**Olfaction Prediction Challenge**

Author: Amit Dhurandhar, Affiliation: IBM Research

**Summary Sentence**

For both sub-challenges and all descriptors linear regularized models were used with scaling (viz. cubic correction) of these predictions, with the details being quite different for the three types of targets (Intensity, Valence, Others).

**Background/Introduction**

Even before we get to the targets just analyzing the inputs I noticed an interesting pattern. If you sequentially partition the training set into 5 parts so that each is of size ~ 69, the same as the LB and FT sets, and perform an SVD on each part you notice that the most prominent eigen vector is in the same direction for all the parts including the LB and FT sets. In fact, the eigen values are also almost identical. The remaining eigen vectors are almost orthogonal in the different parts including the LB and FT sets, with the eigen values being orders of magnitude smaller that the largest eigen value. This means that most reasonable methods that don’t overfit will capture majority of the signal, with the best methods modeling the residual information in a more intelligent way.

Given the dataset size relative to the feature space the following modeling decisions intuitively make sense and also in my experience have found to work the best.

1. *Regularization:* Keeping coefficients or number of predictors small seems necessary (L2 for intensity, L1 for valence and L1/2 for others).
2. *Parameter Tying:* Fitting independent models for each individual even for the 1st sub-challenge is not the best. Some constraining of the parameters (ideally) across individuals or at least constraining the predictions to be close to the mean seems necessary for good performance.
3. *Weighting:* Based on the above discussion about trying to best model the residual information taking a cosine distance between the training and test sets and weighting each example accordingly seems to help a bit.

I have tried to incorporate as many of the above ideas as possible for predicting the 3 sets of predictors given limitations in time and consequently the methods I was able to code or obtain. Hence, I have either used a subset of the above ideas for certain predictors while trying to explicitly enforce certain ideas for others where implicit/composite methods were not readily available.

**Methods**

Rather than distinguishing between methods for each sub-challenge I will distinguish between methods for different targets as there is more distinction there than in the prior. For all the methods the *preprocessing was done exactly the same way as the organizers did in their baseline model*. Moreover, the outputs from the specific models were fitted to a cubic polynomial as was done for the baseline model. The main point of this seems to be to increase the dynamic range of the predictions in either direction. I also used 6 fold cross-validation for selecting the parameters so that each fold is roughly 69 instances which is the size of the final test set. This probably helps in choosing better parameters as each validation set in cross-validation is the size of the test set.

**Intensity**

Predicting intensity seems to be a complex task as the best models seem to be those that consider all the variables. Of course regularization is necessary but spreading the weight across multiple predictors seems desirable than betting on just a few. Hence, ridge like methods seem to be preferable here as opposed to lasso. However, as mentioned in the previous section some parameter tying seems essential and hence I used a cool multitask learning method described in [1]. The idea here is to learn a different L2 regularized linear model for each individual but to make sure that the model parameters across the individuals are not too far apart by L1 regularization. In other words, a model that is in between building independent models for each individual and building a model to fit the mean (this is subchallenge 2). Using this model I was able to obtain predictions for the 49 individuals. The mean of these predictions for each molecule across the individuals was used for both subchallenges. The standard deviation was used as an estimate for the intensity standard deviation in subchallenge 2. *It very weird here that actually building a model and fitting to the intensity standard deviation leads to much worse results.* The best results by direct fitting are if I use a highly non-linear technique such as Random forest but my strategy gave better results in terms of the LB set and so I used that.

I am thus using ideas a) and b) from above with no way of implicitly enforcing c).

**Procedure to obtain results**

1. Run *mtfl\_example.m* (syn3847222) which outputs two files each of which is the learned parameter vector W for mean and std respectively.
2. Run *Predict\_with\_W.m*(syn3847224) for each of these files by changing the input file for W in it. This will output a 69x49 file of predictions for each individual.
3. Calculate the mean across the 49 individuals if this is the W for the mean else calculate the std across the individuals.
4. Use them as predictions for subchallenge 2.
5. For subchallenge 1 just repeat the subchallenge 2 prediction for each individual.

**Valence**

Predicting valence in stark contrast to intensity seems to be a low dimensional function of the molecular descriptors. Here I found the best methods to be the ones that are highly selective and aggressive in feature selection such as lasso or L1 regularized least squares. Hence, for both subchallenges I used lasso.

