Predicting contrast performance for the Gemini Planet Imager

Victoria Borish, Katherine Sytwu, Jean-Baptiste Ruffio

Stanford University
vborish@stanford.edu ksytwu@stanford.edu jruffio@stanford.edu
Project TA: Junjie Qin jqin@stanford.edu

I. Introduction

The Gemini Planet Imager (GPI) is a new exoplanet-hunting instrument mounted on the Gemini South telescope in Chile. Exoplanets, or planets orbiting stars other than the sun, are detected by masking out the light of the star at the center of the image, allowing one to detect objects in its neighborhood. Detection of orbiting planets is often inhibited by noise coming from the turbulence in the atmosphere, which allows light to escape the mask. The instrument tries to actively correct for these perturbations but it is still imperfect, making it difficult to detect the faint exoplanets. There are nights when the atmospheric conditions are such that it is impossible to obtain any usable data, so trying to observe then is a waste of valuable telescope time. There currently is no reliable method to predict the performance of the instrument[1].

A quantitative measure of whether or not an exoplanet is detectable is given by the **contrast**, or the ratio of the brightness of the exoplanet to that of the star it orbits. The target variable for each image is the threshold for the lowest possible contrast of a planet that could be viewed on a given night. Therefore, we want these values to be as low as possible to allow for viewing of faint exoplanets.

We can approach our prediction in two ways. One way is to predict a numerical contrast value using regression and then allow the researchers to decide whether it is worth observing or not. In this case, we decided to use locally weighted linear regression (LWLR) to predict the actual contrast value. The other way is to treat this as a classification problem by predicting if an image will be good or bad on a given night. For this, we used a Support Vector Machine (SVM) algorithm with a Gaussian kernel to solve for a target labeled as either good or bad. In both cases the features are the environmental variables recorded by the telescope for each observation[2]. It includes variables such as— but not limited to— the star brightness, wind direction and turbulence.

The advantage of predicting a numerical contrast value is that it also provides us with a typical performance for the instrument under current conditions. This information can be used to determine if the instrument is doing better or worse than usual in similar conditions. The classification approach might seem less powerful, but it has the advantage of being a simpler problem and therefore may give a more accurate answer.

Previously, our collaborators have examined correlations between the contrast and individual features, determining that performance is mainly driven by the star brightness and the adaptive optics wavefront error [1, 2, 3]. However, a prediction approach utilizing all of the features in a multi-dimensional problem has not been implemented or studied.

In Section II, we describe our data set and preprocessing steps. Then in Sections III and IV, we describe our prediction methods and results for a classification and regression problem, respectively. Finally, in Section V, we

[1]There exist indicators of the current observing conditions but they are not trusted for GPI. Indeed it is not clear what drives GPI performance because GPI utilizes active correction of the turbulence.

[2]The variables are saved as metadata in each image.

compare both methods, and in VI, we present ideas for future work on the project.

II. Training Sets

Our training set was obtained from the GPI database[3]. After filtering out training examples with missing data, two types of training sets were extracted. The first,"raw data", contains 2908 training examples in which an example corresponds to a single 60 second exposure. The second training set, "processed data", considers processed images (images resulting from combining many single exposures) and consists of only 183 training examples. For the latter, both the mean and the standard deviation of the raw features define the new set of features. The processed data represents the data from the astronomers' final images, and thus is what they actually care about, even though it yields a much smaller training set.

We considered the following features ("o" and "+" refer to the raw and processed dataset respectively):

- The **Seeing**$^{o+}$, a measure of the fuzziness of the image.
- The **characteristic time scale**^{o} of the turbulence.
- The **Airmass**$^{o+}$, the distance travelled by a light ray through the Earth's atmosphere.
- The **Wind Direction**$^{o+}$ and its **standard deviation**$^{+}$.
- The **Wind Velocity**$^{o+}$ and its **standard deviation**$^{+}$.
- The **Wavefront Error**$^{o+}$, a measure of the perturbation of the incoming light, and its **standard deviation**$^{+}$.
- The **Apparent Magnitude**$^{o+}$, the star brightness using a logarithmic scale.

