CS57300 Homework	— Homework 5 , Problem 1 —	(April 30, 2016)
Shaobo Fang (0024260324)		

Q1:
(a). Describe which classi er is implemented in class MisteryClassifier. Describe the role of parameter nWL.
(b). The output plot seems deceiving. Explain which parameter combination should be the best and why theory backs up your answer.
(c). If we were to replace the decision tree classi er with a logistic regression classier, should we use strong (zero mean, low variance) or weak (zero mean, high variance) priors for the logistic parameters (i.e., strong or weak regularization)? Why?
A:
(a). This is a decision tree classifier. `nWL` is the number of week learners. Larger the number of `nWL`, more weak learners we will use to construct the final decision tree classifier. As the decision tree we are working on is essentially a boosting, we aim to accurately classify based on the combination of many weak learners.
(b). The F1 score vs. nWL plot is shown in the Figure 1 below. Similarly, we use another set of training

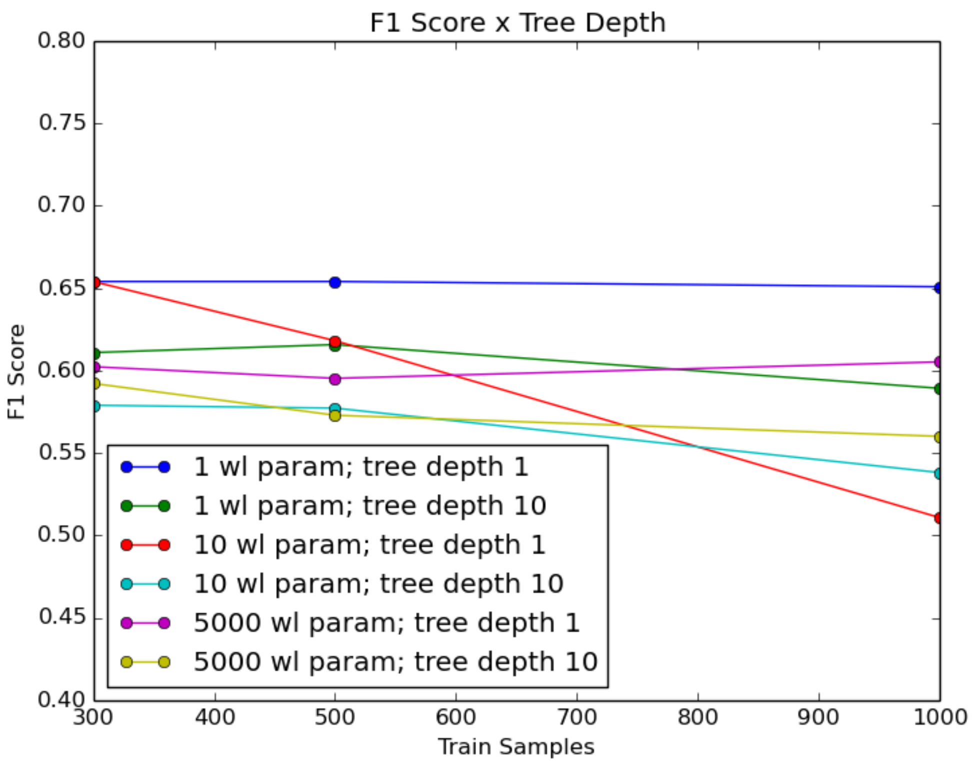

Figure 1: The F1 Score × Tree Depth with Training Size: [300, 500,1000]

data sizes as shown in Figure 2 and 3 below:

To obtain more convincing results, we conduct a few more experiment as shown in Figure **??**

