3.1). In addition, we see that the weights assigned to each template are now very different from before.

4 Tests on toy data

4.1 The initialization problem

In these final tests we attempt to check the performance of the discrete template classifier on a control dataset. For this purpose we initially generated two different two dimensional classes

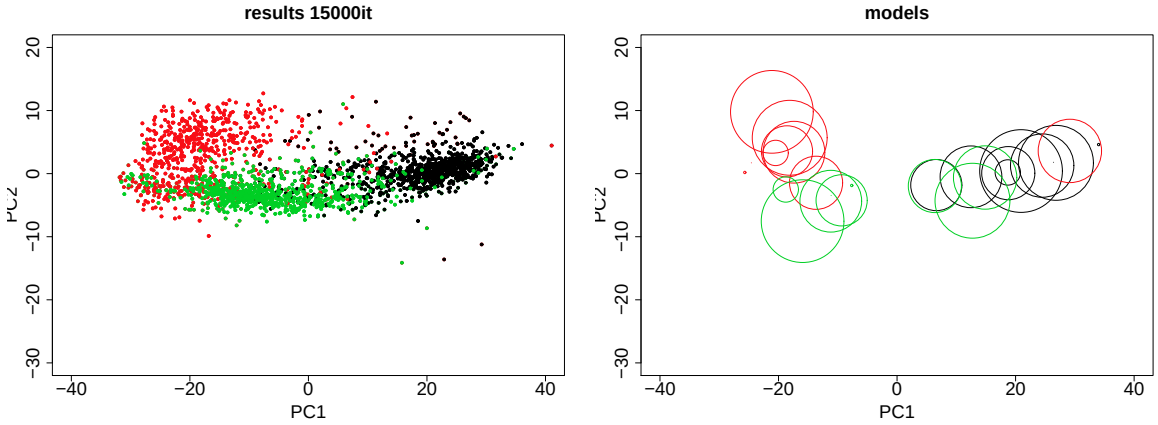

Figure 12: CMD of the results (left) and the labeled data (right) after the application of the method. The black, the red and the green points represent the galaxies, QSOs and stars in our sample. In the right plot the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template).

Table 6: Summary of the performance of the classification models. The rows and columns correspond to the true and predicted classes of the objects respectively.

	DTC		
	G	Q	S
G	680	24	95
Q	58	478	117
Q	140	129	516

that were randomly drawn from two gaussians with the same dispersion and whose centers were separated by a distance proportional to the sigma of the gaussians. This test was performed three times with distances between the two centers equal to 1, 3 and 15 sigmas. During this tests 500 sources were used as labeled and another 500 as unlabeled data. The data for the unlabeled sample is presented in figure 13 while the results of the classification produced by the DTC in table 7.

From table 7 we see that except in the case that the 2 gaussians were separated by 15 sigma distance, where DTC produces no misclasifications as expected, in the other two cases the performance degrades to 81.0% and 55.8% success rate for 3 and 1 sigma respectively. In order to check how much these results are sensitive to the initialization of the α_k and β_j^k parameters we reapplied the method but this time without setting equal initialization values for all the parameters that correspond to the two classes. The results of the classification for these two initializations for the case of the 3σ separation between the two classes are given in table 8. We see that the results of the classifier are very sensitive to the initialization. This problem could be