The previous features were suggested as potentially interesting by our collaborators. Then a forward search, described in Section III was applied to extract the optimal subset of these features for the given training set.

III. Classification

For the classification problem, we first need to set a threshold for the contrast, above which we know the conditions won't allow for a research-quality image. From previous observations, we know that this threshold lies around 6×10^{-5}. This choice led to a skewed dataset with around 80% of the training examples classified as "good".

We decided to use a ℓ_1 soft margin support vector machine which calculates a decision boundary whose shape is determined by a kernel and a small number of training examples close to the boundary. Since our data is not linearly (or polynomially) seperable due to the high amount of noise, we chose a soft margin algorithm, given by **Equation 1**:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, i = 1 \dots \text{m} \\ & \xi_i \geq 0, i = 1 \dots \text{m} \end{aligned} \tag{1}$$

Here, w and b are parameters that define the decision boundary. $(x^{(i)}, y^{(i)})$ are the m training examples where x is the feature vector and y the target. The $\{\xi_i\}$ are error parameters that allow some of the data points to be slightly across the decision boundary, something that is necessary with our noisy data. The penalty coefficient, C, accounts for the noise in the data by determining the relative weight given to each of the two objectives. Thus C is related to the number of training examples we allow the model to classify wrong.

The first three parameters are computed by the SVM, but C is chosen before the SVM is run. We optimized over C by running the SVM on the training set with various values of C ranging from 0.1 to 10. For each value of C, we averaged over many runs, each with a different random test set (and the remaining data as the training set). We then chose the value of C that minimized the average generalization error.

[3]The features and targets were obtained either through a SQL query to the database or by reading manually the images metadata.

In addition to optimizing C, we needed to optimize the features we used to train the model. We used forward feature selection, beginning by iterating through the features and training the data on a set consisting of only a single feature. We then chose the feature that produced the smallest generalization error and added that to the list of features to keep. Continuing this for another iteration through the features meant we trained the data on sets consisting of all pairs of features in which one of the features was the good feature found in the first loop. We repeated this process until we had gone through all the features, adding them one by one, and plotted the generalization error vs. feature. For each run of the SVM, we optimized over the C value.

Figure 1 shows the result of the forward feature selection, where we decided to keep the features selected until the generalization error began to increase. This left us with an optimum training set consisting of 4 features: star magnitude, seeing, wavefront error, and wind direction.

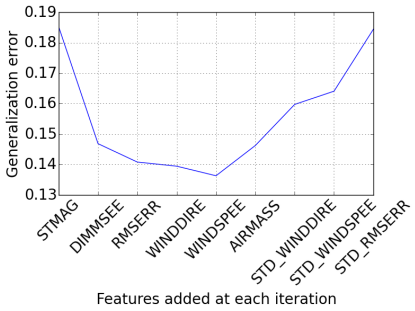

Figure 1: *Forward seach feature selection for the processed data.*

We chose a SVM because it's one of the best off-the-shelf classifiers which would give us a decent preliminary result. Using Python's scikit-learn [4] library, we ran our SVM with a Gaussian kernel because it allows for non-linear decision boundaries and high flexibility. Since our features have unknown relationships to the contrast, the Gaussian kernel would not make any strong assumptions.

We divided our processed (raw) data set into a test set of 10 (300) examples and a training set of variable size from 10-180 (100-2800) examples. We then studied how generalization and training error depend on training set size to better understand our model and error. **Figure 2** shows both the test and training error for the SVM as a function of training set size for the processed data.

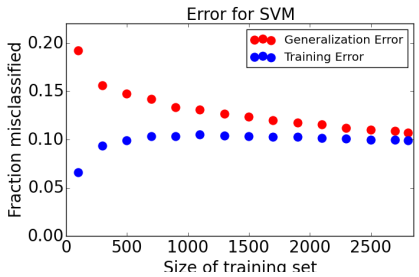

Figure 2: *Generalization and training errors of the processed data for the SVM vs. training set size.*

In **Figure 2**, we see that the generalization error has not quite leveled off at the largest available training set size. This leads us to believe that the best generalization error (13.5%) could still be improved upon with a larger set of data. The training error, however, begins to level off at a value of around 10%. The asymptote of this being non-zero implies that this SVM cannot perfectly classify our entire data set. We plot here the error for the processed data set because that is ultimately what is important to the astronomers. However, when we initially tested the data on the raw data set, we found an error as low as 10.5%. Our error is still larger than we would like it to be, but lower than the error achieved when using regression (see next section).

