UNIVERSITY NAME

DOCTORAL THESIS

Dicionário de Machine Learning

Author:
Marcos Augusto BARBOSA

Supervisor:
Dr. James SMITH

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Research Group Name
Department or School Name

November 15, 2022

Contents

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

List of Symbols

a	distance	m
P	power	W (J s^{-1})
ω	angular frequency	rad

For/Dedicated to/To my…

Chapter 1

Supervised Learning

1.1 Initial Considerations

1. Regression: Linear Regression and Generalized Linear Models (GLM's);
2. Instance-based Algorithms: k-Nearest Neighbor (KNN);
3. Decision Tree Algorithms: CART (Classification and regression tree);
4. Bayesian Algorithms: Naive Bayes;
5. Ensemble Algorithms: Random Forest, AdaBoost, eXtreme Gradient Boosting;
6. Deep Learning Algorithms: Convolution Neural Network.

1.2 Linear Regression

Using the classical linear regression model is justified if we can admit:

1. Linearity of the structure of $E(Y)$;
2. Variance of the error is constant, $Var(Y) = \sigma^2$;
3. Normality of the observations y's
4. Independency of the observations y's

If the assumptions (1) to (3) are not satisfied for the original data, a non-linear transformation of Y might verify them, at least approximately. (The Box and Cox models class tries to transform the dependent variable to satisfy the assumptions (1) to (4))

The R^2 (Coefficient of Determination) represents the proportion of the total variation explained by the relation of **X** and **Y** (regression).

$$R^2 = \frac{SQReg}{SQT} = 1 - \frac{SQRes}{SQT} \tag{1.1}$$

Large values of R^2 indicates that the total variation of **Y** is reduced by the insertion of the explanatory variables $X_1, X_2, \ldots, X_p$ Adding more explanatory variables to the model will increase the R^2. So, here is the $R^2_{adjusted}$, given by

$$R^2 = 1 - \frac{SQRes/(n-(p+1))}{SQT/(n-1)} \tag{1.2}$$

$R^2_{adjusted}$ will increase when the explanatory variable addition reduces the $SQRes/(n-(p+1))$

1.3 Generalized Linear Models

Using the generalized linear model is justified if we can admit:

1. Y_i's are independent;

2. Y belongs to a exponential family $FE(\theta, \phi)$;

3. Exists a function g (doubly differentiable and invertible) that relates the μ_i to a linear predictor $\eta_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$

The linear predictor can return values from $(-\infty, \infty)$. So if the response variable is in that interval, like a normal distribution and the link function is the identity, it is great! But if the response variable is always positive and if the mean is too distant from the zero, you can actually use the classical linear regression, because even though the linear predictor is able to return negative values, the mean is too distant from the zero and there are few values close to zero, it would not probably be able to predict negative values (unless in some extreme combination of covariables values).

1.4 Mixture Models for Density Estimation and Classification

The mixture model is a useful tool for density estimation, and can be viewed as a kind of kernel method. The Gaussian mixture model has the form

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \mathbf{\Sigma}_m) \tag{1.3}$$

with mixing proportions α_m, $\Sigma_m \alpha_m = 1$, and each Gaussian density has a mean μ_m and covariance matrix $\mathbf{\Sigma}_m$. In general, mixture models can use any component densities in place of the Gaussian in (6.32): the Gaussian mixture model is by far the most popular.

The parameters are usually fit by maximum likelihood, using the EM algorithm as described in Chapter 8.

Posso utilizar esse modelo quando há apenas uma variável em disposição? Sim e para quano tiver mais de uma também

In the multivariate Gaussian mixture problem (see Exercise 12.9), the "curse of dimensionality" raises its ugly head, where the number of para- meters grows quickly with the increase in dimensionality. Although PCA is often used as a first step to reduce the dimensionality, this does not help in mixtures problems because any class structure as exists may not be pre- served by the principal components (Chang, 1983).

1.5 k-nearest neighbours (classifiers)

These classifiers are memory-based and require no model to be fit. Given a query point x_0, we find the k training points $x(r), r = 1, \ldots, k$ closest in distance to x_0, and then classify using majority vote among the k neighbors. Ties are broken at random. For simplicity we will assume that the features are real-valued, and we use Euclidean distance in feature space:

$$d(i) = ||x(i) - x_0|| \tag{1.4}$$

Typically we first standardize each of the features to have mean zero and variance 1, since it is possible that they are measured in different units.

1.6 Support Vector Machines (SVM's)

Mostly used in classification problems. We may want to enlarge our feature space in order to accommodate a non-linear boundary between the classes. Produces nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space. The kernel approach that we describe here is simply an efficient computational approach for enacting this idea. What is the advantage of using a kernel rather than simply enlarging the feature space using functions of the original features, as in (9.16)? One advantage is computational, and it amounts to the fact that using kernels, one need only compute $K(x_i, x_i^{'})$ for all (n 2) distinct pairs $i, i^{'}$. This can be done without explicitly working in the enlarged feature space.

- Advantages: efective tool in high-dimensional spaces, memory efficient (Since only a subset of the training points are used in the actual decision process of assigning new members, just these points need to be stored in memory and calculated upon when making decisions and versatility (Class separation is often highly non-linear. The ability to apply new kernels allows substantial flexibility for the decision boundaries, leading to greater classification performance)

- Disadvantages: SVMs are very sensitive to the choice of the kernel parameters and there is no direct probabilistic interpretation for group membership and the black box nature of these functions. The use of kernels to separate nonlinear data makes them difficult (if not impossible) to interpret. Despite its popularity, SVM has a serious drawback, that is sensitivity to outliers in training samples. The penalty on misclassification is defined by a convex loss called the hinge loss, and the unboundedness of the convex loss causes the sensitivity to outliers.

1.7 Random Forest

The essential idea in bagging (Section 8.7) is to average many noisy but approximately unbiased models, and hence reduce the variance. Trees are ideal candidates for bagging, since they can capture complex interaction structures in the data, and if grown sufficiently deep, have relatively low bias. Since trees are notoriously noisy, they benefit greatly from the averaging. Moreover, since each tree generated in bagging is identically distributed (i.d.), the expectation of an average of B such trees is the same as the expectation of any one of them. This means the bias of bagged trees is the same as that of the individual (bootstrap) trees, and the only hope of improvement is through variance reduction. This is in contrast to boosting, where the trees are grown in an adaptive way to remove bias, and hence are not i.d.

The idea in random forests (Algorithm 15.1) is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much. This is achieved in the tree-growing process through random selection of the input variables.

Combination of many decision trees, effectively leveraging and combining the choices of many models (this technique of using a combination of models is known as **ensembling**).

Para classificação, uma medida de impureza que pode ser usada é:

- Misclassification error
- Gini index
- Cross-entropy or deviance

Para regressão, uma medida de impureza que pode ser usada é $SSE = \sum_{i \in S_1}(y_i - \bar(y)1)^2 + \sum_{i \in S_2}(y_i - \bar(y)2)^2$.

Out-of-Bag sample, no contexto de *random forest*, consiste na observação que não foi selecionada na amostra *bootstrap* utilizada por uma determinada árvore de decisão.

Out-of-Bag error, no contexto de *random forest*, consiste na proporção de *out-of-bag samples* incorretamente classificadas.

1.8 AdaBoost

How weights are updated in AdaBoost? Simply put, the idea is to set weights to both classifiers and data points (samples) in a way that forces classifiers to concentrate on observations that are difficult to correctly classify. This process is done sequentially in that the two weights are adjusted at each step as iterations of the algorithm proceed.

