The “Geometry” of Naïve Bayes: Teaching Probabilities by “Drawing” Them

Giorgio Maria Di Nunzio

Department of Information Engineering

University of Padua

Via Gradenigo 6/a, 35131, Padova, Italy

Abstract

Educational Data Mining and Learning Analytics are interdisciplinary fields that exploit statistical, machine-learning, and data-mining algorithms over the different types of educational data. The application of data mining techniques to these educational datasets that come from educational environments allows researchers to address important educational questions. Student learning data collected by online learning systems are explored to develop predictive models in order to measure data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs. In this chapter, we focus on students that are studying foundations of machine learning and in particular probabilistic models for classification. The idea is to build an environment where students are given exercises that should be solved by interacting directly with the mathematical model by means of visual features. Our main goal is to build an interactive tool that addresses the following problems: teach probabilities and the probabilistic classifier in an innovative way, by breaking down learning into small components that can be analyzed and then adapted for each student; use simple geometrical primitives that allow non-experts to understand intuitively how probabilistic classifiers work; distribute the open source code of the application to make this approach available to a wider audience.

Keywords: Naïve Bayes, Bayesian Classification, Bayesian Decision Theory, Visual Data Analytics

The “Geometry” of Naïve Bayes: Teaching Probabilities by “Drawing” Them

Introduction

Educational Data Mining (EDM) is an emerging discipline that studies methods for exploring the data that come from educational settings, and uses those methods to better understand students and the settings in which they learn, as discussed by R. S. Baker and Yacef (2009, 1). A recent survey by the U.S. Department of Education (2012) gives a detailed overview of how EDM is currently applied in institutions, what kinds of questions it can answer, and the relationships with other research fields like Learning Analytics (LA). In general, EDM is more focused on the process of breaking down learning into small components that can be analyzed and then adapted into software designed for students rather than understanding entire systems and supporting human decision making (Siemens & Baker, 2012). Student learning data collected by online learning systems are then explored to develop predictive models by applying educational data mining methods that classify data or find relationships. Indeed, computer-supported interactive learning methods and tools have opened up opportunities to collect and analyze student data, to discover patterns and trends in those data, and to make new discoveries and test hypotheses about how students learn. LA is a closely related field with more emphasis on simultaneously investigating automatically collected data along with human observation of the teaching and learning context (Duval & Verbert, 2012). As defined in the First International Conference on Learning Analytics and Knowledge (LAK 2011): “Learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs”. [1] In the context of Massive Online Open Courses (MOOCs), for example like Khan Academy [2] or Coursera [3], the use of LA becomes crucial. Tools that

[1] https://tekri.athabascau.ca/

[2] https://www.khanacademy.org/

[3] https://www.coursera.org/

provide insights about this learning process are required to analyze and to interpret students’ learning processes on a large scale (Valiente, Merino, Leony, & Kloos, 2015).

EDM and LA are interdisciplinary fields that exploit statistical, machine-learning, and data-mining (DM) algorithms over the different types of educational data. The application of DM techniques to these educational datasets that come from educational environments allows researchers to address important educational questions as suggested by Romero and Ventura (2013). The application of traditional DM techniques to educational data is not trivial and requires some thought (Romero & Ventura, 2010):

> DM tools are normally designed more for power and flexibility than for simplicity. Most of the current DM tools are too complex for educators to use and their features go well beyond the scope of what an educator may want to do. For example, on the one hand, users have to select the specific DM method/algorithm they want to apply/use from the wide range of methods/algorithms available on DM. On the other hand, most of the DM algorithms need to be configured before they are executed. Users have to provide appropriate values for the parameters in advance in order to obtain good results/models, and therefore, the user must possess a certain amount of expertise in order to find the right settings.

Romero and Ventura (2010) propose a solution to this problem which comprises the development of wizard tools that use a default algorithm for each task and parameter-free DM algorithms to simplify the configuration and execution for non-expert users. In this respect, Visual Data Mining can help researchers to examine the streams of data at the right level of abstraction through appropriate visual representations and to take effective actions in real-time (Keim, Kohlhammer, Ellis, & Mansmann, 2010). Finally, EDM tools should be open source and/or freely available in order for them to be used by a much wider and broader population. An analysis made by Romero and Ventura (2013) shows that most of the current specific EDM tools are not available for download.

