Kernel K Nearest Neighbors

Mateus Maia, Anderson Ara

mateusmaia11@gmail.com

Salvador, 13/11/2018

Conference on Statistics and Data Science

Introduction

Introduction

The nearest-neighbor (nn) algorithm is extremely simple and is open to a wide variety of variations.

The "kernel approach" offers the possibility to mapping the data into a high dimensional feature space.

Objectives

Objectives

Study how the "Kernel Trick" can influence the classification performed by KNN.

Analyze the variation in the choosing of each Kernel Function in Kernel KNN extension.

Apply to a text mining task classification the KKNN.

K Nearest Neighbors

K Nearest Neighbors

It's a non parametric technique which the decision rule is given by $g(\mathbf{x})$, where:

$$g(x) = \operatorname*{mode}_{i \in N} y_i = \operatorname*{argmax}_{c \in Y} \sum_{i \in N} \delta(c = y_i) \tag{1}$$

where N is the set of K closest observations from $\mathbf{x}$ and $\delta(c = y_i)$ is the Delta Dirac function, that assumes value one when $y_i = c$ and zero otherwise.

K Nearest Neighbors

The most common measure of distance that’s used in KNN is the Euclidian Distance given by

$$D(x_i, \mathbf{x_j}) = \sqrt{\mathbf{x_i} \bullet \mathbf{x_i} - 2\mathbf{x_i} \bullet \mathbf{x_j} + \mathbf{x_j} \bullet \mathbf{x_j}} \tag{2}$$

WKNN

Another KNN's approach it's the Weighted KNN, where the exists a factor w_i that given more importance to the similar observations. And the decision rule it's given by

$$g(x) = \underset{c \in Y}{\text{argmax}} \sum_{i \in N} w_i \delta(c = y_i) \tag{3}$$

Kernel Trick

Kernel Trick

Defining $\phi : x \rightarrow \phi(x), x \in X, \phi \in F$.

The kernel trick enable, by replacing the inner product with an appropriate kernel function, one can implicitly perform a nonlinear mapping to a high dimensional feature space.
The statement above it's can be written by the equation

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{4}$$

Kernel K Nearest Neighbors

Taking the Kernel Function, and assuming the inner product as a similarity measure, we can define the Kernel KNN decision rule as

$$g(x) = \underset{c \in Y}{\operatorname{argmax}} \sum_{i \in N} w_i \delta(c = y_i) \tag{5}$$

where, here, the weight function is given by

$$w_i = K(x, x_i)$$

Kernel Functions

There a lot of Kernel Functions, but in this work were used the following:

- Gaussian: $K(x,y) = \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2} - ||x-y||^2}$
- Exponential: $K(x,y) = \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2} - ||x-y||}$
- Epanechnikov: $K(x,y) = \frac{3}{4}(1 - ||x-y||^2)$
- Tricube: $K(x,y) = \frac{70}{81}(1 - |||x-y|||^3)^3$
- Logistic: $K(x,y) = \frac{1}{e^{||x-y||} + 2 + e^{||x-y||}}$

Applied Cases: Non-Text Data

Applied Cases: Non-Text Data

Due the complexity of some Text Mining datasets, the Kernel K Nearest Neighbors was tested first in some famous and simpler databases, like:

- Ionosphere (Binary Classification: 'Good' or 'Bad')
- Seeds (Multi-Classification: Three Classes)
- Wines (Multi-Classification: Three Classes)

For each database the metric measured for quality of prediction was the accuracy, and the validation technique was the Repeated Holdout, with 100 repetitions, and K=5.

Ionosphere

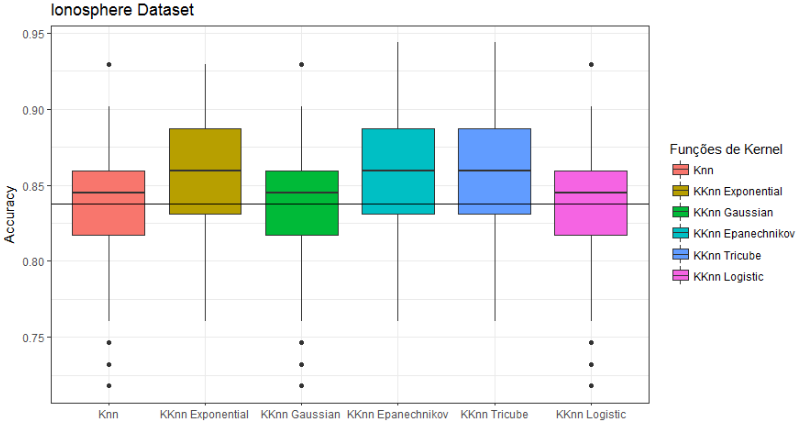

Figure: Comparison of accuracy among each KKNN algorithm .

