Project 1 - finding the Higgs Boson

Léa Bommottet, Nicolas Brunner, Karine Perrard
CS-433 Machine Learning, EPFL, Switzerland

I. Introduction

In this project we have to solve a binary classification problem. In function of a vector of features regarding a proton collision event, we have to determine if this collision produced an Higgs boson or another process, there is at most 30 parameters provided for each classification. We have to implement and use machine learning methods, then feature processing and engineering to solve this problem.

II. Models and methods

A. Choice of Models

First of all, we have to select the core of our model. In this project, there is 6 possible models on which we can base.

1) Linear regression using gradient descent: This model try to minimize the error of the function, by moving the vector w in the reverse trajectory of the gradient.

$$w^{(t+1)} = w^{(t)} - \gamma \nabla L(w^{(t)})$$

$$e = y - Xw$$

$$\nabla L(w) = -\frac{1}{N} X^t e$$

2) Linear regression using stochastic gradient descent: The error function is defined as a sum over the training errors:

$$L(w) = \sum_{n=1}^{N} L_n(w)$$

As for gradient descent, we want to follow the gradient, but this time the updating step is done on the cost contributed by one of the training examples. We can also compute the gradient on a subset of the examples, this is mini-batch stochastic gradient descent.

$$w^{(t+1)} = w^{(t)} - \gamma \nabla L_n(w^{(t)})$$

3) Least squares regression using normal equations: We try to solve: $\nabla L(w^*) = 0$

Then we derive

$$w^* = (X^\top X)^{-1} X^\top y$$

and a unknown data-point would have

$$\hat{y}_m := x_m^\top w^*$$

4) Ridge regression using normal equations: We want to punish complex models and conversely choose simpler ones. In the case of ridge regression, we add a regularizer $\Omega(w) = \lambda ||w||_2^2$ in the quest to minimize w:

$$\min_w L(w) + \Omega(w)$$

the explicit solution of w become:

$$w^*_{ridge} = (X^\top X + 2N\lambda I)^{-1} X^\top y$$

5) Logistic regression using gradient descent or SGD: The problem is about separating the outputs into different labels. To do so we use the logitic function, that gives values in the range [0, 1]:

$$\sigma(z) := \frac{e^z}{1 + e^z}$$

Again we want to minimize the cost function defined as

$$L(w) = \sum_{n=1}^{N} ln[1 + e^{x_n^T w}] - y_n x_n^T w$$

To compute the model, we use gradient descent. The formula is

$$w^{(t+1)} = w^{(t)} - \gamma \nabla L(w^{(t)})$$

where

$$\nabla L(w) = X^T[\sigma(Xw) - y]$$

6) Regularized logistic regression using gradient descent or SGD: We can have troubles if the data is linearly separable. In that case, any vector **w** having the right direction is solution gives the minimum for **L(w)**. The goal of regularization is to add a penalty term, which is the length of the vector.

$$w* = argmin_w - \sum_{n=1}^{N} lnp(y_n|x_n^T, w) + \frac{\lambda}{2}||w||^2$$

The reason we based our model on logistic regression is because we have the least global error on the dataset. Fig.??

Furthermore, in function of the feature named "fit", some parameters are unavailable. Thanks to this important information, we don't train only one model, but 4, since "fit" can be 0,1,2,3. This idea reduce our global error, when compared to an single model without this separation. Fig. 1

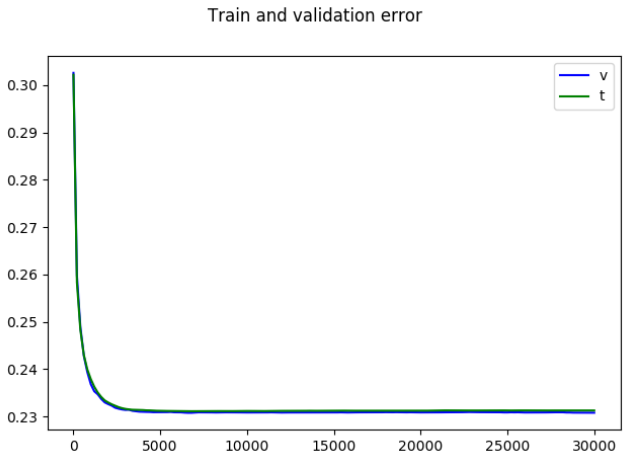

Figure 1. Train and Validation error with no separation according to number of iterations

