MergedFile

Unknown Author

May 4, 2014

Part I

by James Bladen and Steven Chang

Part II

PreProcessing

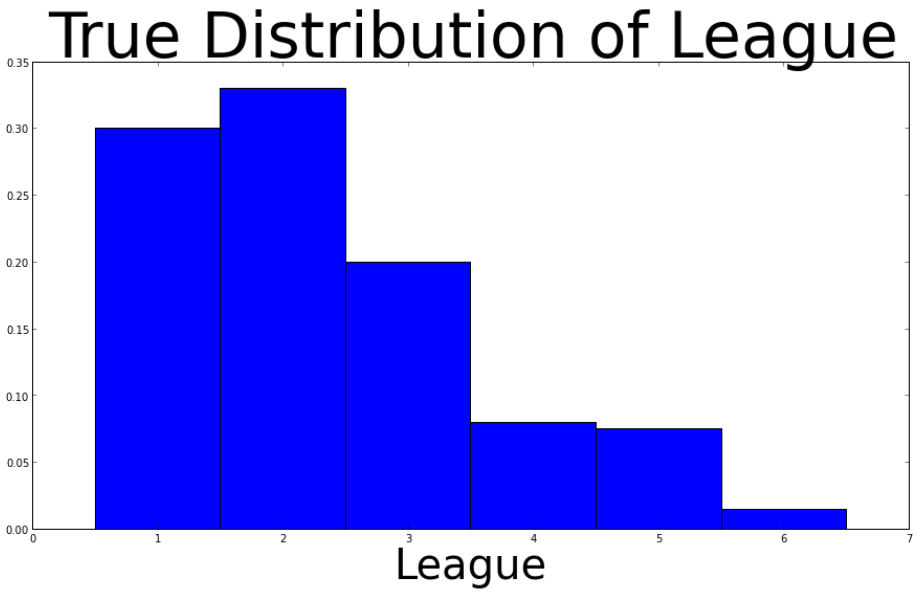


Fig. 1 Note that the true distribution of players is skewed to the right.
This means that there should be more lower-level players.
This data was collected from sc2ranks.com.

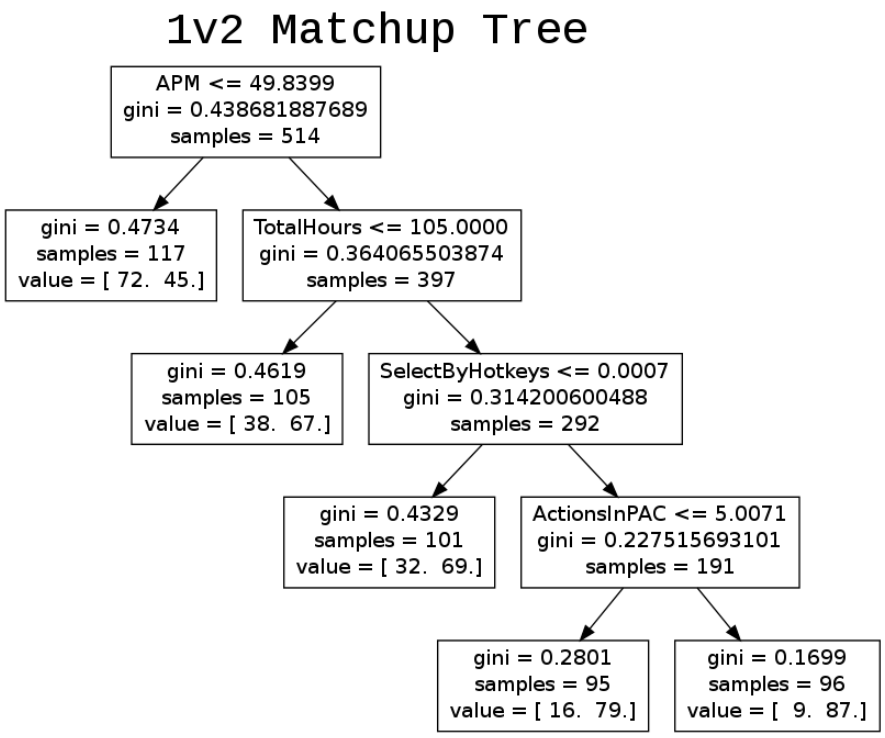
Fig. 2 Notice that we have the opposite skew for the data that we actually collected. This means that our data will be slightly biased towards better players, thus making our predictions slightly skewed towards larger values.

