Train a linear multi-class classification SVM with no kernel. Specify i) Your mapping function and ii) Your loss function

I trained a linear multiclass SVM using sklearn.svm.SVC package. From its user guide, they mapping function is:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i (w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$
$$\zeta_i \geq 0, i = 1, \ldots, n$$

The C and zeta are values that control the strength and distance of samples that are in the wrong part of the boundary because not all samples are perfectly separable.

Sklearn.svm.SVC uses a hinge loss function as their loss function.

Describe a method to estimate your performance using an empirical method. Compare this estimate with a well known theoretical bound. Explain why/if there is a difference.

To measure the error empirically, I used the model (trained using the training data) to predict the training data. I used this equation to find the error:

1 – correct/total

This gave me 12.81%. I am assuming this is the training error.

For the theoretical bound, I used PAC bound #2 and #3 to determine the min and max generalization error. I used the training data obtained from earlier and set delta equal to 0.05 (as the professor did in his lecture). This told me that the generalization error was between 0.08% to 48.27%.

The empirical data is in the range of the theoretical bound, but the error numbers mean a different thing. One depicts the training error and the other depicts the generalization error. Also, the generalization error has such a big range that it would encompass a lot of values.

Implement both types of transfer learning SVM (hypothesis and instance transfer) to train 1 vs 7 (target problem) by transferring in 1 vs 9 (source problem). Report your error estimate for the target problem with i) no transfer, ii) hypothesis transfer and iii) instance transfer. Which performs better? Why?

I implemented no transfer, hypothesis transfer, and instance transfer. Although I was able to extract the hyperplane, I was unable to use the hyperplane to create an SVM/boundary. Therefore, for hypothesis transfer I combined the data of 7's and 9's together and used the 1's data twice (although not exactly hypothesis transfer, it was kind of a work around). For instance transfer I used the support vector points obtained from 1 vs. 7 and moved them over to 1 vs. 9. The errors I received are:

No transfer: 29.99%
Instance transfer: 10.44%
Hypothesis transfer: 13.12%

Obviously hypothesis transfer and instance transfer performed better than no transfer. The way I computed hypothesis transfer was very similar to instance transfer (moving points from one to the other). Because I only moved support vector points from instance transfer, all those points were guaranteed to be in the correct spots but moving all the points also moved the errors. This could be why hypothesis transfer was worst for me.