To understand our error, we calculated the confusion matrix (see **Figure 3**) and found that SVM did extremely well for the good contrast values (classifying 99% correctly for the processed data), but had a large percentage of false positives (62%). The raw data did better on classifying the bad nights, and understanding this and trying to correct for it is still an open problem.

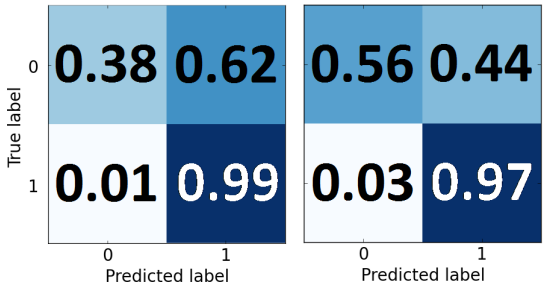

Figure 3: *Confusion matrix for SVM. Left: processed training set. Right: raw training set. 99% (97%) of the good nights were correctly classified as opposed to only 38% (56%) of the bad nights for the processed (raw) case.*

IV. Regression

Since our data is extremely noisy, linear regression did very poorly and did not improve upon the addition of higher and lower power features. Instead, we want a model that is just as flexible as SVM and makes few assumptions about our data. Locally weighted linear regression satisfies both, as it only assumes two things: local linear dependence of the features and little dependence on training examples far away from the test example in feature-space. We decided to use Gaussian weights for our model for the same reasons as stated for the SVM model. For any test example x, the weight of each training example $x^{(i)}$ is given by

$$w_i = \exp\big(-\gamma \|x - x^{(i)}\|_2\big) \tag{2}$$

where γ is a parameter we choose that controls the relative weighting of nearby training examples.

We can directly solve for the predicted contrast $\hat{y}$ given the test example x using a modified normal equation:

$$\hat{y} = \Big(\big(X^T W X\big)^{-1} X^T W y\Big)^T x \tag{3}$$

where X is a mxn matrix of the training examples (n is the number of features), W is a diagonal matrix with the weights defined in **Equation 2**, and y is a vector of the training target values.

We split up the data into the test and training sets as described in Section III and determined the optimal γ value in the same way that we determined C in the SVM model. We then similarly compared the generalization and training error as a function of training set size. Since our contrast values are small, we calculated percentage error, and found that LWLR, on average, with the processed data set gives a 2.5% error in the contrast value.

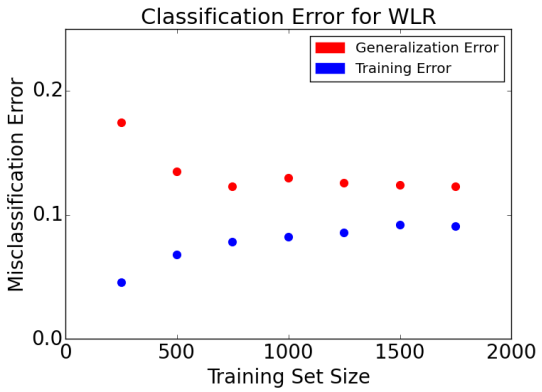

Figure 4: *Generalization and training errors of the raw data for the "classified" locally weighted linear regression vs. training set size.*

We can compare LWLR with our SVM model from before by converting our predictions into binary "good" and "bad" values using the same cutoff as we used for the SVM. The resulting generalization error for the processed data leveled off at around 19%, which is worse than the SVM model. However, we initially ran it on the raw data, achieving an error of around 13% (as seen in **Figure 4**), slightly larger than that of the SVM. LWLR relies on nearby data points, so it isn't surprising that the raw data does better than the processed data because it containes 10 times as many training examples. For both data sets, the generalization error leveled off, so more training examples probably would not improve our prediction.

