CS-433 Machine Learning Project 1

Matthias Minder, Zora Oswald, Silvan Stettler

***Abstract*—**

Introduction

Collision events in the Large Hadron Collider at CERN do not create directly observable results. An essential part of elementary particles such as the Higgs boson therefore rely on classifying the collisions based on a number of variables that can be measured in the collider. In our dataset, a vector containing 30 dimensions describes one collision event. A binary classification algorithm can be used to predict the presence of a Higgs boson based on this feature vector. The model is trained with data from collisions for which the existence of a Higgs boson is determined. Our classifications of the collision events consist of three main steps:

1) Normalization and imputation of missing data
2) Parameter optimization using cross-validation
3) Predictions on test data using the best-fitting binary model

Furthermore, the importance of variables for the final model was assessed using best subset selection.

Methods

As a first step, all features were transformed to have mean zero and unit variance, disregarding missing values.

As a second step, missing values were imputed. These missing values are due to physical measures being made impossible under certain conditions. However, since the classification methods used in the scope of this project don't naturally support the presence of missing values, they were imputed using the following linear regression approach:

Analysis of the raw data showed that there are a total of six distinct patterns of missing features. A pattern of observation $\boldsymbol{x}$, denoted $P(\boldsymbol{x})$, is characterized by values of $\boldsymbol{x}$ missing in dimensions M and values being present in dimensions M'. Approximately 60'000 observations of the training data contained no missing values, i.e. $M_{P(\boldsymbol{x})} = \emptyset$. For each incomplete pattern $P_i(\boldsymbol{x})$ and every missing feature $l \in M_{P_i(\boldsymbol{x})}$, a linear regression was fitted to complete observations, taking all features $k \in M'_{P_i(\boldsymbol{x})}$ as observations and feature l as response. This fit was then applied to impute the missing feature l of all $\boldsymbol{x}$ corresponding to $P_i(\boldsymbol{x})$. All linear regression models were fitted using gradient descent. Missing values of the test data were imputed using the fits on the training data.

This method for missing value imputation was chosen because it follows the natural structure of the data: Observations corresponding to the same physical preconditions leading to a specific pattern of missing values were all subjected to the same fit for missing value imputation. Simple linear regression was chosen due to its easy interpretability. Fitting more complex regression models would only make limited sense, since the absence of ”true” values for a given missing value pattern makes reasonable model comparison impossible. Finally, we chose this approach over ”simple”, constant imputation using the feature mean or median because it allows to capture more of the data variability.

However, by imputing missing values, the information about their underlying physical reasons is lost. To capture this information, dummy variables were created that encode every non-complete pattern of missing values.

Thereafter, two general model categories were fitted to the preprocessed data: L_2 regularized logistic regression and support vector machines. Gradient descent was used to minimize the cost function for L_2 regularized logistic regression that was shown in class.

In addition to logistic regression, the collisions were classified with a Support Vector Machine (SVM) method. The basic idea behind SVM methods is to find a hyperplane that separates the data-set into two classes. This can be achieved by minimizing a cost function based on hinge loss.

$$\mathcal{L}(\boldsymbol{w}) = C\sum_{n=1}^{N} max(0, 1 - y_n\boldsymbol{x}_n^T\boldsymbol{w}) + \frac{1}{2}||\boldsymbol{w}||^2 \qquad y_i \in -1, 1 \tag{1}$$

If the output is predicted on the right side of the hyperplane, meaning that $sgn(y_i) = sgn(\boldsymbol{x}_n^T\boldsymbol{w})$, then that particular observation does not contribute to the loss. Points on the wrong side of the hyperplane or too close to the hyperplane contribute, however. Therefore, the gradient of the hinge loss $\nabla\mathcal{L}$ either takes the value $\mathbf{0}$ or $-y_n\boldsymbol{x_n}$ plus the contribution of the penalty function $\lambda\boldsymbol{w}$. This expression for the gradient was used in order to minimize the SVM cost function by stochastic gradient descent.