AdaBoost is also extremely sensitive to Noisy data and outliers so if you do plan to use AdaBoost then it is highly recommended to eliminate them. AdaBoost has also been proven to be slower than XGBoost.

1.9 Gradient Boosting Machine (Regression)

We start with a leaf that is the average value of the variable we want to predict. Then we add a tree based on the Residuals, tree is scaled by a contrbution (fixed learning rate). Then we add another tree based on the new residuals and we keep adding trees based on the errors made by the previous tree.

1.10 XGBoost

XGBoost is an exceptionally useful machine learning method when you don't want to sacrifice the ability to correctly classify observations but you still want a model that is fairly easy to understand and interpret.

XGBoost, scalable machine learning system for tree boosting, was designed to be used with large, complicated datasets. Just like Gradient Boost (unextreme), XGBost fits a Regression Tree to the residuals but uses a unique Regression Tree. Let's callt it XGBoost tree.

We prune the XGBoost tree based on the gain of the branch and a γ

λ is a regularization parameter, which means that it is intended to reduce the prediction's sensitivity to individual observations, prevents over fitting the training data. It results in more pruning, by shrinking the Similarity Scores and it results in smaller Output Values for the leaves

There is a learning rate parameter also.

We calculate Similarity Scores and Gain to determine how to split the data and we prune the tree by calculating the differences between Gain values and a user defined Tree Complexity Parameter γ. If positive, then do not prune. If negative, then prune.

Then we calculate the Output Values for the remaining leaves.

The minimum number of Residuals in each leaf is determined by calculating something called Cover

One thing that is relatively unique about XGBoost is that it has default behavior for missing data. In Python, all we have to do is identify missing values and make sure they are set to 0

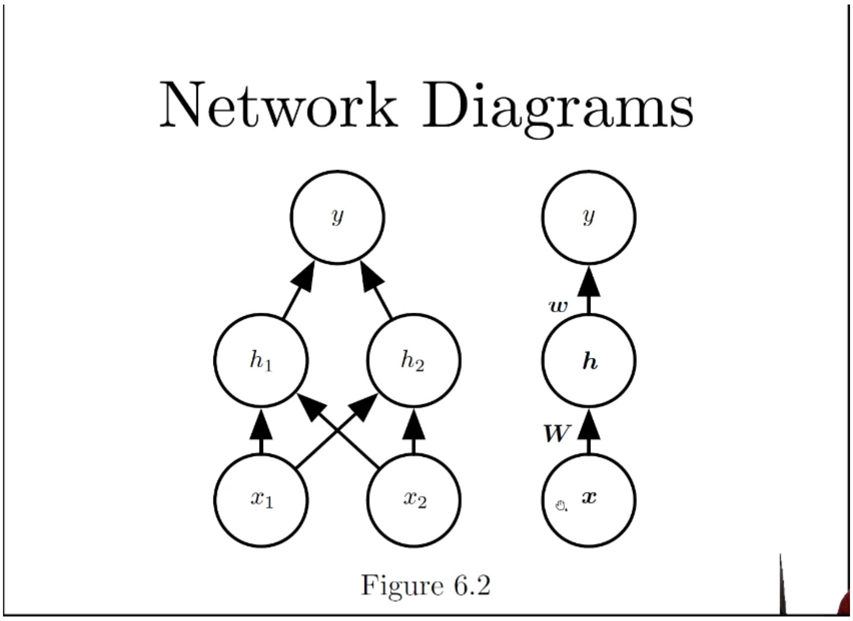
Uses the Second Order Taylor Approximation for both Regression and Classification.

For classification, the negative log-likelihood is the most commonly used loss function.

There are several different ways to calculate feature importances. By default, “gain” is used, that is the average gain of the feature when it is used in trees. Other types are “weight” - the number of times a feature is used to split the data, and “cover” - the average coverage of the feature. You can pass it with $importance_type$ argument.

1.11 Neural Networks

FIGURE 1.1: NeuralNetwork



Back-propagation is "just the chain rule" of calculus, one of the key concepts of NN. É um algortimo para calcular o gradiente de modo extramammente eficiente (baixa complexidade computacional), Is the method to calculate the gradient of the loss function with respect to the weights in an artificial neural network

1. Forward prop: start an the input x, apply the weight and biases and compute y and the cost

2. Back prop: based on the associated y and cost, update weighs and biases?

3.

When we use a feedforward neural network to accept an input x and produce an output $\hat{y}$, information flows forward through the network. The inputs x provide the initial information that then propagates up to the hidden units at each layer and finally produces $\hat{y}$. This is called forward propagation . During training, forward propagation can continue onward until it produces a scalar cost $J(\boldsymbol{\theta})$. The back-propagation algorithm, often simply called backprop, allows the information from the cost to then flow backwards through the network, in order to compute the gradient.

The term back-propagation is often misunderstood as meaning the whole learning algorithm for multi-layer neural networks. Actually, back-propagation refers only to the method for computing the gradient

Activating functions: the softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.

1.12 Conceitos importantes

1.12.1 Multicolinearidade

Multicollinearity happens when independent variables in the regression model are highly correlated to each other. It makes it hard to interpret of model and also creates an overfitting problem. It is a common assumption that people test before selecting the variables into the regression model. How to check wheter Multicollinearity occurs?

1. Plot the correlation matrix of all the independent variables

2. Use the Variance Inflation Factor (VIF) for each independent variable. It is a measure of multicollinearity in the set of multiple regression variables. The higher the value of VIF the higher correlation between this variable and the rest.

How to fix the Multi-Collinearity issue?

1. Variable selection

2. Variable transformation

3. PCA (Principal Component Analysis): the character of variable independence

1.12.2 Cross Validation

Consiste em dividir o conjunto de dados (treino + teste) em k pedaços. Em seguida, para cada combinação de $k-1$ pedaços, é ajustado o modelo de interesse e calculado, no pedaço restante, a(s) métrica(s) de validação. Útil para comparar a performance entre modelos distintos e para selecionar os melhores *tunning parameters*. Algumas variações desse procedimento:

1. **K-Fold Cross Validation**

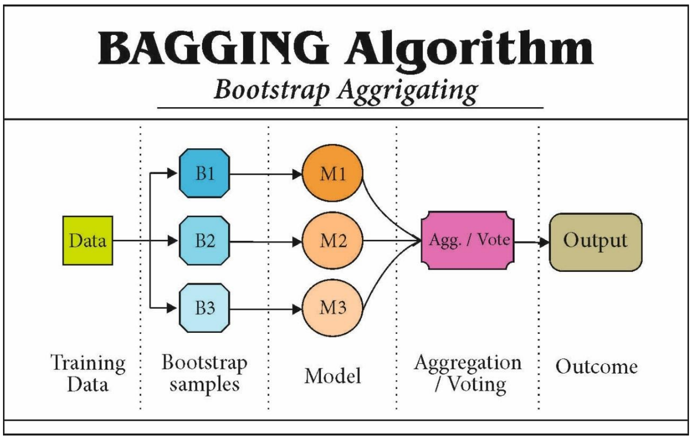
2. **Leave One Out Cross Validation**

3. **Repeated Cross Validation**

1.12.3 Bagging (Boostrap Aggregating)

Dado um conjunto de dados de treino, o procedimento *bagging* gera novas m amostras com reposição desse conjunto (esse tipo de amostra é chamada de *bootstrap*), para, em seguida, ajustar m modelos nas m amostras *bootstrap* (um modelo para cada amostra) e combinar o output em uma média (no caso de regressão) ou votação (no caso de classificação). The essential idea in bagging is to average many noisy but approximately unbiased models, and hence reduce the variance. The number of trees B is not a critical parameter with bagging; using a very large value of B will not lead to overfitting. In practice we use a value of B sufficiently large that the error has settled down.