Out [33]:

Part III

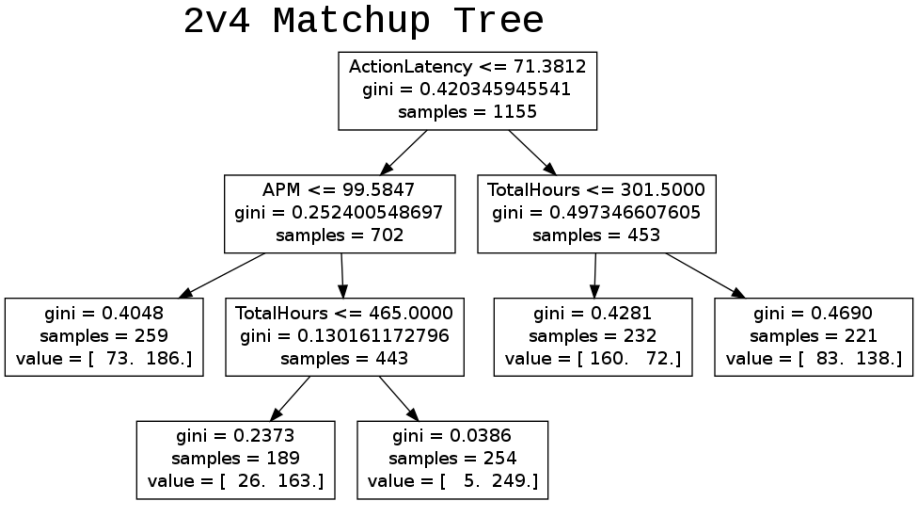
Trees

Out [47]:



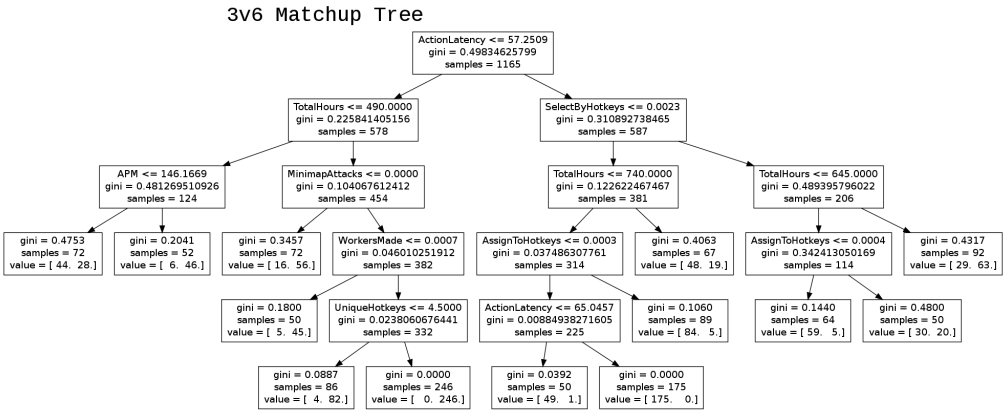
This tree features a matchup of the two lowest leagues. the first variable the tree splits on.

Out [48]:



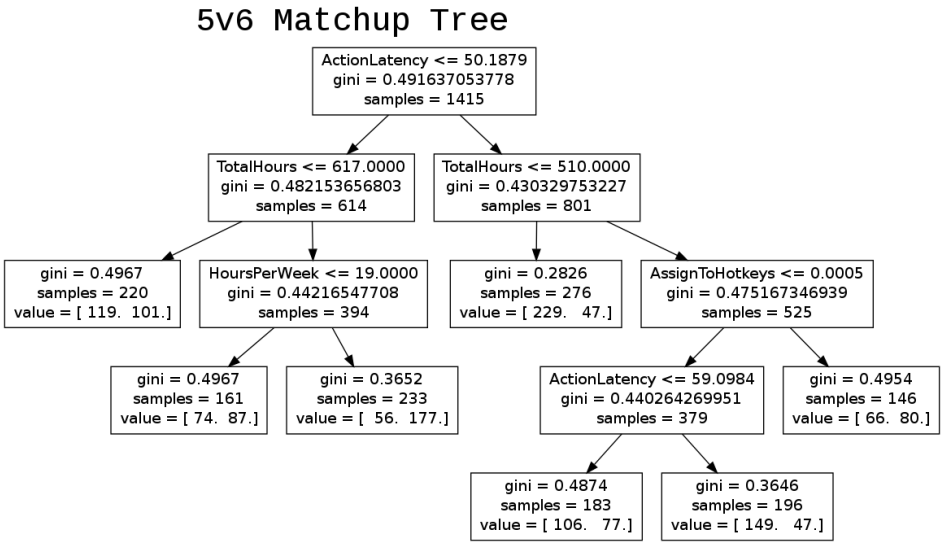
This tree features a matchup of a low and middle league. We see that APM is still present, but ActionLatency has become the first variable the tree splits on.