The feature vector $\boldsymbol{X}$ that was used as input for both models contains a constant and polynomial basis expansion (degree 2) terms so that

$$\boldsymbol{x}_{enh.} = [1 \;\; \boldsymbol{x}_n \;\; x_n i x_n j] \qquad i \in dim(\boldsymbol{x}_n), j < i \tag{2}$$

Both L_2-regularized logistic regression and SVM depend on a hyperparameter which controls the penalization of false classification. This parameter was chosen to maximize accuracy, as determined in ten-fold cross-validation. The different classification methods were compared in terms of their achieved accuracy on an independent test set.

In order to gain insight into the decision making process of our classifier, a forward-greedy best subset selection was performed. The training set was randomly split into a training and a validation set containing 80% and 20% of the original training data respectively. Then, starting with an empty model (containing no features), the following procedure was repeated:

1) For every feature not yet in the model, the model plus that respective feature was fitted to the training set.
2) The accuracy of every fit was assessed on the validation set.
3) The feature, including which the greatest accuracy was obtained, was included into the model.

These three steps were repeated until the full model was obtained. This allows to obtain an importance ordering for the variables. The process was repeated ten times to assess result stability. Furthermore, the dummy variables were disregarded during this process in order to obtain an importance ordering of the original data. Finally, for performance reasons, the gradient descent method was run with less iterations and a larger step size.

Results

The following graphs show the accuracies obtained during cross-validation. The best results were obtained with enhanced data, SVM and λ smaller than 10^{-3}.

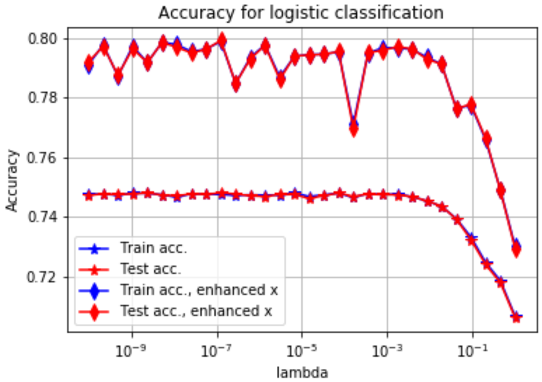

Figure 1. 10000 iterations, γ = 0.001

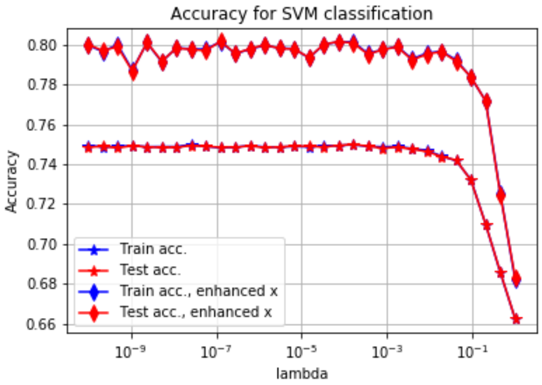

Figure 2. 10000 iterations, γ = 0.001

The fact that train and test accuracies are very similar and equally good for small λ shows that overfitting does not occur. It is rather probable that the model underfits and more training iterations could still improve the results.

Plot Number Accuracy vs no features (plotnr hie) suggests that around 15 features suffice for the model, adding additional features to the model leads to overfitting. However, recall that this plot was obtained by a less refined gradient descent than the final model. When we recreated a fit containing only 15 features with the refined gradient, the performance on the test set was worse than for the full model. This suggests that the overfitting phenomenon decreases as the SVM fit grows more precise. Plot Number HEATMAP shows that certain features are consistently included earlier into the model than others (WELI?), suggesting that they are much more important for the classifier than others.

Conclusion

We presented a classifier based on support vector machines and using polynomial basis expansions which outperformed all other classifiers examined in the scope of the project in terms of accuracy. However, only classifiers based on SVMs and logistic regression were assessed. A better performance may be achieved using other, more sophisticated classification models such as kernel SVMs, random forest or neural networks.

A downside of support vector machines is that they only output the class, but not a class probability. This makes it impossible to apply a more stringent cutoff for the detection of Higgs bosons in order to reduce the type I error.

Moreover, we assessed variable importance using forward-greedy best subset selection. (WAS GSEHT ME???)

References

[1]