Main Contribution. In this chapter, we focus on students that are studying foundations of machine learning and in particular probabilistic models for classification. The idea is to build an environment in which students are given exercises that should be solved by interacting directly with the mathematical model by means of visual features. The interaction data can be used to study how many students struggled in that exercise, who could not do the task, who did the task correctly at least once, and who obtained proficiency in this exercise. In the same way that MOOCs capture student actions (i.e. Khan Academy monitor each time a student attempted to answer an exercise or earned a badge for completing a task), these data can be transformed into useful information that can be exploited to improve the learning process (Valiente et al., 2015).

Our main goal is to build an interactive tool that addresses the following problems:

- Teach probabilities and the probabilistic classifier in an innovative way, by breaking learning down into small components that can be analyzed and then adapted for each student;
- Use simple geometrical primitives that allow non-experts to understand intuitively how the probabilistic classifier work; therefore, tools are designed to be easier for educators and students;
- Distribute open source code of the application to make this approach available to a wider audience.

Based on the idea of Likelihood Spaces (Singh & Raj, 2004), we present a geometric interpretation of one of the most used probabilistic classifiers in the literature: the Naïve Bayes classifier. We introduce the properties of the two-dimensional representation of probabilities proposed by Di Nunzio (2009, 2014) which allows us to provide an adequate data visualization approach to understand, step by step, how to present complex concepts like parameter optimization and cost sensitive learning in an easy and intuitive way. At each step, we suggest exercises that can be monitored to track the learning curve of the student. We also apply this geometrical interpretation to a real case scenario of text categorization (Sebastiani, 2002) to show how this intuitive visualization can be used effectively not only for teaching probabilities but also for analyzing data.

Related Works

One of the key areas of applications of EDM is improvement of student models that would predict student’s performances with high accuracy. Dangi and Srivastava (2014) study the prediction of students performance, knowledge, and score by means of a Naïve Bayes classifier. The accuracy of the prediction highly depends on the choice of the most relevant variables that describe the data set. This can be achieved by means of feature selection techniques (Ramaswami & Bhaskaran, 2009). As previously mentioned, LA models the behavior and performance of students while they use learning systems; nevertheless, students’ behavior outside of the system may also influence how well the students learn. Xing and Goggins (2015) study off-task behavior in which students’ attention becomes lost and disengaged from the learning environment and activities by means of Naïve Bayes classifiers, the type of classifiers that we are going to study in this chapter.

Interactive Machine Learning (IML) is a relatively new area of Machine Learning (ML) where interaction with users allows ML models to be updated fast and very accurately. In IML, even non-expert users can solve machine learning problems with minimum effort by means of intuitive visualization tools (Amershi, Cakmak, Knox, & Kulesza, 2014). It has also been shown that cooperation between humans and machine learning algorithms is a key point for building classification algorithms effectively (Ankerst, Ester, & Kriegel, 2000; Ware, Frank, Holmes, Hall, & Witten, 2002). The Interactive and Classification approach presented by Amershi et al. (2015) has been designed to enable lay people to train interactively both classifiers and extractors (functions that map an input item to a sequence of annotated segments) using large datasets containing 100 million examples or more. Exploratory learning environments are educational tools designed to foster learning by supporting students in freely exploring relevant instructional material. Amershi and Conati (2009) study, among other things, an Adaptive Coach for Exploration (ACE) learning environment to test their user modelling framework. This tool allows students to study quadratic equations by means of interactions that are very similar to the ones presented in this work.

The Geometry of Naïve Bayes Classification

Naïve Bayes classifiers have been widely used in the literature of Data Mining (DM) and Machine Learning (ML) since they are easy to train and reach satisfactory results which often can be used both as a baseline for comparison purposes with and as an assessment of how difficult the classification is (Han, Kamber, & Pei, 2011, Chapter 8). Building these types of classifiers is easy, but their optimization is often lacking if not missing all together. In this work, we propose a visualization approach that directly involves users in the process of building the probabilistic classifier, as suggested by Ankerst et al. (2000), in order to obtain a twofold result: first, the pattern recognition capabilities of a human can be used to increase the effectiveness of the classifier; and second, a visualization of the probabilistic model can be used to teach non-experts how these kinds of models work.