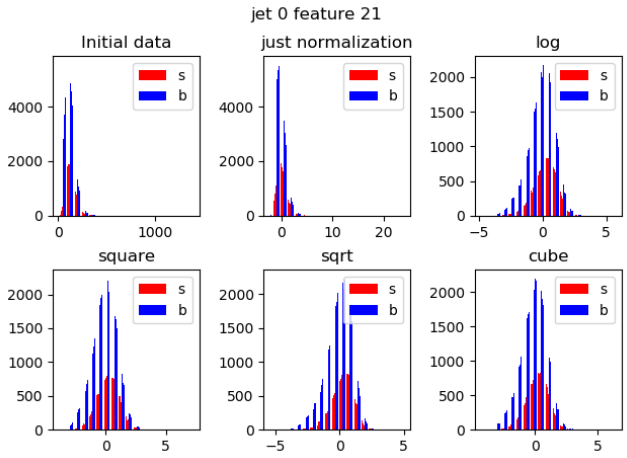

Figure 2. Function applied on the data of the feature 21

B. *Choice of parameters*

Blablabla Cross-correlation by K-fold:

As seen in Fig.??, the parameters that reduces most the train error are lambda = ??? and gamma = ???.

C. *Choice of features*

We can improve our model by adding new features, which are function of other parameters. We want more features that have a Normal distribution.

Natural logarithm
: Applying the logarithm on a parameter can highlight its Normal behavior. One example is this function on the feature 21 (Fig. 2). In this case, we notice a much nicer Gaussian distribution on the values, than the initial data. By doing an exhaustive research, we find that the useful parameters with this function are $\{0, 1, 2, 3, 4, 5, 8, 9, 10, 13, 16, 19, 21, 23, 26, 29\}$, when discarding the NaN value.

Square root
: Same as the natural logarithm, the square root can bring out a normal distribution. The feature 21 is also transformed with this function in our model (Fig. 2). Again if we don't consider the NaN value, the parameters $\{0, 13, 16, 21, 23, 26, 29\}$ have a Normal behavior under the square root

Threshold
: Some features are mostly distributed along two peaks (Fig. 3). In that case we move values to their closest peak to get two well defined peaks. This is the case for parameters $\{11, 12\}$.

Nothing max
: Some features have no defined distribution, but their values explode to big numbers. In that case we divide by maximum to get new value in range $[0, 1]$. Such distributed parameters are parameters $\{6, 14, 17, 24, 27\}$

Nothing norm
: Some features have no distribution and their values are already between 0 and 1. For parameter 7 we do nothing.

Distance
: When we plot 2 parameters in a graph, we can find some correlation between (Fig. 4). To reduces this property to a normal distribution, we compute the Manhattan distance, to flatten all the perpendicular result on the plot $y = -x$. The Manhattan distance is computed as follow:

$$dist(x_1, x_2) = |x_1 - x_2|$$

N^{th} power
: Some feature sometimes have a better behavior when they are raised to a power greater than 2, as it is the case for feature 19 when raise to the cube (Fig. 5)

III. Results

On compare entre model simple et le 4 jets.

On montre les graph features, pourquoi certaines sont sqrt, del, cube.

On compare avec et sans ces features en plus.

On montre notre erreur global, on parle de kaggle?

IV. Summary

Separating our model into 4 different ones was a great idea, reducing considerably our global error (Fig.??) in the same way that including more parameters (Fig.??). Those added criterion improve the quality of our model, adjusting from ??? to ???. 0.18% of error is kinda ok with the simplicity of the core models. If we want to improve it more, we would need to find more useful parameters to boost our data.