FIGURE 1.2: Illustration of bagging



1.12.4 Boosting

A weak classifier is one whose error rate is only slightly better than random guessing. The purpose of boosting is to sequentially apply the weak classification algorithm to repeatedly modified versions of the data, thereby producing a sequence of weak classifiers $G_m(x)$, $m = 1, 2, \dots, M$.

Boosting works in a similar way of the bagging procedure except that the trees are grown sequentially: each tree is grown using information from previously grown trees. Boosting does not involve bootstrap sampling; instead each tree is fit on a modified version of the original data set. Boosting has three parameters:

1. The number of trees B. Unlike bagging and random forests, boosting can overfit if B is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select B.

2. The shrinkage parameter λ, a small positive number. This controls the rate at which boosting learns. Typical values are 0.01 or 0.001, and the right choice can depend on the problem. Very small λ can require using a very large value of B in order to achieve good performance.

3. The number d of splits in each tree, which controls the complexity of the boosted ensemble. Often d = 1 works well, in which case each tree is a stump, consisting of a single split. In this case, the boosted ensemble is fitting an additive

model, since each term involves only a single variable. More generally d is the interaction depth, and controls the interaction order of the boosted model, since d splits can involve at most d variables.

1.12.5 Regularization

Regularized regression consists in estimating a penalized function of the form.

$$\min_{f \in H} \Big[\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \Big], \tag{1.5}$$

,

where $L(y, f(x))$ is the chosen loss function, $J(f)$ is a penalty functional and H is a space of function on which $J(f)$ is defined (Hastie, Tibshirani, and Friedman, 2009)

Produces models that are more parsimonious and have similar prediction error as the full model and it is usually robust enough to not be influenced by the correlated variables.

For tree-based methods, there is not yet a well established regularization procedure in the literature.

1. **Ridge Regression (L2)**: consiste em adicionar uma penalidade equivalente aos quadrados das magnitudes dos coeficientes, na função de custo. É útil para fazer *shrink* das estimativas dos parâmetros (pode chegar perto de zero, mas não exatamente 0) e reduzir a complexidade do modelo e a multicolinearidade. Conforme *Introduction to Statistical Learning*, tem-se que *it is best to apply ridge regression after standardizing the predictors*.

   When the number of variables p is almost as large as the number of observtions n, the OLS estimates will be extremely variable. And if $p > n$, then the OLS estimates do not even have a unique solution, whereas ridge regression can still perform well by trading off a small increse in bias for a large decrease in variance (bias and variance tradeoff)

2. **Lasso Regression (L1)**: consiste em adicionar uma penalidade equivalente às magnitudes dos coeficientes, na função de custo. É útil para reduzir *overfitting* e selecão de variáveis (os coeficientes podem chegar a zero). Sendo assim, resulta em uma equação mais imples e mais fácil de interpretar.

3. **Regularization Elastic Net**: consiste em uma combinação entre *Ridge Regression (L1)* e *Lasso Regression (L2)*. Pode ajudar a excluir certos parâmetros.

1.12.6 Métricas de avaliação

Regressão

1. Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{1.6}$$

   Like variance, mean squared error has the disadvantage of heavily weighting outliers.[11] This is a result of the squaring of each term, which effectively weights large errors more heavily than small ones. This property, undesirable in many applications, has led researchers to use alternatives such as the mean absolute error, or those based on the median.

2. Mean Absolute Error (MAE)
3. Mean Absolute Percentage Error (MAPE)

Classificação

1. *Accuracy*: (TN + TP) / (TN + FP + FN + TP)
2. *Precision*: TP / (TP + FP)
3. *Recall, Sensitivity, True Positive Rate*: TP / (TP + FN)
4. $f_1 score = 2\frac{precision*recall}{precision+recall}$
5. Curva ROC: gráfico entre *True Positive Rate* e *False Positive Rate* (FP / (FP + TN)). It summarizes all of the confusion matrices that each threshold produced

A utilização da acurácia não é recomendado no caso de um conjunto de dados desbalanceados.

Chapter 2

Unsupervised Learning

2.1 Partitioning Methods

Suitable for finding spherical-shaped clusters or convex clusters. In other words, they work well for compact and well separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data. Unfortunately, real life data can contain: i) clusters of arbitrary shape (oval, linear and "S" shape clusters); ii) many outliers and noise.

2.1.1 K-means

There are various methods to find the optimal/best value of k. In this article we will cover two:

- Elbow Method
- Silhouette Method: The silhouette Method is also a method to find the optimal number of clusters and interpretation and validation of consistency within clusters of data. The silhouette method computes silhouette coefficients of each point that measure how much a point is similar to its own cluster compared to other clusters by providing a succinct graphical representation of how well each object has been classified. Compute silhouette coefficients for each of point, and average it out for all the samples to get the silhouette score

  The value of the silhouette ranges between [1, -1], where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

Importante pensar cuidadosamente se é necessário padronizar as variáveis (deixá-las com desvio-padrão igual 1).

2.1.2 PAM clustering

2.1.3 Clusterização Hierárquica

É importante pensar cuidadosamente se é necessário padronizar as variáveis (deixá-las com desvio-padrão igual 1). alguns métodos de aglomeração:

- *Single-Linkage*: mínima dissimilaridade intercluster. Consiste em em computar todas as *pairwise* dissimilaridades entre as observações do cluster A e as observações do cluster B e registrar a menor dessas dissimilaridades.
- *Complete-Linkage*: máxima dissimilaridade intercluster. Consiste em em computar todas as *pairwise* dissimilaridades entre as observações do cluster A e as

observações do cluster B e registrar a maior dessas dissimilaridades. Tende a gerar cluster's mais balanceados.

- *Average-Linkage*: média dissimilaridade intercluster. Consiste em em computar todas as *pairwise* dissimilaridades entre as observações do cluster A e as observações do cluster B e registrar a média dessas dissimilaridades. Tende a gerar cluster's mais balanceados.

- *Centroid-Linkage*: dissimilaridade entre centróide do cluster A e centróide do cluster B. Pode resultar em indesejáveis inversões.

2.2 Density methods

2.2.1 DBScan (Density-Based Spatial Clustering and Application with Noise)

Non-linear algorithm and it is insensitive to order. Unlike to K-means, DBSCAN does not require the user to specify the number of clusters to be generated, DBSCAN can find any shape of clusters. The cluster doesn't have to be circular. DBSCAN can identify outliers.

2.3 Considerações

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.

- In the case of hierarchical clustering,

    1. What dissimilarity measure should be used? Euclidiana, alguma distância baseada em correlação?
    2. What type of linkage should be used?
    3. Where should we cut the dendrogram in order to obtain clusters?

- In the case of K-means clustering, how many clusters should we look for in the data?