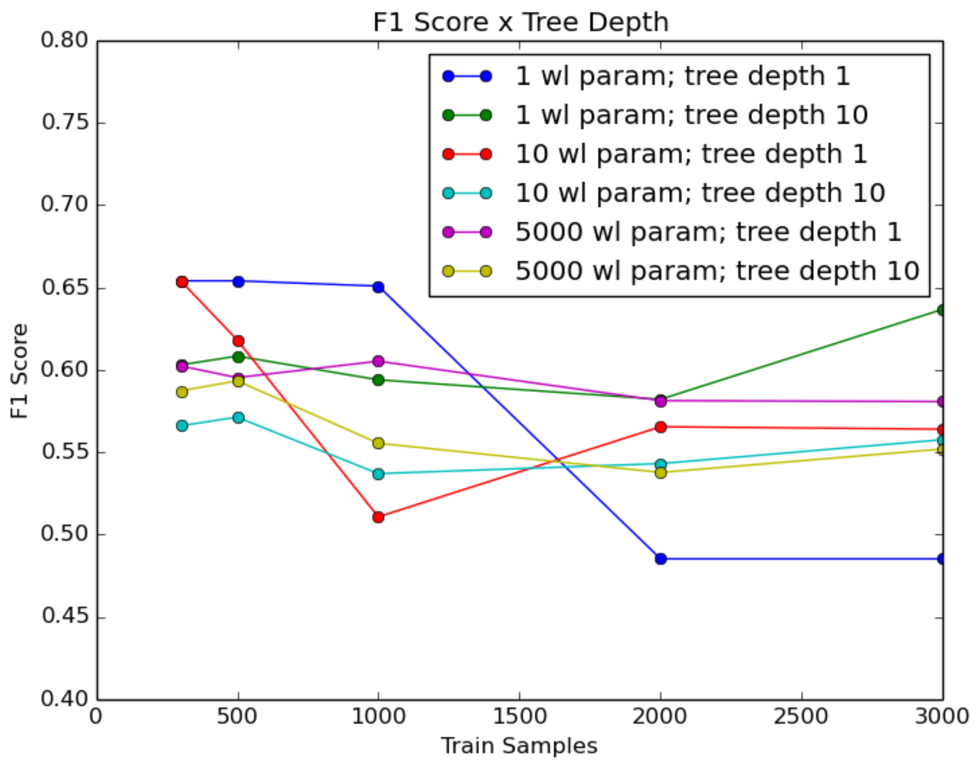

Figure 2: The F1 Score × Tree Depth with Training Size: [300, 500,1000,2000,3000]

Suggested by experimental results, the decision tree classifier can easily over-fit the data if the tree depth is large. This can be backed up by our observations that when the training data is a smaller size, the more complex decision tree (larger depth) always tend to perform worse compared to the decision trees that have the same number of weak learners. This also can be explained using the intuitives that: as our goal is to simply return a decision of 'yes' or 'no', the decision tree should not span very deep as eventually the end leaves should return either 'yes' or 'no' answer. **Hence, we believe that for the bank dataset the depth of the tree should be 1**.

However, setting the depth of the tree to 1 might introduce another problem, that as our model is relatively simple, the decision tree is too biased for our data.

Luckily, as we are working with boosting, whose advantage is to counter-act the overly bias model toward the data, so that we can actually resolved the problem. As shown in the Figures above, we have seen that larger number (e.g. nWL = 5000) of weak learners combined with a relatively simple decision tree (depth = 1) returns very stable suggested by Figure 4 and Figure 5 results and when the we have sufficient data size, the decision trees with large nWL tends to yields a reasonable results.

Suggested by Figure 2 and Figure 3 we have seen that with the same tree depth (Tree depth = 1), with larger number of weak classifier (nWL = 5000) the F1 score is more consistent across all the training data size and out-perform most of the decision tree classifiers with other parameters.

Hence, our final decision is that the best classifier is the one with tree depth = 1 and nWL = 5000. This can be backed up by theory that we first create a decision tree that is biased toward our bank dataset (model might be overly simple with tree depth set to 1) and then use large number of weak classifiers to counter-act the weakness of biased classifier.