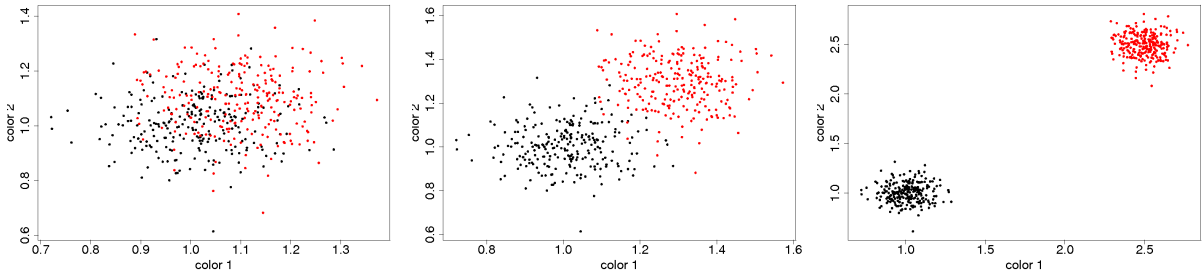

Figure 13: Color-color diagram of the 3 different realizations of the 2 toy data classes (black and red points) for the unlabeled sample. In each case the data was drawn from two 2D gaussians with the same variance and distance between the two centers equal to 1, 3 and 15 sigmas.

Table 7: Summary of the performance of the classification models with the DTC. The rows and columns correspond to the true and predicted classes of the objects respectively. The results are for 3 different realizations of the 2 toy data classes. In each case the data was drawn from two 2D gaussians with the same variance and distance between the two centers equal to 1, 3 and 15 sigmas.

	1σ		3σ		15σ	
	class1	class2	class1	class2	class1	class2
class1	101	149	186	64	250	0
class2	72	178	31	219	0	250

overcome by optimizing the α_k and $\beta_j^{(k)}$ parameters starting with many random initializations and selecting the one that resulted to the highest performance. Unfortunately this is not easily done for large datasets since the time needed for each run is very large.

4.2 Outlier detection

As a final test we check the ability of the classifier to identify outliers, i.e. objects that do not belong to the one of the labeled classes. To do so we use again the same toy data set as before with a separation of 3σ but this time we generate one additional class with the same separation from both the other two classes. In the tests that follow we use the two classes as labeled data and one time we include the third class in the unlabeled dataset and the other time we use only the two classes to produce the final model and apply it to the third class. The classification results of the two models for the third class are presented in figure 14.

From figure 14 we see that the weights $\beta_j^{(k)}$ assigned to each labeled source are different in the cases that 2 or 3 classes are used in the unlabeled sample. In the case that 3 classes were used, objects closer to the area occupied by the third class are assigned larger weights than in the case where only 2 classes were used. The classification results show that in the case that 3 unlabeled

Table 8: Summary of the performance of the classification models with the DTC for the case of the 3 sigma separation between the two classes. The rows and columns correspond to the true and predicted classes of the objects. The results are for 3 different initializations of the α_k and $\beta_j^{(k)}$ for each class. These initializations are equal to 0.5 and 0.5, 0.018 and 0.982 and 0.12 and 0.88 for all the α_k and 0.004 and 0.004, 0.00014 and 0.00786, 0.00095 and 0.00705 for all the $\beta_j^{(k)}$ of class1 and class2 respectively.

	0.5 - 0.5		0.018 - 0.982		0.12 - 0.88	
	class1	class2	class1	class2	class1	class2
class1	186	64	204	46	175	75
class2	31	219	25	225	23	227

classes were used in order to define the model 89 objects from the third class were found in class 1 while 161 in class 2. When only two classes were used in this process 98 were found in class 1 and 152 in class 2. In more detail the values of the $\beta_j^{(k)}$ coefficients for the case that two and three classes were used are presented in figure 15. In the same figure we have also plotted the classification probability of each source to belong to class 1 for both cases.

From figure 15 we see that no obvious outlier candidates can be detected based on the DTC probabilities for the objects in the third class.

5 Conclusions

In this technical note we have presented the implementation and the performance of the DTC in R. The code has been extensively tested with photometric data from SDSS, simulated Gaia spectra and toy data and compared with other classification methods such as SVM. The tests presented above led to a set of interesting conclusions about the method.

The performance of the DTC was proven to be very good for almost all the tests presented here. This shows that semi-supervised methods can be used quite successfully for classification problems.