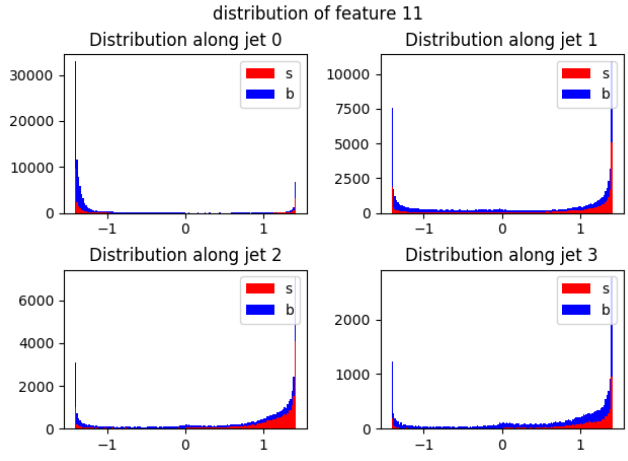

Figure 3. Distribution for feature 11

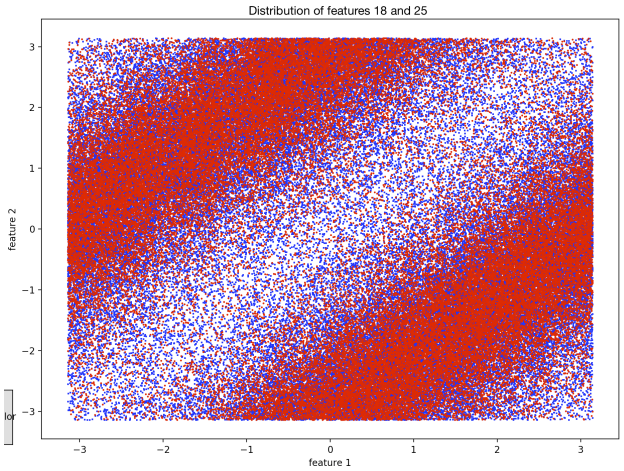

Figure 4. Repartition of features 18 ans 25

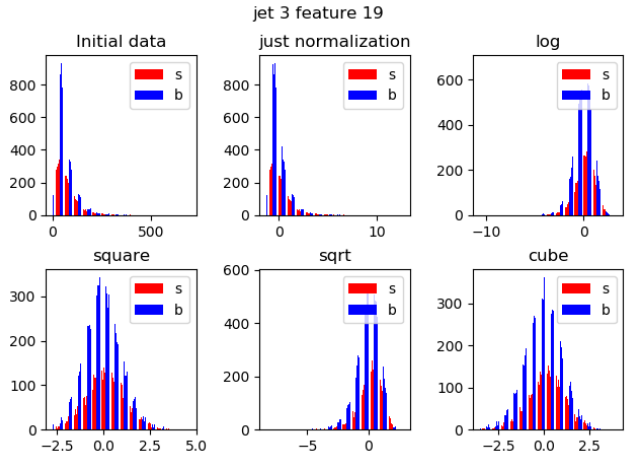

Figure 5. Function applied to feature 19

References

[1] Editorial, "Scientific writing 101," *Nature Structural & Molecular Biology*, vol. 17, p. 139, 2010.

[2] S. P. Jones, "How to write a great research paper," 2008, microsoft Research Cambridge.

[3] G. Anderson, "How to write a paper in scientific journal style and format," 2004, http://abacus.bates.edu/ ganderso/biology/resources/writing/HTWtoc.html.

[4] J. B. Buckheit and D. L. Donoho, "Wavelab and reproducible research," Stanford University, Tech. Rep., 2009.

[5] R. H. Kallet, "How to write the methods section of a research paper," *Respiratory Care*, vol. 49, no. 10, pp. 1229–1232, 2004.

[6] A. Hunt and D. Thomas, *The Pragmatic Programmer*. Addison Wesley, 1999.

[7] J. Spolsky, *Joel on Software: And on Diverse & Occasionally Related Matters That Will Prove of Interest etc..: And on Diverse and Occasionally Related Matters ... or Ill-Luck, Work with Them in Some Capacity*. APRESS, 2004.

[8] M. Schwab, M. Karrenbach, and J. Claerbout, "Making scientific computations reproducible," *Computing in Science and Engg.*, vol. 2, no. 6, pp. 61–67, 2000.