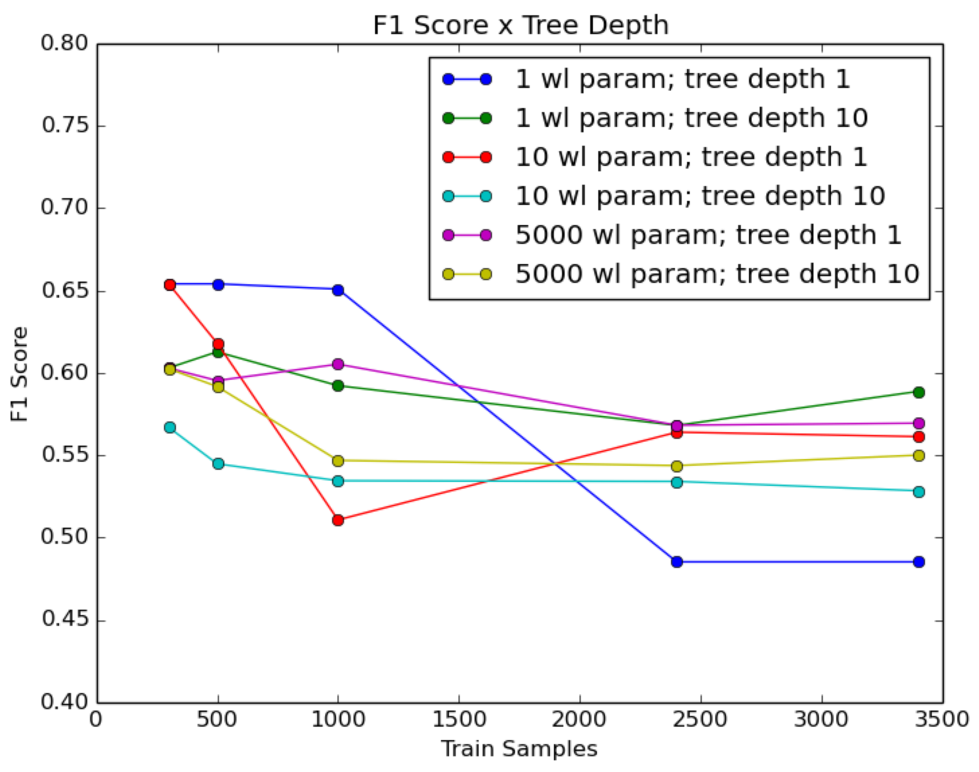

Figure 3: The F1 Score × Tree Depth with Training Size: [300, 500,1000,2400,3400]

(c). We apply what we have concluded in the previous problem, if we replace the decision tree classifier with a logistic regression classifier, we should use weak priors (zero mean, high variance) for the logistic parameters (weak regularization). The main reason is that as we have shown in previous part, the classifier can come with the problem of bias when fitting the data (the problem was solved by using many weak learners). If we use strong regularization, then basically we are creating even more bias hence the performance will not be good. (This can probably also explained by intuitives that we should treat each bank customer as an individual instead of using strong biased view).