Seeds

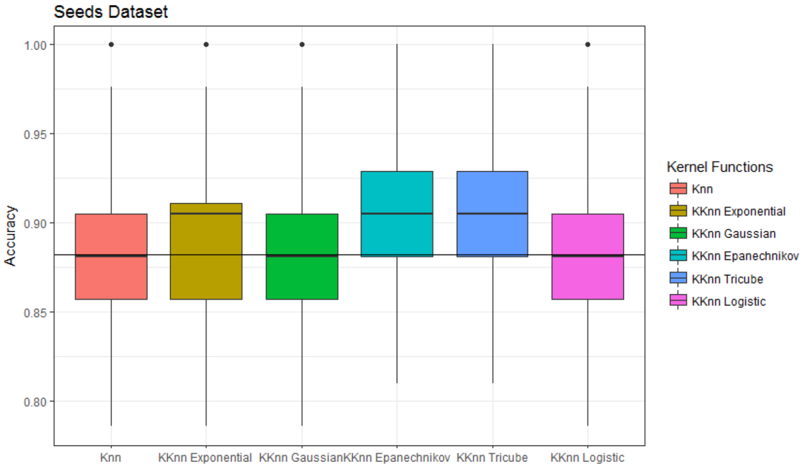

Figure: Comparison of accuracy among each KKNN algorithm .

Wines

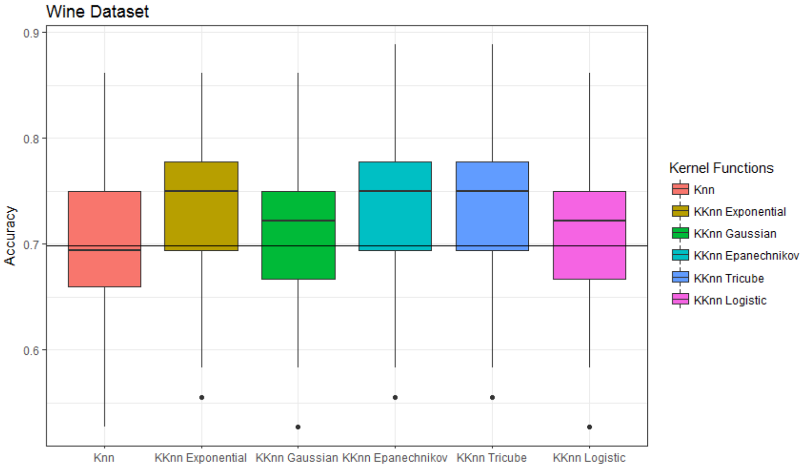

Figure: Comparison of accuracy among each KKNN algorithm .

Applied Cases: Non-Text Data

As we can observe, the KKnn's algorithms, in generall, slightly have greater accuracy when compared to the classic knn.

Most of the cases the KKnn Tricube and KKnn Epanechnikov had the best performance.

Applied Cases: Text Data

Applied Cases: Text Data

As mentioned before, in general text mining applications have a more complex data structure when compared to standard databases.

The main object to modelling in text mining area it's the matrix called **Bag of Words**, where the rows represents the documents, and each column it's a word.

Applied Cases: Text Data (Spam)

Again, to start was analyzed first the simplest case, the database of called **Spam**, where was no need of pre-processing or weighting.

The dataset consists in 4601 e-mails as spam or non-spam, where 57 variables indicating the frequencies of certaing words and characters.

Applied Cases: Text Data(Spam)

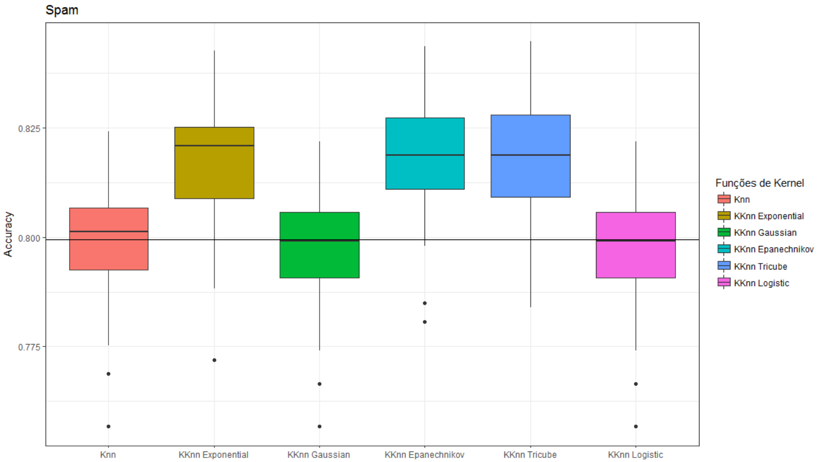

Figure: Comparison of accuracy among each KKNN algorithm.