Chapter 3

Implementação de modelo ML

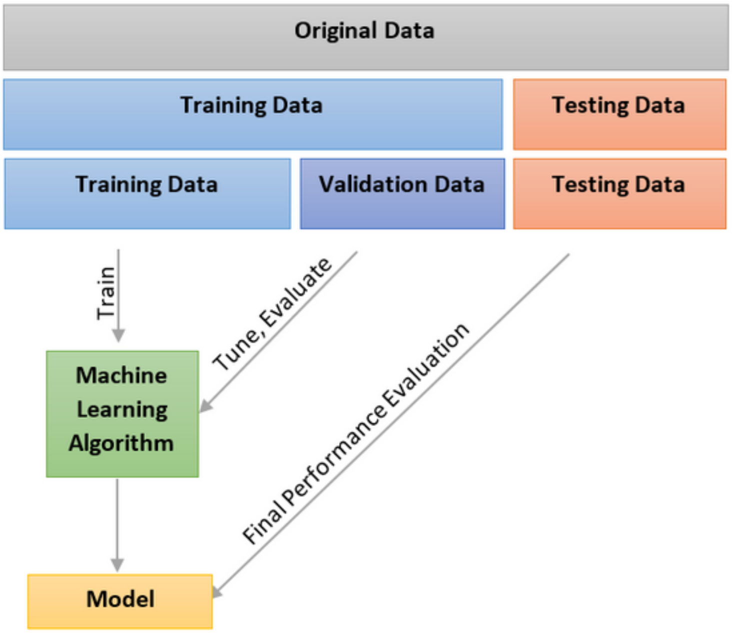
3.1 Considerações Iniciais

Conforme Vincent, o foco deve estar em entender com profundidade o problema, dar um passo atrás e ser humilde, se para você, o problema ainda não está claro. Não se deve focar, em primeiro lugar na solução, e pensar o problema como algo meramente analítico ("Vou ajustar uma rede neural e encontrar os melhores hiperparâmetros e o problema será resolvido!). De modo geral, o que mas importa é todo o conjunto de componentes ao redor do modelo de ML, não o modelo em si. As componentes seriam:

- Quais dados temos acesso?
- Qual é a utilidade do modelo para o usuário final?
- Como seria feito o monitoramento do modelo? Quão complicado seria corrigir erros podem surgir em produção? Corrigir erros de uma regressão linear é muito mais fácil e rápido que corrigir os erros de uma rede neural com milhares/milhões de parâmetros
- Um sistema simples de regras poderia desempenhar tão bem quanto um modelo de ML?
- Como avaliar a qualidade do modelo proposto? Há algum benchmark? Pode-se criar um benchmark?
- Fazer um desenho à mão de toda a arquitetura é uma boa ideia

3.2 Particionamento dos Dados

FIGURE 3.1: Data Partitioning



- We usually devide the data to train and test set. We will not touch test set until the end of the computation and the final perpormance evaluation. Then, we can devide the train set to train and validation sets. We use the validation data set to tune the model.
- Traditional train test method suffer from high variance test problem. It means tha by changing the test set the result of the prediction changes. To over come this problem we use k-fold validation method in our train and validation set
- Separating samples into training and test sets is not always enough. There can be various reasons for duplicate samples that appear in both sets, and it is important to detect and remove these duplicates. For example, one of the most used datasets in Computer Vision, CIFAR, is shown to contain duplicate samples in training and test sets. This becomes more of a problem for models that are trained on huge amounts of data such as BERT and GPT-3. Best practices: Remove duplicates before splitting the data, check for partial duplicates as well, sort by different columns, and examine the data.

3.3 Passo a Passo

1. Estudo profundo das fontes de informação disponíveis (banco de dados, API's, planilhas);
2. Identificação de variáveis relevantes. Essa etapa costuma assumir um dos dois caminhos seguintes:

(a) Um seleção de variáveis é feita, com base no conhecimento do negócio envolvido. Se o cientista de dados não conhece bem o campo de conhecimento envolvido, fica difícil fazer isso. Então, consultar especialistas é extremamente recomendado

(b) Milhões, milhares de variáveis são escolhidas e jogadas para treinar o modelo mais para frente.

3. Fazer Exploratory Data Analysis do conjunto de dados:

   - Há dados faltantes?
   - Há dados que não fazem sentido algum? Por exemplo, uma pessoa tem altura negativa!
   - As strings, para uma determinada coluna, são consistentes? Por exemplo, a string que se refere a cidade "New York" possui diferentes formatos, e.g., "NEWYORK", "ny", "NY", "new york".

3.4 Considerações sobre predição

Um modelo de aprendizado de máquina supervisionado treinado com um conjunto de dados de treino é capaz de fazer predições e, por meio dele, conseguimos testar sua performance no conjunto de dados de teste.

Contudo, é de extrema importância que um modelo de ML, em produção, em uma empresa ou aplicação na vida real passe por uma análise de sua performance com um conjunto de dados que só será obtido no futuro.

Um modelo que classifica se uma reserva de hotel será cancelada, com o snapshot dos dados atuais "apenas" classifica as observações em não canceladas ou canceladas. Entretanto, a gerência do hotel (usuário do modelo) está interessada especificamente em prever, no futuro, um cancelamento. Classificar uma observação atual em não cancelada ou cancelada é algo distinto de prever, no futuro, um cancelamento de reserva de hotel.

Sendo assim, é necessário que de tempos em tempos, com novas safras de dados e as previsões feitas antecipadamente sejam estudadas. É necessário simular o uso final do modelo!

Chapter 4

Outros tópicos de ML

4.1 "Off-the-Shelf" Procedures for Data Mining

Predictive learning is an important aspect of data mining. As can be seen from this book, a wide variety of methods have been developed for predic- tive learning from data. For each particular method there are situations for which it is particularly well suited, and others where it performs badly compared to the best that can be done with that data. We have attempted to characterize appropriate situations in our discussions of each of the re- spective methods. However, it is seldom known in advance which procedure will perform best or even well for any given problem. Table 10.1 summarizes some of the characteristics of a number of learning methods. Industrial and commercial data mining applications tend to be especially challenging in terms of the requirements placed on learning procedures. Data sets are often very large in terms of number of observations and number of variables measured on each of them. Thus, computational considerations play an important role. Also, the data are usually messy: the inputs tend to be mixtures of quantitative, binary, and categorical vari- ables, the latter often with many levels. There are generally many missing values, complete observations being rare. Distributions of numeric predictor and response variables are often long-tailed and highly skewed. This is the case for the spam data (Section 9.1.2); when fitting a generalized additive model, we first log-transformed each of the predictors in order to get a reasonable fit. In addition they usually contain a substantial fraction of gross mis-measurements (outliers). The predictor variables are generally measured on very different scales. In data mining applications, usually only a small fraction of the large number of predictor variables that have been included in the analysis are actually relevant to prediction. Also, unlike many applications such as pat- tern recognition, there is seldom reliable domain knowledge to help create especially relevant features and/or filter out the irrelevant ones, the inclu- sion of which dramatically degrades the performance of many methods. In addition, data mining applications generally require interpretable mod- els. It is not enough to simply produce predictions. It is also desirable to have information providing qualitative understanding of the relationship between joint values of the input variables and the resulting predicted re- sponse value. Thus, black box methods such as neural networks, which can be quite useful in purely predictive settings such as pattern recognition, are far less useful for data mining. These requirements of speed, interpretability and the messy nature of the data sharply limit the usefulness of most learning procedures as off- the-shelf methods for data mining. An "off-the-shelf" method is one that can be directly applied to the data without requiring a great deal of time- consuming data preprocessing or careful tuning of the learning procedure. Of all the well-known learning methods, decision trees come closest to meeting the requirements for serving as an off-the-shelf procedure for data mining. They are relatively fast to construct and they produce interpretable models (if the trees are small). As discussed in Section 9.2, they naturally incorporate mixtures of numeric and categorical predictor variables and missing values. They are invariant under (strictly monotone) transforma- tions of the individual predictors. As a result, scaling and/or more general transformations are not an issue, and they are immune to the effects of pre- dictor outliers. They perform internal feature selection as an integral part of the procedure. They are thereby resistant, if not completely immune, to the inclusion of many irrelevant predictor variables. These properties of decision trees are largely the reason that they have emerged as the most popular learning method for data mining. Trees have one aspect that prevents them from being the ideal tool for predictive learning, namely inaccuracy. They seldom provide predictive ac- curacy comparable to the best that can be achieved with the data at hand. As seen in Section 10.1, boosting decision trees improves their accuracy, often dramatically. At the same time it maintains most of their desirable properties for data mining. Some advantages of trees that are sacrificed by boosting are speed, interpretability, and, for AdaBoost, robustness against overlapping class distributions and especially mislabeling of the training data. A gradient boosted model (GBM) is a generalization of tree boosting that attempts to mitigate these problems, so as to produce an accurate and effective off-the-shelf procedure for data mining.