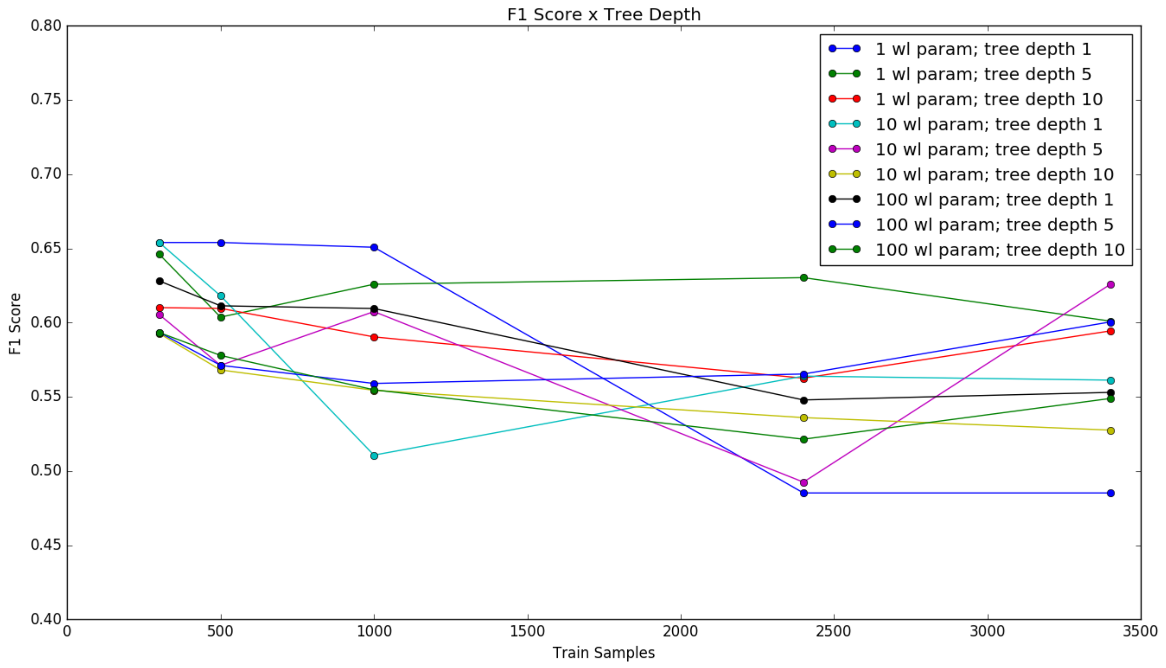

Figure 4: The F1 Score × Tree Depth with Training Size: [300, 500,1000,2400,3400], nWL = [1,10,100], Tree Depth = [1,5,10]

CS57300 Homework	— Homework 5 , Problem 2 —	(April 30, 2016)
Shaobo Fang (0024260324)		

Q2:
(i). Provide 6 plots, one for each of the components.
(ii). Entries with non-zero values of the rank-one component, belong" to that 3-mode community. Give:
(a) the count of src-IPs in this community
(b) the src-IPs of the community
(c) the count of dst-IPs in this community
(d) the dst-IPs of those users
(e) the ports used in this community
(iii). Suppose we add a time mode to our tensor using the timestamp of the connections. Describe how the PARAFAC tensor decomposition can help better identify botnets (botnet attacks tend to be synchronized).
A:
(i). We set the the parameters to compute for PARAFAC tensor decomposition to be 2. The six plots for components $\vec{a}_1$, $\vec{a}_2$, $\vec{b}_1$, $\vec{b}_2$, $\vec{c}_1$, $\vec{c}_2$ are as followed shown in Figure 6: As Tensor is similar to SVD, we are basically analyzing the components (or referred to "community", as described in part (ii)). As a high value $\vec{a}_{1,i}$ means that source-IP i participates in the first "concept" (or "triple" (three factor vectors)). Apply the same terminology, we determine that:

1. $\vec{a}_1$: Those $\vec{a}_{1,i}$ that are non-zero are those Source IPs i that participate in "concept 1" (or event 1.)
2. $\vec{b}_1$: Those $\vec{b}_{1,j}$ that are non-zero are those Destination IPs j that participate in "concept 1".
3. $\vec{c}_1$: Those $\vec{c}_{1,k}$ that are non-zero are those Ports numbers k that participate in "concept 1".
4. $\vec{a}_2$: Those $\vec{a}_{2,i}$ that are non-zero are those Source IPs i that participate in "concept 2" (or event 2.)
5. $\vec{b}_2$: Those $\vec{b}_{2,j}$ that are non-zero are those Destination IPs j that participate in "concept 2".