Based on the idea of Likelihood Spaces (Di Nunzio, 2009; Singh & Raj, 2004) which represent probabilities on a two-dimensional space, we have developed, designed and implemented a Web application in R [4] using a package named Shiny (Chang, 2015) which is a new package of the R programming language which allows for rapid prototyping of interactive Web applications. [5] The source code of the application is available freely for download. [6]

[4] http://www.r-project.org/

[5] http://shiny.rstudio.com/

[6] https://github.com/gmdn/educational-data-mining

In summary, the main steps of our approach are:

- A geometrical definition of the Bayes' rule;
- A geometrical definition of NB classifiers;
- An interactive Web application to show how these concepts work in practice both on a toy-problem and on a real case scenario.

In the remainder of this section, we introduce the basic mathematical notation and definitions that will be used to build the visualization tool.

Mathematical Notation

In general, the problem of classification of objects requires a set of predefined classes $C = \{c_1, ..., c_i, ..., c_n\}$ that are used to organize documents. A generic object o can belong to one or more classes (or even none of them) and the act of classification is also called 'labelling'. In this work, we deal with binary classification problems. A binary classification problem is a special case of single-label classification in which the object o belongs to one category, the "positive" class is indicated with c_i (or c without subscript when there is no risk of misinterpretation), or its complement, the "negative" class $\bar{c_i}$ (or $\bar{c}$). Binary classification is actually a standard approach in ML and DM to break down multi-class problems into several binary classification problems (Rocha & Goldenstein, 2014).

Deciding whether to label a document or not requires a careful evaluation of some function which minimizes the classification error. Among the many possible choices described in the literature, probabilistic classifiers have the nice property of computing the uncertainty on such decisions; for example, calculating the probability that an object o belongs to class c. We use the usual simplified notation for the probability of events, like $P(c)$ and $P(o)$ for the probability of a class and the probability of an object, respectively; the conditional probabilities are instead written in the usual way $P(c|o)$. [7]

[7] We use values and omit variables to simplify formulae. For example, $P(\bar{c}|o)$ instead of $P(C = \bar{c}|O = o)$.

In its simplest form, a probabilistic classifier puts o into category c if the following statement is true:

$$P(c|o) > P(\bar{c}|o) \tag{1}$$

that is, if the probability of the class c given o is greater than the probability of its complement $\bar{c}$ given o. In order to justify this statement, and develop the two-dimensional representation of probabilities, we need to add one important building block: Bayesian Decision Theory.

Bayesian Decision Theory

Bayesian Decision Theory is a statistical approach to the problem of classification of objects. This approach is based on quantifying the tradeoffs between classification decisions and the costs that accompany such decisions (Duda, Hart, & Stork, 2000, Chapter 2). For example, let us suppose that we need to diagnose a rare disease; let us call c the category of the people with this disease and $\bar{c}$ the category of healthy people. We know from experience and past tests that the probability of the disease is $P(c) = 0.001$, that is to say one out of 1000 people has the disease; and therefore $P(\bar{c}) = 0.999$ (this example was inspired by Kruschke (2014, Chapter 5)). These two probabilities reflect our prior knowledge of how likely the disease is distributed within the population. Suppose that we are now forced to make a decision about the health of a patient without any information about the patient. If a decision must be made, the most reasonable decision rule (and most correct under some conditions) is: if $P(\bar{c}) > P(c)$ then the patient is healthy. In fact, with this decision rule, we would be correct 999 times out of 1,000. However, in real situations:

- We usually do not make decisions with so little information. Objects (in the examples patients) are, in general, described by features that can be measured; for example, we may ask the patient to undergo some medical tests that measure the level of white cells in his/her blood before making any decision.

- The costs of decisions are rarely symmetric. For example, a patient with a disease that is classified as healthy is a decision that may have deadly consequences (hence a very high cost). The opposite situation may have negative consequences for the patient (maybe psychological for resulting positive to the disease) but less costly.