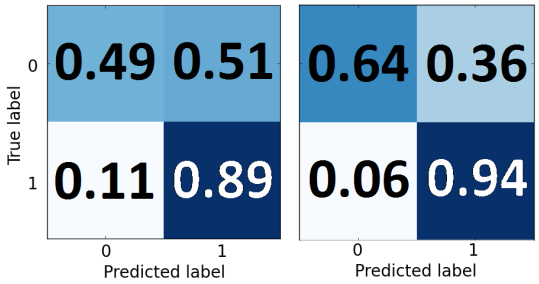

Figure 5: *Confusion matrix for LWLR. Left: processed training set. Right: raw training set.*

Figure 5 shows the confusion matrix for the regression on the processed and raw data. LWLR does a better job classifying the "bad" data (0) than SVM; however, it does worse at classifying the "good" data (1). The improvement in classifying the bad data is most likely due to the extra information on the contrast values during regression.

V. Discussion and Future Work

In addition to SVM producing a lower generalization error than LWLR for both the raw and processed data sets, it also performs better for our application because it has fewer false negatives. False negatives carry a much larger penalty than false positives, as false negatives would imply astronomers are not collecting data on a night when they could have gotten good data. The SVM currently predicts 99% of good nights correctly and still saves the astronomers from collecting data on almost 40% of the bad nights.

In order to try to improve the performance of both the algorithms, we characterized the test examples that were being misclassified. We first used principle component analysis, which projects our high dimensional data onto three orthogonal axes that capture the most variance. Our PCA decomposition, however, only captured about 65% of the variance, and the boundaries were hard to distinguish. We also plotted a histogram of the distance between each misclassified point and the decision boundary for the SVM and the absolute error for the regression. From our histogram, we saw that most of the misclassified points are close to the decision boundary and are not outliers, but that was all the useful information we were able to obtain.

Feature selection was our most successful strategy. We confirmed, as expected, that the features with the most predicting power are the star brightness and the wavefront error. However, the timescale of the turbulence was thought to better correlate with the contrast than the seeing which is contradicted by the present study[4]. We also found that the standard deviations of the wind speed and wind direction were important features in our prediction. These parameters hadn't been considered in previous studies.

VI. Conclusion

In conclusion, we have demonstrated the first machine learning approach to predict the worth of observing on a given night. Support vector machine classification gave better results than locally weighted linear regression, particularly with the processed data set (the data that reflect the final image the astronomers will use). Although the locally weighted linear regression did better at predicting the contrast on the bad nights, the SVM gave 1% false negatives, which is the most important figure, because astronomers never want to miss an opportunity to observe.

Before beginning, we thought the classification problem would work better than the regression problem because it's simpler, and that proved to be the case. With more time (or a rotation student), we hope to use the framework we have created and the knowledge of which features are the most helpful to apply this problem to models designed for more complicated systems, such as neural networks.

[4]The more accurate conclusion might be that the DIMM instrument that measures the seeing does a better job than the MASS instrument.

References

[1] Rajan, A., Patience, J., Nielsen, E. L., Wang, J. J., De Rosa, R. J., Macintosh, B., Graham, J. R. *et. al.* (2015). Gemini PLanet Imager Exoplanet Survey: A study of the contrast and planet detection sensitivity *Poster at Spirit of Lyot conference*

[2] Rajan, A., Nielsen, E. L., Wang, J. J., De Rosa, R. J., Macintosh, B., Graham, J. R. *et. al.* (2015). GPIES Planet Yield Simulation Update *Poster at Spirit of Lyot conference*

[3] Poyneer, L. A. *et. al.* (2015). Performance of the Gemini Planet Imager's adaptive optics system *Submitted*

[4] Pedregosa, F. *et. al.* (2011). Scikit-learn: Machine Learning in Python *JMLR 12*, pp. 2825-2830