FIGURE 4.1: LearningMethods

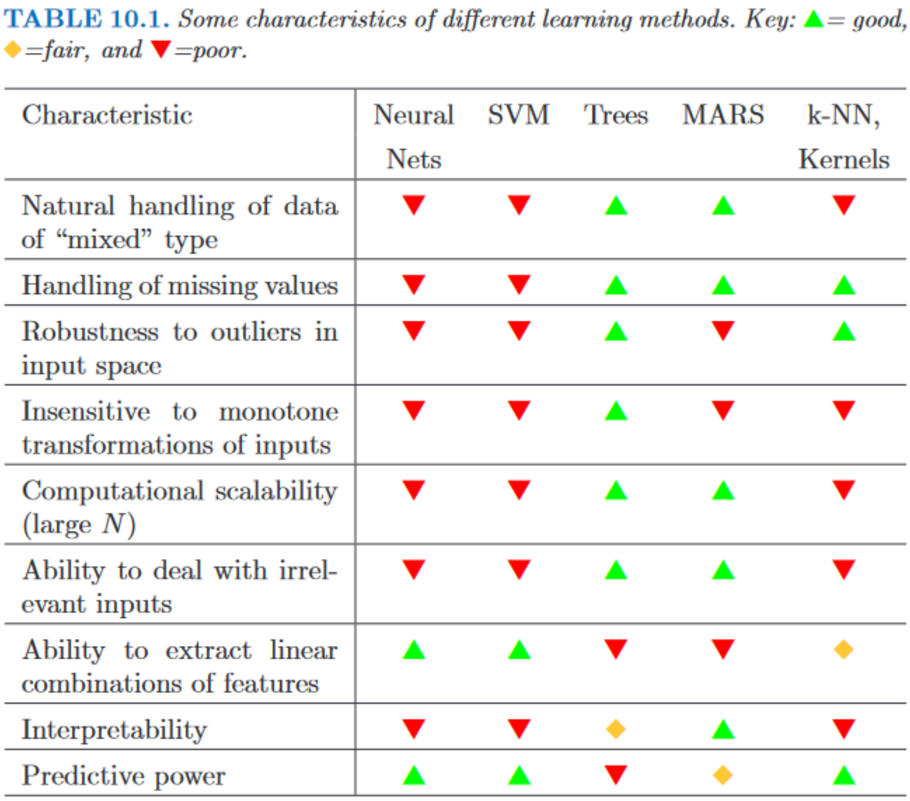


TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲*= good,* ◆*=fair, and* ▼*=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

4.2 Principal Components Analysis

The basic goal of principal components analysis is to describe variation in a set of correlated variables, $\mathbf{x}^T = (x_1, \ldots, x_q)$ in terms of new set of uncorrelated variables $\mathbf{y}^T = (y_1, \ldots, y_q)$, each of which is a linear combination of the $\mathbf{x}$ variables.

The new variables are derived in decreasing order of "impor- tance" in the sense that y_1 accounts for as much as possible of the variation in the original data amongst all linear combinations of $\mathbf{x}$. Then y_2 is chosen to account for as much as possible of the remaining variation, subject to being uncorrelated with y_1, and so on. The new variables defined by this process, $y_1, \ldots, y_q$, are the principal components.

The first principal component of the observations, y1, is the linear combination

$$y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1q}x_q \tag{4.1}$$

whose sample variance is greatest among all such linear combinations. Because the variance of y_1 could be increased without limit simply by increasing the coefficients $a_1^T = (a_{11}, a_{12}, \ldots, a_{1q}x_q)$, a restriction must be placed on these coefficients. To find the coefficients defining the first principal component, we need to choose the elements of the vector $\mathbf{a}_1$ so as to maximize the variance of y_1 subject to the sum of squares constraint, which can be written $\mathbf{a}_1^T\mathbf{a}_1 = 1$. The sample variance of y_1 that is a linear function of the x variables is given by $\mathbf{a}_1^T\mathbf{S}\mathbf{a}_1 = 1$, where S is the qxq sample covariance matrix of the x variables. Lagrange multiplier approach leads to the solution that $\mathbf{a}_1$ is the eigenvector of the sample covariance matrix, $\mathbf{S}$, corresponding to this matrix's largest eigenvalue. The eigenvalues λ and eigenvectors γ of a qxq matrix $\mathbf{A}$ are such that $\mathbf{A}\gamma = \lambda\gamma$.

The second principal component, y2, is defined to be the linear combination

$$y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2q}x_q \tag{4.2}$$

(i.e, $y_2 = \mathbf{a}_2^T\mathbf{x}$, $a_2^T = (a_{21}, a_{22}, \ldots, a_{2q}x_q)$ and $\mathbf{x}^T = (x_1, \ldots, x_q)$) that has the greatest variance subject to the following conditions:

$$\mathbf{a}_2^T\mathbf{a}_2 = 1(normalized) \tag{4.3}$$

$$\mathbf{a}_2^T\mathbf{a}_1 = 0(orthogonal) \tag{4.4}$$

(The second condition ensures that y_1 and y_2 are uncorrelated.

Application of the Lagrange multiplier technique demonstrates that the vector of coefficients defining the jth principal component, $\mathbf{a}_j$, is the eigenvector of $\mathbf{S}$ associated with its jth largest eigenvalue.

4.3 Missing data

Missing data can be MAR, MCAR, and MNAR.

- MCAR (Missing completely at random): The values in the missing column are randomly missing and do not depend on the other column values.
- MAR (Missing at random): The values in the missing column are dependent on some additional features.
- MNAR (Missing not at random): The data is not missing randomly there might be some reason behind that.

  Lembrar do exemplo do Vincent Warmerdam: uma pessoa registra as alturas das pessoas que aparecem com a cabela acima do muro de sua casa. Os dados serão viesados pois as pessoas com altura abaixo do muro não serão utlizadas!!!

4.4 Data imputation

- Imputation using (mean/median) values
- Imputation using (most frequent) or (Zero/Constant) values
- Imputation using k-NN
- Imputation using Multivariate Imputation by Chained Equation (MICE)
- Stochastic regression imputation: similar to regression imputation which tries to predict the missing values by regressing it from other related variables in the same dataset plus some random residual value

4.5 Como lidar com evento raro?

- **Oversampling**: simple to implement and fast to execute, which is desirable for very large and complex datasets
- **Undersampling:** simple to implement and fast to execute, which is desirable for very large and complex datasets
- **SMOTE (Synthetic Minority Oversampling Technique)**: Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space. It is vital that you do not use SMOTE on the full data set. You MUST use SMOTE on the training set only (after you split). Then validate on your val/test sets and see if your SMOTE model out performed your other model(s). If you do not do this there will be data leakage and your model is essentially cheating.

4.6 Feature Selection

4.6.1 Permutation Importance

No contexto de redes neurais ou black-box models, como funciona o algoritmo:

1. Ajusta-se o modelo e calcula-se a métrica de avaliação.
2. Para cada covariável, será feito uma permutação na ordem da covariável.
3. Então calcula-se novamente a métrica de avaliação para o modelo com a covariável permutada.
4. Então, compara-se a métrica original com a métrica da variável permutada. Essa variação da métrica original será considerada como a importância da covariável permutada. Obs.: Pode-se repetir o passo 4 várias vezes, tomando a média do processo como a importância da variável.