The results of the optimization of the α_k and $\beta_j^{(k)}$ parameters show in practice that the former correspond to the ”true” class ratio in the templates while the later represent the importance of each template in the classification of the unlabeled sample, as was theoretically expected.

The comparison of the DTC results with the ones obtained with SVM in all the tests performed here showed that SVM always performs better but not by a lot. Another drawback of the DTC compared to the SVM is the time required for the code to run. Since the DTC model is extracted based on both the labeled and unlabeled data it is more time consuming than the one of

the SVM that use only the labeled dataset. On the other hand the tests presented here showed that the performance of the DTC when no optimization of the parameters is applied (in which case the method is very fast) is almost as good as the one obtained when the α_k and $\beta_j^{(k)}$ are optimized. Finally, the optimization of the DTC is not convex which makes it highly dependent on the initialization which strongly effects the results. On the other hand SVM are also sensitive to the selection of the parameters Cost and gamma that they are using. However, in the case of SVM our experience with mainly simulated Gaia spectra has shown that the results are not so sensitive on the values of these parameters as in the case of the DTC algorithm. In addition, the comparison of the resulting probabilities produced by SVM and DTC show that in our test examples the SVM probabilities have a more continuous distribution than the ones derived by the DTC which tend to be very close to 0 or 1 values. This may be a result of the distance from the continuous boundaries used by SVM and from the discrete set of templates used by DTC to define the output class probabilities. Finally, the comparison of the completeness and contamination between the results of the two algorithms showed that even though they present differences they are not so important. A more extensive comparison between SVM and a method very similar to the DTC when applied to the problem of star-galaxy classification is performed in Fadely et al. 2012.

As a last test we used toy data in order to check the ability of the DTC to detect outliers which seemed not to be an easy task based on the output class probabilities.

References

Bailer-Jones C.A.L & Hogg D. W. 2011, GAIA-C8-TN-MPIA-CBJ-061
Fadely R., Hogg D. W., William B. 2012, arXiv1206.4306