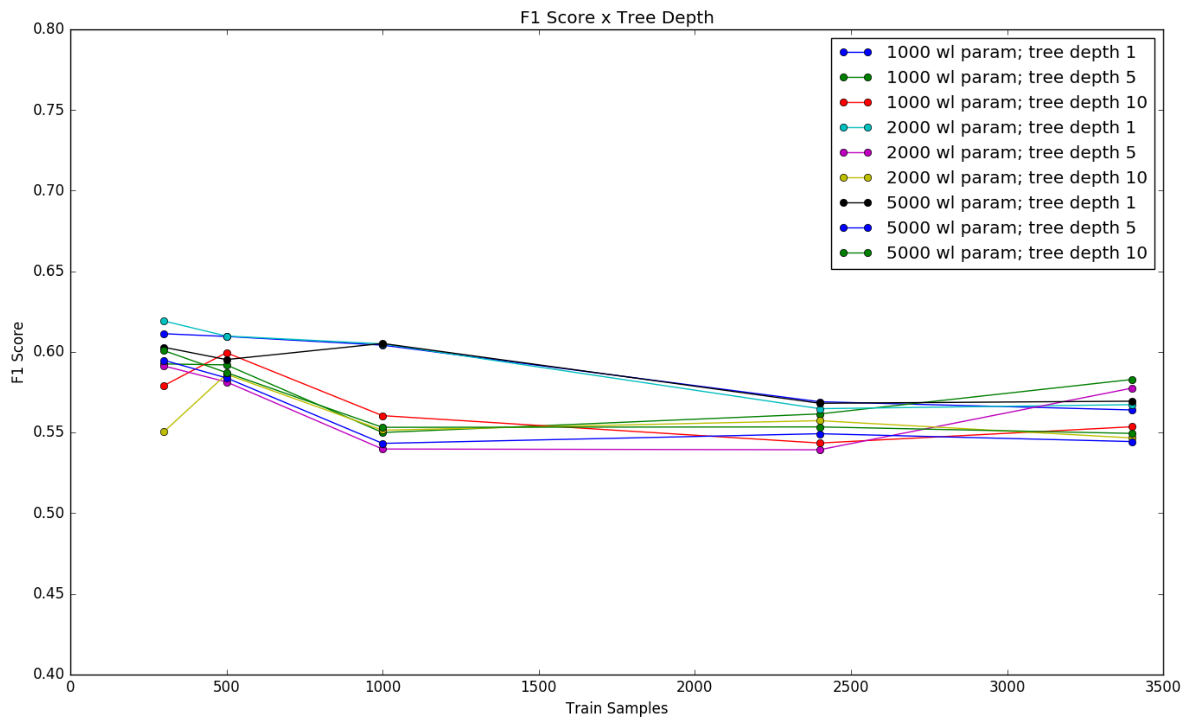

Figure 5: The F1 Score × Tree Depth with Training Size: [300, 500,1000,2400,3400], nWL = [1000,2000,5000], Tree Depth = [1,5,10]

6. $\vec{c}_2$: Those $\vec{c}_{2,k}$ that are non-zero are those Ports numbers k that participate in "concept 2".

(ii). As we are only considering the first component $(\vec{a}_1, \vec{b}_1, \vec{c}_1)$, we count those entries with non-zeros of the ran-one component ("belong" to that 3-mode community).

1. (a). The count of the src-IPs in this community is: 10 (by counting non-zero entries in $\vec{a}_1$)
2. (b). The src-IPs are [6, 7, 8, 20, 21, 23, 24, 25, 27, 30] by examining the $\vec{a}_1$ plot. (index associated with the non-zero entries).
3. (c). The count of the dst-IPs in this community is: 11 (by counting non-zero entries in $\vec{b}_1$)
4. (d). The dst-IPs are [2, 5, 7, 8, 9, 12, 20, 25, 29, 30, 33] by examining the $\vec{b}_1$ plot. (index associated with the non-zero entries).
5. (e). The ports used in this community is [1, 3, 4, 5, 8, 13, 14, 15, 19, 21, 23, 28] by examining the $\vec{c}_1$ plot. (index associated with the non-zero entries).

Note that we have noticed that for each execution of the Tensor decomposition the results might be different (the order of "concept" flipped, or the +/− sign flipped), hence the counting is only for the plots that we show above.

(iii). If we add the time mode to our tensor using the timestamp and connections. The PARAFAC tensor decomposition can be used to help better identify the botnets attack. For the timestamp that we have botnet attacks, we should be able to observe non-zeros values associated with a time stamp index.

Explanation: If we add the time stamp into our tensor, our tensor will have the 4th dimension. We are able to decomposite the tensor into 4 vectors instead of 3 in current example, for each concept. One vector associated with src-IPs, one vector associated with dst-IPs, one associated with ports number and the final one associated with time stamp index. We examine the non-zero entries with the 4th vector. Those time stamp index associated with non-zero entries would be the time stamps that participate highly in the components, hence should be the botnet attach.