Bayesian analysis allows us to infer the posterior belief we have on the patient based on some evidence (e.g. the result of a blood test). For example, we can adjust our belief on the probability of the category c by applying Bayes'rule:

$$\underbrace{P(c|o)}_{\text{posterior}} = \frac{\overbrace{P(o|c)}^{likelihood}\ \overbrace{P(c)}^{prior}}{\underbrace{P(o)}_{data}} \tag{2}$$

Bayesian Decision Theory allows us to formally define the risk in taking a decision (classify a patient as healthy), assign costs to these decisions, and find the decision that minimizes the risks with that particular action. Suppose that we observe an object o, the risk in classifying o in category c is defined as a weighted sum:

$$R(c|o) = \lambda_{cc} P(c|o) + \lambda_{c\bar{c}} P(\bar{c}|o) \tag{3}$$

where $\lambda_{c\bar{c}}$ is the *loss* we incur when we predict c while the true category for the object o is $\bar{c}$. In the example of the patient and the disease, $\lambda_{c\bar{c}}$ should be very high because we classify a person with a disease as healthy, while λ_{cc} should be equal to zero because we predict the correct case. The risk in assigning o to $\bar{c}$ is defined accordingly:

$$R(\bar{c}|o) = \lambda_{\bar{c}c} P(c|o) + \lambda_{\bar{c}\bar{c}} P(\bar{c}|o) \tag{4}$$

The optimal classification choice is the one that minimizes the overall risk; for example, we assign the object o to c when

$$R(c|o) < R(\bar{c}|o) \tag{5}$$

that is,

$$\begin{aligned}
\lambda_{cc}P(c|o) + \lambda_{c\bar{c}}P(\bar{c}|o) &< \lambda_{\bar{c}c}P(c|o) + \lambda_{\bar{c}\bar{c}}P(\bar{c}|o) \\
(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})P(\bar{c}|o) &< (\lambda_{\bar{c}c} - \lambda_{cc})P(c|o) \\
P(\bar{c}|o) &< \frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}P(c|o) \qquad (6)
\end{aligned}$$

which, for $\lambda_{cc} = \lambda_{\bar{c}\bar{c}} = 0$ and $\lambda_{\bar{c}c} = \lambda_{c\bar{c}} = 1$ (also known as *zero-one* loss function), we obtain the intuitive, but now mathematically sound, solution presented in Eq. 1, $P(c|o) > P(\bar{c}|o)$. By substituting Eq. 2 into Eq. 6 we obtain:

$$\frac{P(o|\bar{c})P(\bar{c})}{P(o)} < \frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}\frac{P(o|c)P(c)}{P(o)} \qquad (7)$$

This last equation is the main building block we need to study the “geometry” of probabilistic classifiers.