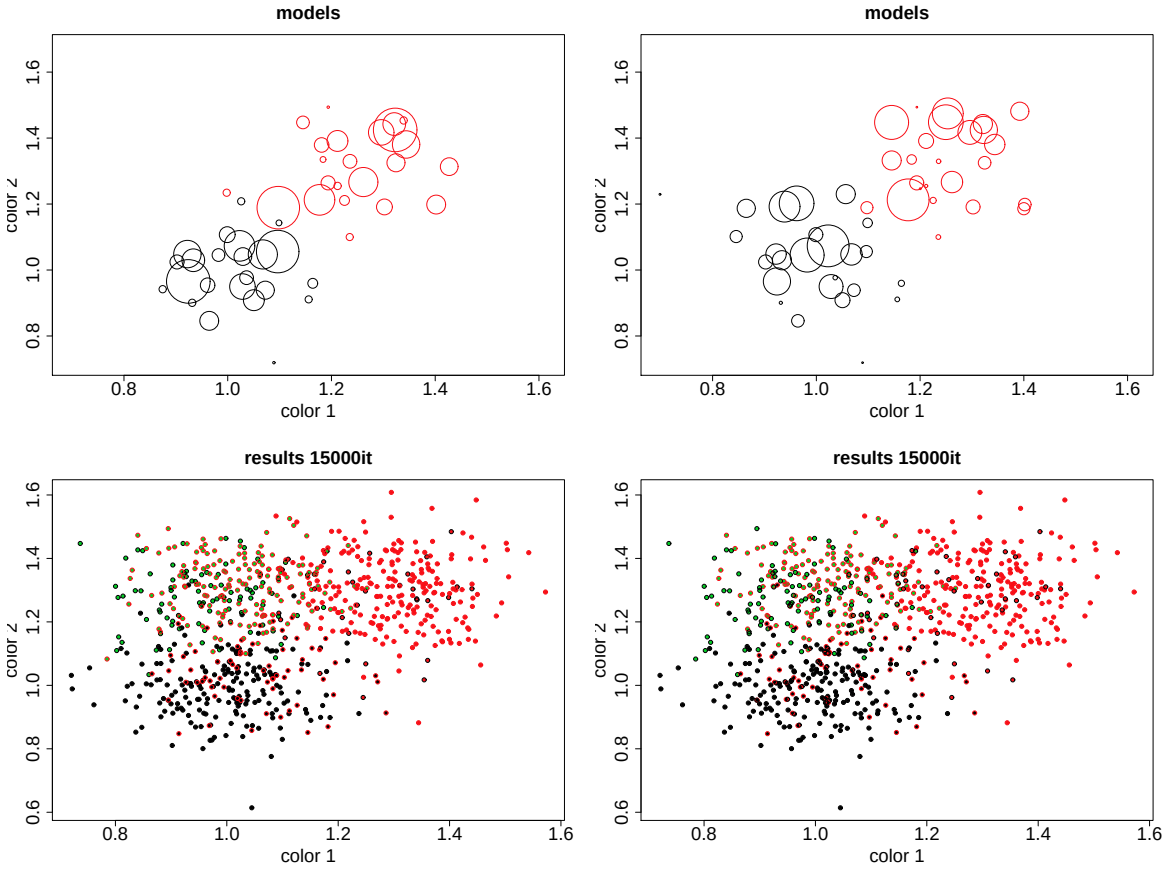

Figure 14: The results (left) for the 2D labeled data (top panel) and for the unlabeled data (bottom panel) after the application of the method. The left and the right column plots correspond to the cases where the third unlabeled class was excluded and included in the definition of the model respectively. The black, the red and the green points represent classes 1, 2 and 3. In the top panel plots the diameter of the circles represents the weight of each template based on our method (the diameter is proportional to the value $\beta_j^{(k)}$ that was estimated for that template). In the bottom panels the inner points correspond to the true class of each source, while the outer circles indicate the result of the classification.

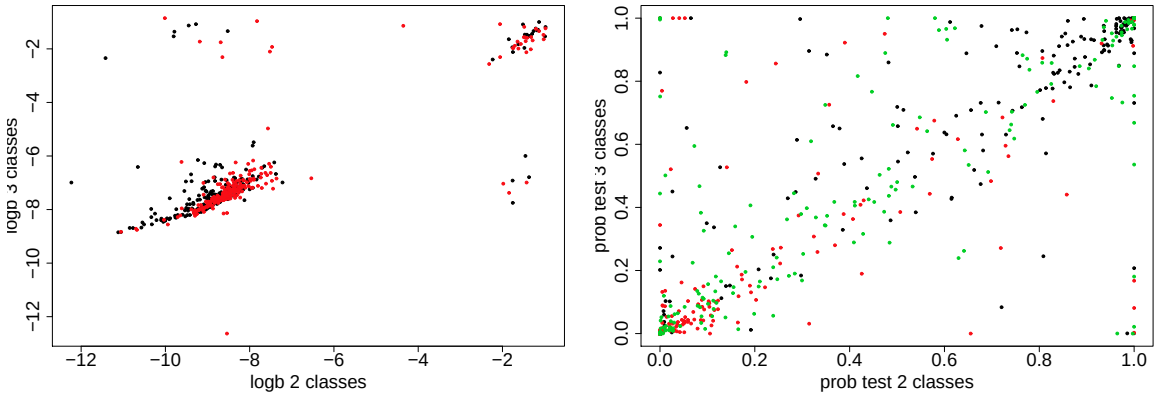

Figure 15: Left: The log of the $\beta_j(k)$ values that were produced by the optimization for the case when 3 vs. 2 unlabeled classes were used. Right: The probability for each source to belong to class 1 when 3 vs. 2 unlabeled classes were used. The black, the red and the green points represent classes 1, 2 and 3.