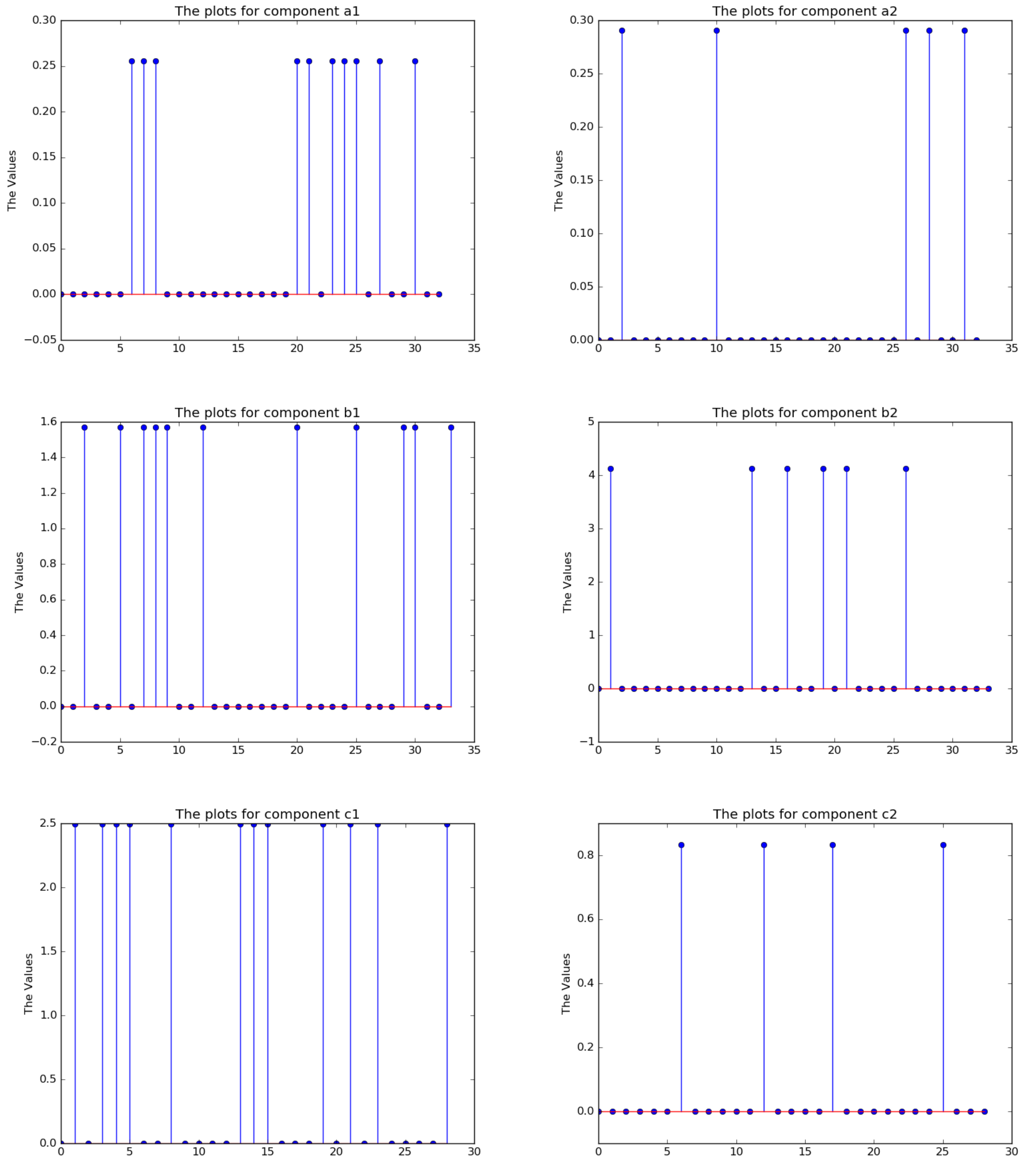

Figure 6: Plots one for each components $\vec{a}_1$, $\vec{a}_2$, $\vec{b}_1$, $\vec{b}_2$, $\vec{c}_1$, $\vec{c}_2$