Two Dimensional Probabilities

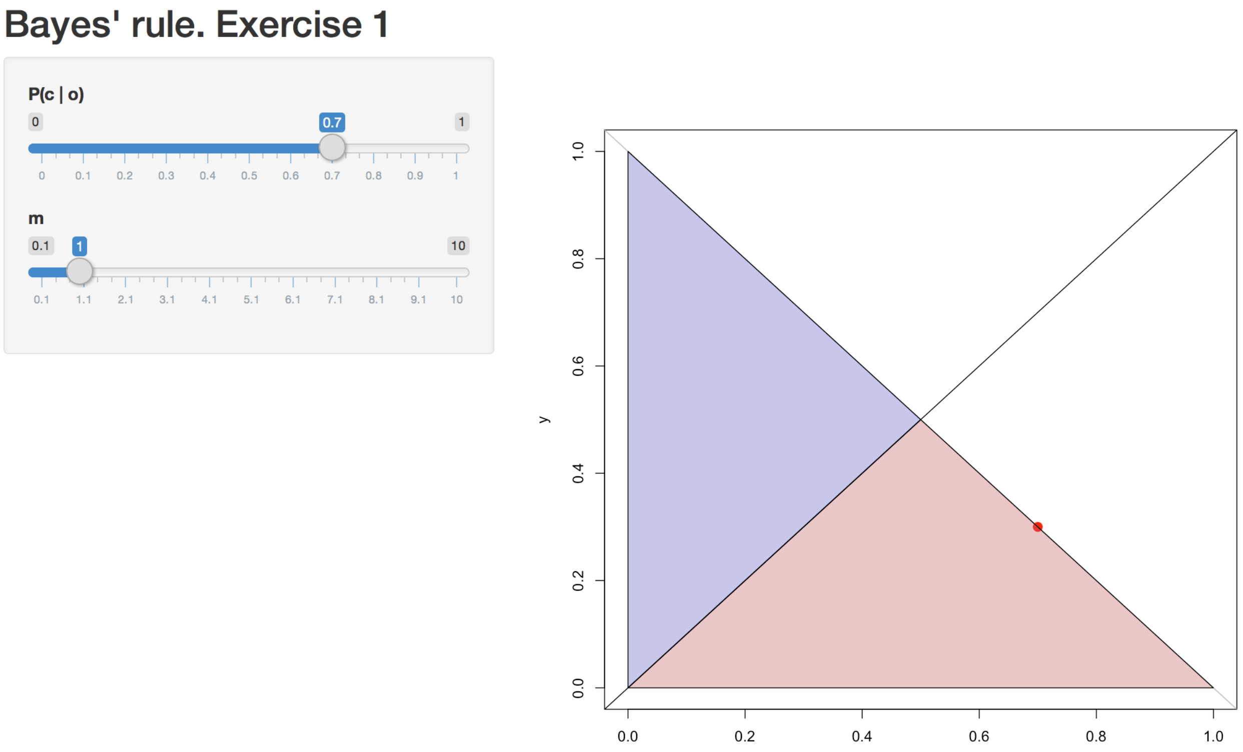
The two dimensional definition of the NB classifier starts from Eq. 6. If we rewrite it in the following way:

$$y < mx \qquad (8)$$

we can immediately make some considerations:

- $x = P(c|o)$ and $y = P(\bar{c}|o)$ can be seen as two coordinates of a Cartesian space.
- Since $P(\bar{c}|o) = 1 - P(c|o)$, i.e. $y = 1 - x$, a point with coordinates (x, y) lies on the segment with endpoints $(0, 1) - (1, 0)$.
- The decision line $y < mx$ splits the plane in two: all the points that are below the line are assigned to c, all the points above the line to $\bar{c}$.

In Figure 1, we show the first example of the interface to teach how classification works on a two-dimensional space. On the left side of the figure, we have the sliders that control the posterior probability of an object $P(c|o)$ (and consequently the probability



Bayes' rule. Exercise 1

P(c | o)