Out [49]:



This true features a matchup of a middle and high league. Notice how it is more complicated than the previous two trees. New variables involve the use of hotkeys.

Out [63]:



This tree features a matchup of two high leagues. It is not as complicated as the previous tree. ActionLatency is still the first variable the tree splits on.

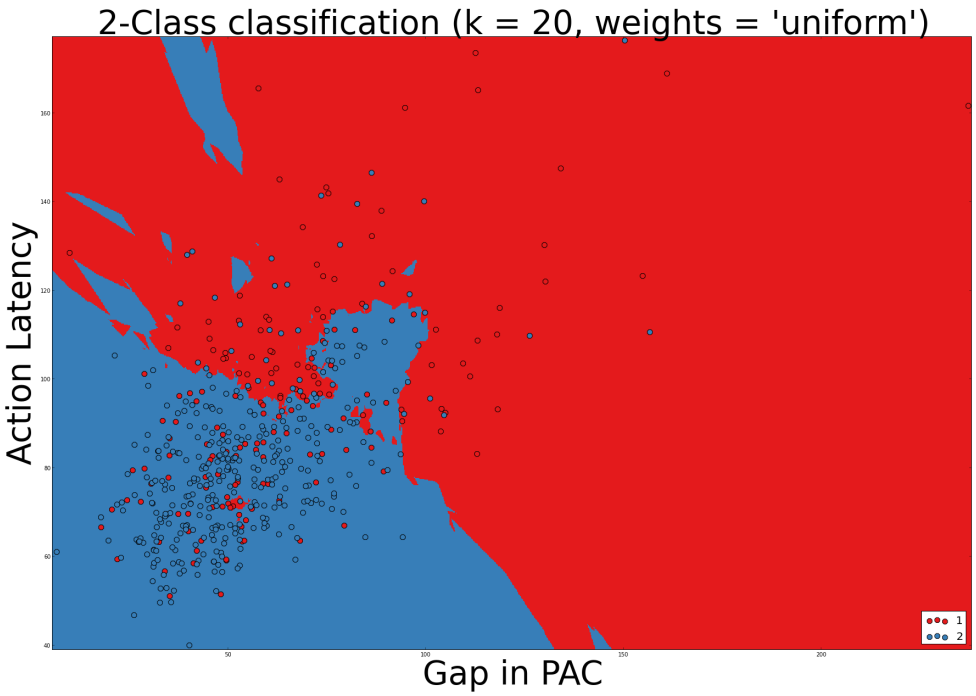
Important Variables for Each Matchup

1v2	APM	SelectByHotkeys	TotalHours
1v3	APM	NumberOfPACs	TotalHours
1v4	APM	ActionLatency	TotalHours
1v5	APM	ActionLatency	TotalHours
1v6	APM	TotalHours	ActionLatency
2v3	NumberOfPACs	TotalHours	APM
2v4	ActionLatency	TotalHours	APM
2v5	APM	ActionLatency	TotalHours
2v6	APM	TotalHours	ActionLatency
3v4	ActionLatency	TotalHours	MinimapAttacks
3v5	ActionLatency	SelectByHotkeys	TotalHours
3v6	ActionLatency	TotalHours	SelectByHotkeys
4v5	ActionLatency	APM	TotalHours
4v6	ActionLatency	TotalHours	SelectByHotkeys
5v6	ActionLatency	TotalHours	AssignToHotkeys