4.6.2 Computing the amount of Impurity

Computing the amount of Impurity (typically variance in case of regression trees and gini coefficient or entropy in case of classification trees) each feature removes when it is used in node.

4.6.3 Boruta

No contexto de random forests for regression, Boruta is a feature selection algorithm which is statistically grounded and works extremely well even without any specific input by the user.

1. In practice, starting from X, another dataframe is created by randomly shuffling each feature. These permuted features are called **shadow features**. At this point, the shadow dataframe is attached to the original dataframe to obtain a new dataframe (we will call it X_boruta), which has twice the number of columns of X.

   Now, we take the importance of each original features and compare it with a threshold. This time, the threshold is defined as the highest feature importance recorded among the shadow features. When the importance of a feature is higher than this threshold, this is called a "hit". The idea is that a feature is useful only if it is capable of doing better than the best randomized feature.

2. The maximum level of uncertainty about the feature is expressed by a probability of 50%, like tossing a coin. Since each independent experiment can give a binary outcome (hit or no hit), a series of n trials follows a binomial distribution.

4.7 Tipos de Amostragem

4.7.1 Amostragem sistemática

Utilizada quando os elementos estão dispostos de maneira organizada (ex.: fila, lista) e aleatória. Escolhe um ponto de partida e seleciona-se cada k -ésimo elemento da população (ex.: o 50○elemento). Por exemplo, Em uma fábrica de lâmpadas, a cada 100 peças produzidas, uma é retirada para teste.

4.7.2 Amostragem Estratificada

Indicada quando a população está dividida em grupos distintos, denominados estratos. Dentro de cada estrato é realizada uma amostragem aleatória simples. O tamanho da amostra pode ou não ser proporcional ao tamanho do estrato. Por exemplo, uma comunidade universitária com 8000 indivíduos está estratificada da seguinte forma Estrato = [Professores, Funcionários, Estudantes], População = [800, 1200, 6000] e Amostra = [80, 120, 600]

4.7.3 Amostragem por Conglomerado

A área da população é dividida em seções (ou conglomerados, ex.: bairros, quarteirões). Os conglomerados são selecionados aleatoriamente. Dentro de um conglomerado, todos os elementos são amostrados.

4.7.4 Calibration

Adjusting the predictions of a model so that they are probabilistically meaningful.

Chapter 5

Python versus R

5.1 Observações

Python, currently, does not have libraries that perform the forward, backward and stepwise feature selection based on statistical methods (p-values). R does have pretty good libraries to work with that.

Chapter 6

Data Engineering

6.1 Principles of Relational Database Design

A relational database should satisfy the *normal forms*. Data normalization simplifies the structure of complex databases that avoids redundancy and facilitates searching data. The 3 normal forms are:

- **First Normal Form (1FN):** each field (column) accepts only one value, repetition of values is not allowed. If there is a necessity to repeat a value (for example, a client might have multiple telephone numbers), it is mandatory to create another table that can be related to the previous;
- **Second Normal Form (2FN):** the table should satisfy the 1FN and, furthermore, the fields that are not key should depend only to the key. For example, on client's registration, the field "last purchase date" does not only depend from the primary key (client identificator). This information should be obtained from another table (purchase history, for example);
- **Third Normal Form (3FN):** the table should satisfy the 2FN and, furthermore, the fields should not depend from other fields. For example, if the fields weight and height are present on the table, the field BMI (that is function of weight and height) should be removed.

6.2 API (Application Programming Interface)

- É um serviço web que provê uma interface para aplicações para manipular e extrair dados;
- It is necessary to have an API key to be able to do requests to get data;
- ENDPOINT: point of entry in a communication channel when two systems are interacting. It refers to touchpoints of the communication between an API and a server;
- Em suma, CRUD é um conjunto de operações primitivas (principalmente para bancos de dados e armazenamento de dados estáticos), enquanto o REST é um estilo de API de nível muito alto (principalmente para serviços da Web e outros sistemas "ativos");
- Some methods:
  1. GET: comando que simplesmente puxa os dados presentes em um endpoint;

2. POST: comando que possibilita uma seleção mais minuciosa dos dados. Por exemplo, filtrar os dados em uma determinada janela de tempo ou que envia dados (enviar novo login de usuário);
3. PUT: update something, modifying information;
4. DELETE:...

6.3 Amazon Web Services

6.3.1 AWS S3

Serviço de armazenamento de dados. Há funcionalidade de versionamento também;

6.3.2 AWS Lambda

- Service to run serverless code (Python, Node.js, C#, Go, etc);
- Amazon Lambda enables functions that can run up to 15 minutes;
- Each AWS Lambda execution environment provides 512 MB of disk space in the /tmp directory which can be used for some data processing and can be used for temporary storage. This /tmp disk space is preserved for the lifetime of the execution environment and provides a transient cache for data between invocations;
- Para utilizar bibliotecas, e.g requests, BeautifulSoup, no AWS Lambda, é necessário adicionar uma Layer com essa biblioteca antes. Essa Layer pode ser criada de algumas maneiras. O jeito mais fácil e é o que eu faço é o seguinte:
  1. Instalar a biblioteca com pip install <nome da biblioteca> dentro de uma pasta chamada python/ na máquina local;
  2. Transformar a pasta que contem a subpasta python/ para .zip;
  3. Fazer o upload do arquivo .zip diretamente no serviço da AWS Lambda. Ficar atento na hora de escolher a versão de runtime do python asssociada a layer;

6.3.3 AWS Step Functions

AWS Step Functions is a serverless function orchestrator that makes it easy to sequence AWS Lambda functions and multiple AWS services into business-critical applications. Through its visual interface, you can create and run a series of checkpointed and event-driven workflows that maintain the application state;

6.3.4 AWS Glue

- Sobre a ferramenta **Crawler**: Um crawler se conecta a um datastore, passa por uma lista prioritária de classificadores para determinar o esquema dos seus dados e, em seguida, cria tabelas de metadados em seu catálogo de dados (AwsDataCatalog);
- Após atualização dos dados em uma pasta que não afete a estrutura (adição de colunas, renomeação de colunas, mudança do tipo da coluna), não é necessário rodar o crawler. A tabela já estará atualizada automaticamente no AWS Athena;

- Após adição de partição no S3, é necessário rodar o AWS Glue Crawler novamente para aparecer os dados no AWS Athena

6.3.5 AWS Athena

- Serviço para fazer queries, criar views em linguagem SQL;
- Pode ser integrado com o AWS Quicksight (serviço de visualização de dados);

6.3.6 AWS CloudWatch

- É possível criar um alarme para erro de rodagem de alguma função AWS Lambda.
  - Criar novo alarme;
  - Selecionar a métrica de "Errors" vinculada a função AWS Lambda desejada (Metrics > AWS namespaces > Lambda > Por nome da função > Função desejada);
  - Selecionar/Criar notificação no AWS SNS, enviar e-mail/SMS, avisando que houve erro na rodagem.

6.3.7 AWS Quicksight

- Para views criadas no AWS Athena e adicionadas em Datasets, a atualização é imediata, mesmo sem utilizar SPICE;
- Após atualização dos dados em uma pasta que não afete a estrutura (adição de colunas, renomeação de colunas, mudança do tipo da coluna), não é necessário rodar o crawler. O dataset já estará atualizado automaticamente.

6.3.8 AWS Simple Email Service

Amazon Simple Email Service (SES) is a cloud-based email service that provides cost-effective, flexible and scalable way for businesses of all sizes to keep in contact with their customers through email.