For subchallenge 2 I directly applied lasso to fit to the mean. For subchallenge 1 I tried to explicitly enforce b) as I wasn’t able to obtain a multitask learning method similar to predicting intensity but with L1 penalty. Hence here I first fitted a mean model like in subchallenge 2 and then for every individual I fitted a lasso to the residual of that persons rating to the mean model, where the max features for any individual was limited to a 100.

Predicting standard deviation for valence is highly non-linear and I used the following “crazy” procedure to get decent results. For standard deviation I first fitted a stepwise regression model with cubic correction to the 5th power of the target, just so that there is enough separation to model. Lower powers also gave decent results just that this gave the best. Then I trained the baseline model. I took the min of my predictions and the baseline predictions on the test set (tstmin) and the max for training set (trmax). Then for each prediction if tstmin < trmean (training mean) I scaled down the prediction by minimum of the training set value divided by the minimum of the trmax. I did exactly the analogous procedure if tstmin value was greater than trmean, that is scaled it up by the max of the training set value divided by the max of trmax. This highly non-linear scaling seemed to give reasonable correlation for std of valence.

**Procedure to obtain results**

1. Run *lasso\_or\_elasticnet.m* (syn3847278). The 2nd column in the outputted file is the prediction for Valence mean in subchallenge 2.
2. Run *lasso\_or\_elasticnet\_per\_subject\_V\_residual.m* (syn3847280). The output is a 69x49 file which are the predictions for valence in subchallenge 1.
3. Run *StepwiseReg\_std.m* (syn3847320). Then run *ScaleVstd.m* (syn3847318). The output file has the predictions for std for each molecule.

**Others**

For the other odors on average a sparse but not as sparse a model as we have seen for Valence seems to work the best. Thus, a L1/2 regularizer closer to L1 seems to be the best. We use c) and the mean model seems to be the best for both subchallenges after explicitly trying to perform b). If I had more computational resources/time I would have tried more individual specific variants than the ones I did.

For standard deviation a lasso fit seems good. This was the only place where I found a consistent procedure good to predict standard deviation based on correlation.

**Procedure to obtain results**

1. Run *lasso\_or\_elasticnet.m* (syn3847253) in the Others folder. Note that the settings are different here than for valence. Columns 3-21 in the outputted file are predictions for the mean of the 19 descriptors respectively for subchallenge 2. Repeat these predictions for each individual thus obtaining results for subchallenge 1.
2. Run *lasso\_or\_elasticnet\_std.m* (syn3847255)**.** Columns 3-21 in the outputted file are predictions for the std of the 19 descriptors respectively for subchallenge 2.

**Discussion**

I think beyond the challenge models that implicitly model a), b) and c) for all targets might be interesting. Moreover, more analysis regarding how to combine the individual models with the mean model for the Other predictors would be interesting. A residual model might be good here as I did for valence. For intensity there is an interesting paper [2] published this year, which talks about fitting a quadratic model to the intensity squares rather than a standard linear model to the intensity. I briefly explored this w/o much success, but it is definitely something that can be looked at in more detail.

Using mean squared error is probably a better metric for the standard deviations of at least intensity and valence than zscore as the methods I chose to predict standard deviations of intensity and valence have high mean squared error (MSE) but somehow have reasonable correlation. Most of the standard linear methods have relatively low MSE but also very low correlation.

On the whole the challenge was quite enjoyable and I thank the organizers for that.

**Acknowledgment**

I would like to thank Abhishek Kumar for sharing with me the multitask learning code. I would also like to thank Kush Varshney for pointing me towards relevant papers and sharing the nuclear norm regularization code. Last but not the least I would to thank Pablo Meyer for answering my queries about the challenge as well as Bruce and the organizers for conducting a successful event.

**References**

1. A. Argyriou and T. Evgeniou. Multi-Task Feature Learning. Neural Inf. Proc. Systems, 2006.
2. L. Yan, J. Liu and D. Fang. Use of a Modified Vector Model for Odor Intensity Prediction of Odorant Mixtures. Sensors, 2015.
3. H. Zou, T. Hastie. Regularization and Variable Selection via the Elastic Net. Journal of the Royal Statistical Society, 2005.