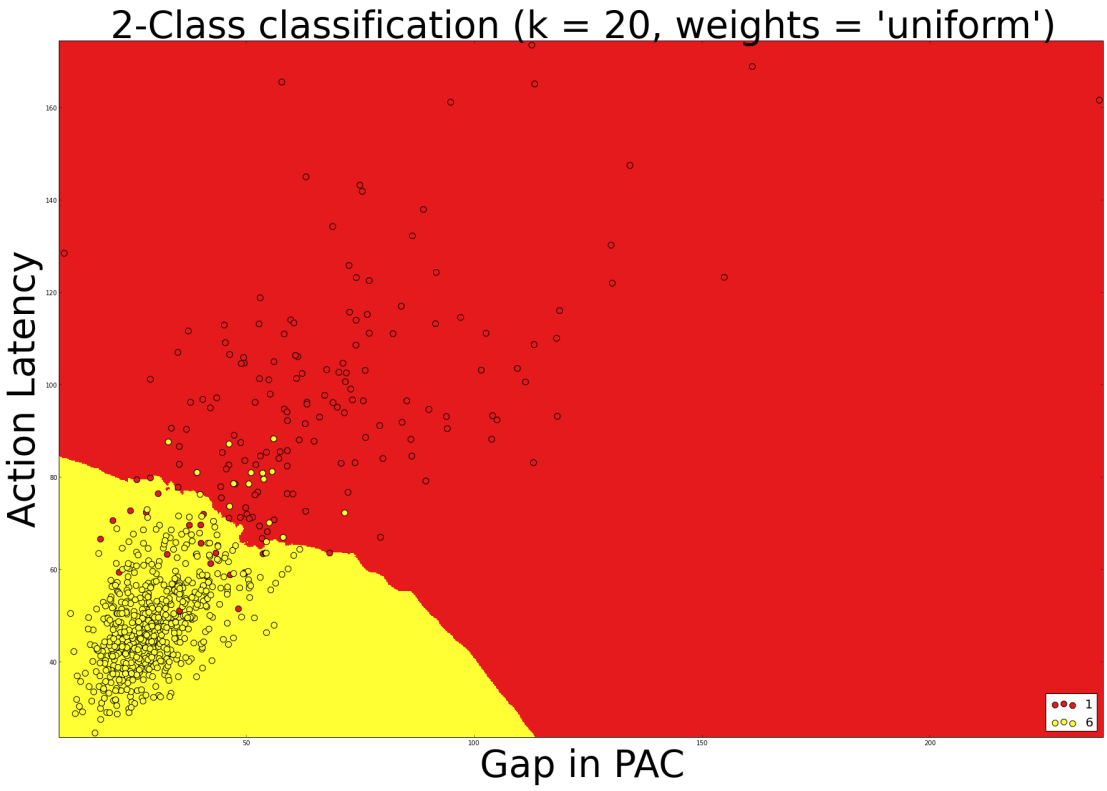
Table 1. Variables are arranged so that importance decreases from left to right. We see that in the lower leagues, APM is deemed most important. As we go up to higher leagues, ActionLatency becomes more important. This makes intuitive sense because lower players are still learning the mechanics of the game while at the higher levels, players have more or less the same level of mechanics; it is more a question of reflex times and map awareness that make an impact.

Part IV

K Nearest Neighbors



The point of this graph is to show you that when you are looking at two classes that are right next to each other, then you see a lot of overlap in the data. Here, the red color is class 1 (bronze) and the blue color denotes class 2 (silver). It would seem that KNN will not really do a good job of accurately being able to distinguish between any of the classes that are one apart. This should be pretty obvious, since players from classes right next to one another probably have similar skill levels, and perhaps have not converged to their true league yet.



This graph is supposed to be parallel to the one previously, but now we are looking at two classes furthest apart from each other. You see that this time the boundary does a much better job of identifying the two. Here, the red color is class 1 (bronze) and the yellow color is class 6 (diamond). Even with the classes being very far from each other, there are still some data points that are categorized wrong. What this suggests is that perhaps these two variables are not enough to distinguish between players, and are not a great representative of player skill. Our actual model uses more than just these two variables, so it is okay for this to not be perfect at classifying here.

KNN Confusion Matrix

	Pred=1	Pred=2	Pred=3	Pred=4	Pred=5	Pred=6
League=1	1.1	1.5	1.5	0.9	0.0	0.1
League=2	0.7	2.8	3.2	3.1	0.6	0.1
League=3	0.2	1.5	5.2	6.5	2.7	0.6
League=4	0.2	1.2	3.1	11.0	7.1	2.0
League=5	0.0	0.1	0.9	6.3	11.6	5.4
League=6	0.0	0.0	0.2	1.9	7.3	9.4

This is a confusion matrix to compare the values of the predicted league compared to the current league league of the player. The values insidethe matrix represent the percentage of the data that is in that category. So this means that any data in the diagonal of the matrix would be where our method is the same as the current league of the player and sums up to about 40% of the data. Also keep in mind that since the response variable is ordered, that the level of misclassification is different. For example, if we predict someone plays in the same way that players of league 6 do but in reality their current league is 1, then we are believing that the current ranking is extremely wrong. Since we can assume the current ranking israrely extremely off, we would want very few data points to be in the bottom left and top right corners of the matrix. This is the case, although we dotend to class people as being better (to the right of the diagonal) than their true league compared to worse (to the left).