6.3.9 AWS DynamoDB

- Serviço de banco de dados NoSQL;
- Percebi que a ordem das linhas e das colunas não é constante. Então, é recomendável, criar uma partition_key;

Chapter 7

Software Development

7.1 Docker

- Container: a way to package application with all the necessary dependencies and configuration. It's a portable artifact, easily shared and moved around.
- Some considerations:
  - A container lives on a container repository
  - Docker Hub: public repository for Docker
  - A container consists of layers of images. Mostly Linux Based Image, because small size
  - A container has a port which makes it possible to talk to the application
  - What are some simple and inexpensive ways to deploy the app on the cloud? I like Digital Ocean for simple things, probably AWS for real-world apps or you could use also Hetzner or Heroku.
- Docker Image: it is the actual package that is not running currently. But when you pull the image and start the application inside your machine, it becomes a Docker Container (running now). Basically, a Docker Container is a running environment for a Docker Image
- A Dockerfile is a simple text file that contains the commands a user could call to assemble an image whereas Docker Compose is a tool for defining and running multi-container Docker applications. Docker Compose define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment. It gets an app running in one command by just running docker-compose up. Docker compose uses the Dockerfile if you add the build command to your project's docker-compose.yml. Your Docker workflow should be to build a suitable Dockerfile for each image you wish to create, then use compose to assemble the images using the build command.

7.2 Git

- So, git fetch origin fetches any new work that has been pushed to that server since you cloned (or last fetched from) it. It's important to note that the git fetch command only downloads the data to your local repository — it doesn't automatically merge it with any of your work or modify what you're currently working on. You have to merge it manually into your work when you're ready;
- HEAD: reference to the last commit in the currently checked-out branch;

- origin: shorthand name for the remote repository that a project was originally cloned from;

7.3 Python

- É possível fazer uma instalação minuciosa do seguinte modo:

  1. The __init__.py make directories appear as libraries. That way you can import then and all the things inside them with simpler commands;
  2. É possível utilizar __all__ dentro de __init__.py, para controlar quais files são importados quando o pacote é importado, e.g. __all__ = ["file1", "file2", "file3"]
  3. Classes allows us to logically group our data and functions in a way that's easy to reuse and also easy to build upon if need be
  4. Criar um arquivo requirements.txt, que especifica as bibliotecas e suas versões;
  5. Rodar, no diretório do arquivo requirements.txt, o seguinte comando: pip install -r requirements.txt -t .
  6. Python Coding Style Conventions (About Blank Lines)
     - Leave 2 blank lines between class definitions and module-level functions
     - Leave 1 blank line between methods in a class
     - Use blank lines as needed in functions, methods, and modules to visually split up logical blocks of code

- Sobre ambientes virtuais:
  - Basicamente, uma pasta isolado no seu computador que está isolada/blindada do resto do computador. Nela, as bibliotecas são instaladas com versões específicas

7.4 AWS Configuration

- Instalar AWS CLI. Em seguida, no prompt de comando, digitar **aws configure**. O arquivo de configuração e o arquivo de credenciais serão criados.

- Instalar AWS Toolkit (VS Code extension). Automaticamente, o perfil será conectado

Chapter 8

Statistical Theory

8.1 Theorems

- **Bayes Theorem:** First, let's define a partition: The events $C_1, C_2, \dots, C_k$ form a partition of the sample space Ω, if $\Omega = \bigcup_{i=1}^{k} C_i$ and $C_i \bigcap C_j =, \forall_{i \neq j}$

  Second, let's define the Law of Total Probability: If $C_1, C_2, \dots, C_k$ form a partition of the sample space Ω and $A \in$, then:

  $$P(A) = \sum_{i=1}^{k} P(A \cap C_i) = \sum_{i=1}^{k} P(A|C_i)P(C_i) \tag{8.1}$$

  Finally, the Bayes Theorem: If $C_1, C_2, \dots, C_k$ form a partition of the sample space Ω and $A \in$, then:

  $$P(C_j|A) = \frac{P(A|C_j)P(C_j)}{\sum_{i=1}^{k} P(A|C_i)P(C_i)}, j = 1, 2, \dots, k \tag{8.2}$$

- **Strong Law of the Large Numbers:** Let $X_1, X_2, \dots, X_n$ be a sequence of i.i.d random variables with expected value $EX_i = \mu$. Then, with probability 1,

  $$\frac{X_1 + X_2 + \cdots + X_n}{n} \to \mu \text{ when } n \to \infty \tag{8.3}$$

- **The Central Limit Theorem:**

  Let $X_1, X_2, \dots, X_n$ be a sequence of i.i.d random variables with expected value $EX_i = \mu < \infty$ and variance $0 < Var(X_i) = \sigma^2 < \infty$. Then, the random variable

  $$Z_n = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}} = \frac{X_1 + X_2 + \cdots + X_n - n\mu}{\sqrt{n}\sigma} \tag{8.4}$$

  converges in distribution to the standard normal random variable as n goes to infinity, that is

  $$\lim_{n \to +\infty} P(Z_n \leq x) = \Phi(x), forall x \in \mathbb{R} \tag{8.5}$$

  where $\Phi(x)$ is the standard normal CDF.

8.2 Estimation

- **Maximum likelihood estimation:** in statistics, maximum likelihood estimation (MLE) is a method of estimating the parameters of an assumed probability distribution, given some observed data. This is achieved by maximizing a likelihood function so that, under the assumed statistical model, the observed data is most probable. The point in the parameter space that maximizes the likelihood function is called the maximum likelihood estimate.The logic of maximum likelihood is both intuitive and flexible, and as such the method has become a dominant means of statistical inference.

  The method of maximum likelihood is, by far, the most popular technique for deriving estimators. Recall that if $X_1, X_2, \dots, X_n$ are an i.i.d sample from a population with pdf or pmf $f(x|\theta_1, \dots, \theta_k)$, the likelihood function is defined by

  $$L(\theta|x) = L(\theta_1, \dots, \theta_k|x_1, \dots, x_k) = \prod_{i=1}^{n} f(x_i|\theta_1, \dots, \theta_k) \tag{8.6}$$

  Maximum-likelihood estimators have no optimum properties for finite samples, in the sense that (when evaluated on finite samples) other estimators may have greater concentration around the true parameter-value.However, like other estimation methods, maximum likelihood estimation possesses a number of attractive limiting properties: As the sample size increases to infinity, sequences of maximum likelihood estimators have these properties:

  1. Consistency: the sequence of MLEs converges in probability to the value being estimated ($\hat{\theta}_n \xrightarrow{p} \theta$), i.e,

     $$P(|\hat{\theta}_n - \theta| > \epsilon) \to 0, \text{ as } n \to \infty \tag{8.7}$$

  2. Functional equivariance;
  3. Efficiency, i.e. it achieves the Cramér–Rao lower bound when the sample size tends to infinity. This means that no consistent estimator has lower asymptotic mean squared error than the MLE (or other estimators attaining this bound), which also means that MLE has asymptotic normality.

     $$\sqrt{n}\,\left(\widehat{\theta}_{\text{mle}} - \theta_0\right) \ \xrightarrow{d}\ \mathcal{N}\left(0,\, \mathcal{I}(\theta_0)^{-1}\right)\,, \tag{8.8}$$

     where $\mathcal{I}$ is the Fisher Information Matrix.