Applied Cases: Text Data(Spam)

The value was K=3 for that model, and the validation was a repeated holdout (50 times), with the splitting ratio 80%-20%.

To the spam data, again the KKnn slightly have greater accuracy when compared to the classic knn.

Applied Cases: Text Data(BBC News)

The last database was set of 400 summaries from BBC News, where 200 are classified as Bussiness News while the other 200 are classified as Sports News.

In that case, it’s a raw dataset, so the pre-processing was necessary.

Applied Cases: Text Data(BBC News)

The pre-processing followed the steps

- **Stemming:** The reduction of the words to their root form. This can avoid plurals or verbs conjugations be considered as different expressions.
- **Stop Words**: Stop Words can be considered as the most commons words in a language, which means, that they add no value to context.
- **Formatting text**: this is general, and involves removing numbers and punctuation, or lowercase all words.

Applied Cases: Text Data(BBC News)

Weighting the terms is also important.

The parameter for weighting is given by the *tf.idf*, which the multiplication of the Term Frequency by the Inverse Document Frequency, that last is given by the equation:

$$idf(W) = log\frac{\#(documents)}{\#\text{documents containg the word}(W)}$$

After that, the main object to input the modelling it's the matrix called **Bag of Words.**

Applied Cases: Text Data(BBC)

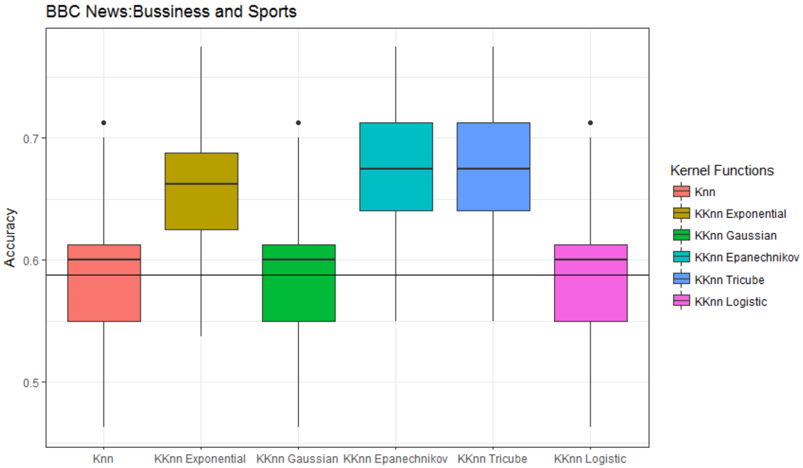

Figure: Comparison of accuracy among each KKNN algorithm.

Text Data(BBC): Tuning

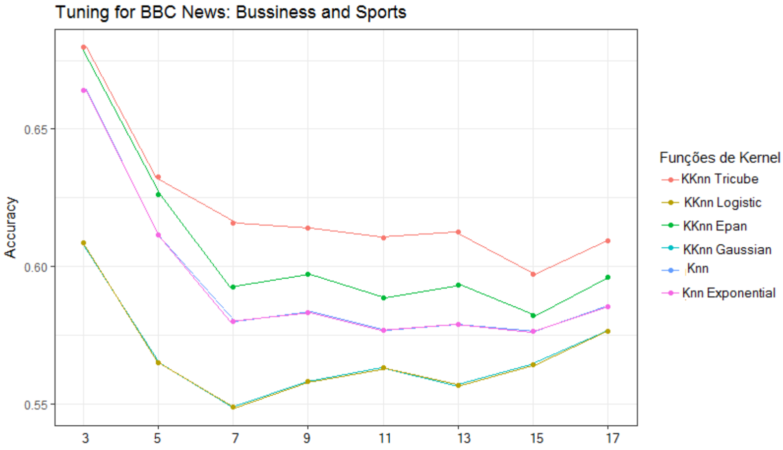

Figure: Tuning for each kernel function.

Applied Cases: Text Data(BBC)

- The validation method used was the Repeated Holdout (50 times), with the splitting ratio 80%-20%.
- Some kernels slightly increased the accuracy when compared to the classic knn
- The tuning appointed to K=3 to best hyperparameter for all techninques.

Conclusion

Conclusion

- Kernel K-Means slightly improved the classification for some kernel functions and datasets.
- It's important to measure others functions and try to vary the hyperparameters in order to find the optimum configuration for this method.
- Perhaps the slight improvement on accurracy, calculate the Kernel Matrix can be a costly operation. So, it's necessary evaluate when it's really necessary.

References

- Smola, Alex J., and Bernhard Scholkopf. Learning with kernels. Vol. 4. GMD-Forschungszentrum Informationstechnik, 1998.
- Bijalwan, V., Kumar, V., Kumari, P., Pascual, J. (2014). KNN based machine learning approach for text and document mining. International Journal of Database Theory and Application, 7(1), 61-70.

Thank you for attention!

CAPES