     If asymptotic normality holds, then asymptotic efficiency falls out because it immediately implies

     $$\hat{\theta} \ \xrightarrow{d}\ \mathcal{N}\left(\theta_0,\, (\mathcal{I}_n(\theta_0))^{-1}\right)\,, \tag{8.9}$$

     I use the notation $\mathcal{I}_n$ for the Fisher Information for $X = (X_1, \dots, X_n)$ (finite sample) and $\mathcal{I}$ for the Fisher Information for a single $X_i \in X$. Therefore, if the data provided are i.i.d, $\mathcal{I}_n = n\mathcal{I}$

  4. Second-order efficiency after correction for bias.

The asymptotic distribution of the MLE estimators can be used for constructing approximate confidence intervals. Alternatively, bootstrap confidence intervals can also be constructed and this may be especially suitable for small sample sizes.

- **Bayes estimators:** in the classical approach, the parameter, θ, is thought to be an unknown, but fixed quantity. A random sample $X_1, X_2, \ldots, X_n$ is drawn from a population indexed by θ and, based on the observed values in the sample, knowledge about the value of θ is obtained. In the Bayesian approach θ is considered a quantity whose variation can be described by a probability distribution (called the *prior distribution*). This is subjective distribution, based on the experimenter's belief and is formulated before the data are seen (hence the name prior distribution). A sample is then taken from a population indexed by θ and the prior distribution is updated with this samples information. The updated prior is called the *posterior distribution* This updating is done with the use of Baye's Rule, hence the name Bayesian statistics.

  $$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta} \tag{8.10}$$

  Notice that the posterior distribution is a conditional distribution, conditional upon observing the sample. The posterior distribution is now used to make statements about θ, which is still considered a random quantity. For instance, the mean of the posterior distribution can be used as a point estimate to θ:

  $$\hat{\theta} = E(\theta|x) \tag{8.11}$$

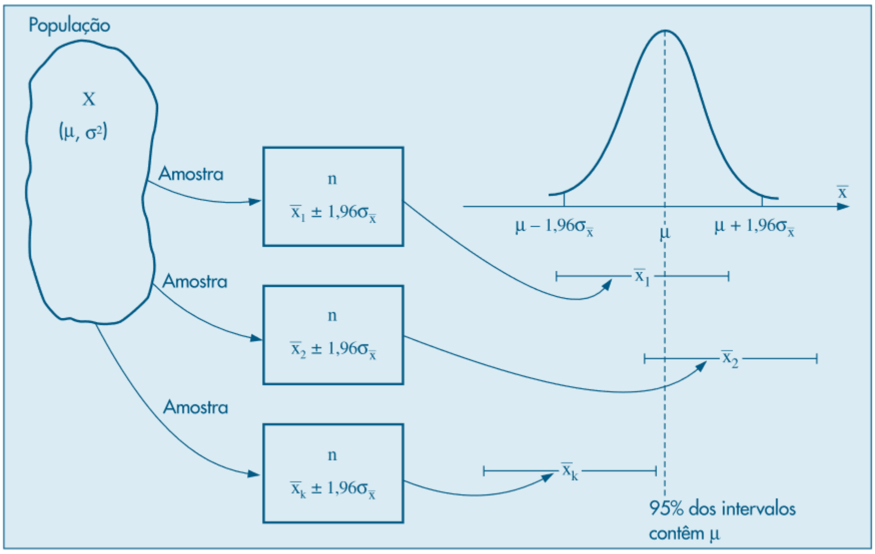
8.3 Some key concepts

8.3.1 Confidence Interval

Importante lembrar que um intervalo de confiança é utilizado para um parâmetro e não para uma variável aleatória. Se pudéssemos construir uma quantidade grande de intervalos (aleatórios) (lim inf, lim sup) (todos baseados em amostras de tamanho n, 95% deles conteriam o parâmetro μ)

FIGURE 8.1: ConfidenceInterval

Figura 11.3: Significado de um IC para μ, com $\gamma = 0{,}95$ e σ^2 conhecido.


População
X
(μ, σ²)
Amostra
Amostra
Amostra
n
$\bar{x}_1 \pm 1{,}96\sigma_{\bar{x}}$
n
$\bar{x}_2 \pm 1{,}96\sigma_{\bar{x}}$
n
$\bar{x}_k \pm 1{,}96\sigma_{\bar{x}}$
$\bar{x}$
$\mu - 1{,}96\sigma_{\bar{x}}$
μ
$\mu + 1{,}96\sigma_{\bar{x}}$
$\bar{x}_1$
$\bar{x}_2$
$\bar{x}_k$
95% dos intervalos contêm μ

8.3.2 Bootstrapping

Método de reamostragem que trata uma amostra como uma população finita; então são geradas amostras a partir da original, para estimar característica populacionais e fazer inferência sobre a população.

É uma classe de métodos de Monte Carlo não paramétricos que estimam a distribuição da população por reamostragem.

O termo "bootstrap" pode ser dirigido a bootstrap não-paramétrico ou bootstrap paramétrico. Seguiremos com o primeiro.

A distribuição da população finita representada pela amostra pode ser encarada como uma pseudo-população, com características análogas às da verdadeira população. Através da geração repetida de amostras aleatórias, com reposição, desta pseudo-população (reamostragem), a distribuição de amostragem de uma estatística pode ser estimada.

O bootstrap gera amostras aleatoriamente a partir da distribuição empírica da amostra.

Propriedades de um estimador tal como o viés ou o erro padrão podem ser estimadas por reamostragem.

É possível utilizar bootstrapping para calcular um intervalo de confiança para avaliar a acurácia de um modelo em um conjunto de dados de teste. Inclusive, para estudar AUCROC.

8.3.3 Methods of Evaluating Estimators

- **Standard Error (Erro Padrão)**: According to Bussab, if $\hat{\theta}$ is a estimator for θ, we call the standard error (erro padrão) of $\hat{\theta}$ the following quantity:

$$EP(\hat{\theta}) = \sqrt{Var(\hat{\theta})} \tag{8.12}$$

The variance of $\hat{\theta}$ depends of distribution parameters. Generally, to obtain an estimate of the standard error, we use:

$$\widehat{EP}(\hat{}\theta) = \sqrt{\widehat{Var}(\hat{}\theta)} \tag{8.13}$$

The standard error for the mean $\bar{X}$ of a sample of size n is $\sqrt{Var(X)/n}$

- **Mean Square Error (MSE) of an Estimator:** the mean square error (MSE) of an estimator $\hat{\theta}$ of a parameter θ is $E[(\hat{\theta} - \theta)^2]$. Do not confuse the MSE with the RMSE used to evaluate the predictions of a regression model.

8.3.4 Fisher Information:

in mathematical statistics, the Fisher information (sometimes simply called information) is a way of measuring the amount of information that an observable random variable X carries about an unknown parameter θ of a distribution that models X.

$$\mathcal{I}(\theta) = \operatorname{E}\left[\left(\frac{\partial}{\partial\theta}\log f(X;\theta)\right)^2\middle|\theta\right] = -\operatorname{E}\left[\left(\frac{\partial^2}{\partial\theta^2}\log f(X;\theta)\right)\middle|\theta\right] = \int_{\mathbb{R}}\left(\frac{\partial}{\partial\theta}\log f(x;\theta)\right)^2 f(x;\theta)\,dx, \tag{8.14}$$

8.3.5 Hessian Matrix:

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a function taking as input a vector $x \in \mathbb{R}^n$ and outputting a scalar $f(x) \in \mathbb{R}$. If all second partial derivatives of f exist, then the Hessian matrix H of f is a square $n \times n$, usually defined and arranged as follows:

$$\mathbf{H}_f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{8.15}$$

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:
`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.
If you want to completely hide the links, you can use:
`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.
If you want to have obvious links in the PDF but not the printed text, use:
`\hypersetup{colorlinks=false}`.