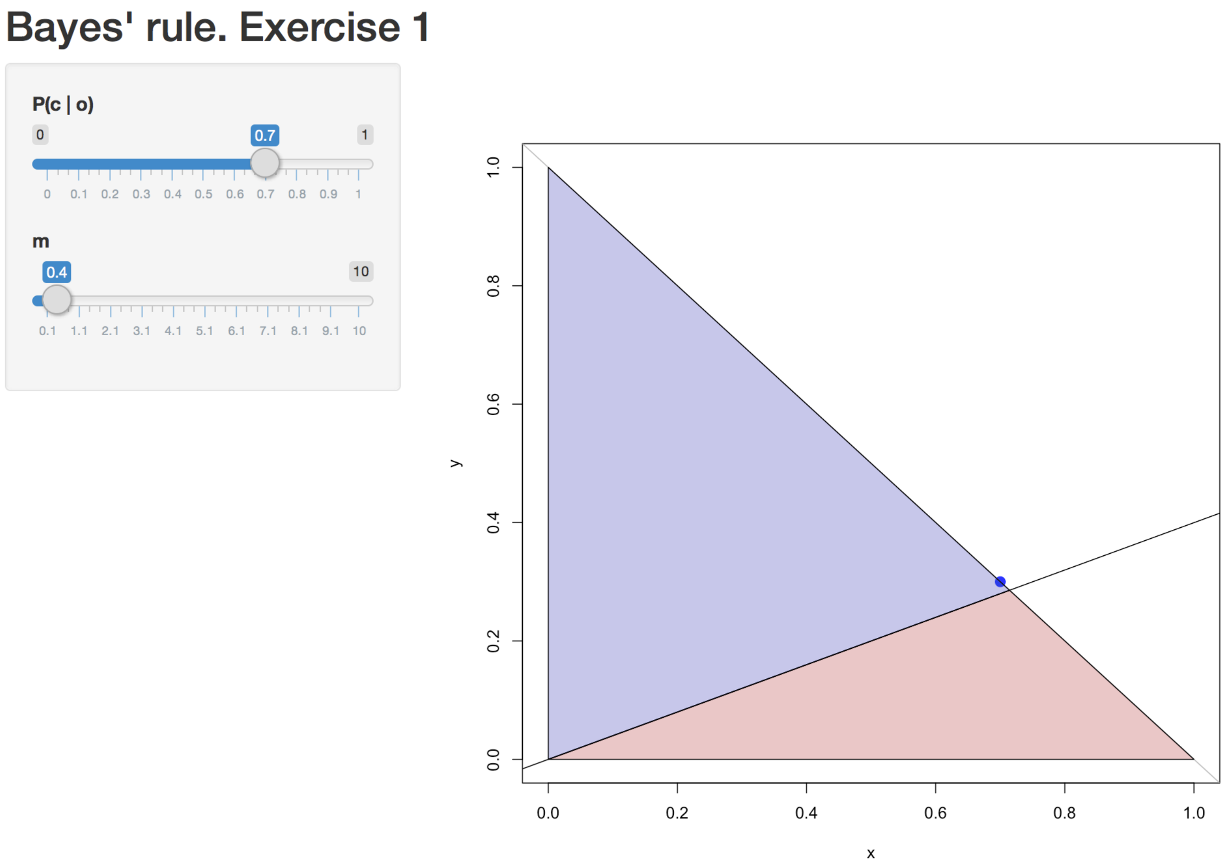
0.7

m

1

Figure 1. Two-dimensional representation of probabilities. Coordinates are $x = P(c|o)$ and $y = P(\bar{c}|o)$.

$P(\bar{c}|o) = 1 - P(c|o))$ and the angular coefficient m of the decision line. Initially, the probability that the object belongs to class c is $P(c|o) = 0.7$ and $m = 1$ which is the value of m that corresponds to the standard zero-one loss function (Equation 7, $\lambda_{cc} = \lambda_{\bar{c}\bar{c}} = 0$, $\lambda_{\bar{c}c} = \lambda_{c\bar{c}} = 1$). With these settings, the object is classified under category c. What if the true class of the object is not c? Is there anything we can do to classify that object into class $\bar{c}$? We cannot change the value of the posterior probability (because that number is actually what we have computed), but we can adjust the slope of the decision line. By decreasing m, we can reach the point where the decision line is ‘flat’ enough to put the



Bayes' rule. Exercise 1

P(c | o): 0.7

m: 0.4

Figure 2. Two-dimensional representation of probabilities. Coordinates are $x = P(c|o)$ and $y = P(\bar{c}|o)$. A change in the slope m results in a different classification decision.

point in the space above the line. This limit can be computed by rearranging Eq. 8:

$$\frac{y}{x} < m \tag{9}$$

when m is greater than the ratio y/x the point is below the line (and classified under c), while if m is less than that ratio, the point is above the line (and classified under $\bar{c}$). In the example, when $m < 0.3/0.7 \simeq 0.43$, the points are classified under $\bar{c}$, as shown in Figure 2.

Exercise.

- Set a value for $P(c|o)$ and compute the angular coefficient m needed to classify it under c.

- Compute the values of the coefficients λ_{cc}, $\lambda_{\bar{c}\bar{c}}$, $\lambda_{\bar{c}c}$, $\lambda_{c\bar{c}}$ that produce this solution. Describe the two types of possible combinations of costs that are needed.

Working with Likelihoods and Priors Only

In the first example, we were able to input directly the value of the posterior probability $P(c|o)$. In real cases, we need Bayes' rule to compute this probability. Therefore, we need to rewrite coordinates in terms of the prior and the likelihood:

$$\frac{P(o|\bar{c})P(\bar{c})}{P(o|c)P(c) + P(o|\bar{c})P(\bar{c})} < \frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})} \frac{P(o|c)P(c)}{P(o|c)P(c) + P(o|\bar{c})P(\bar{c})} \tag{10}$$

where we substituted $P(o)$ with the sum of the conditional probabilities. [8] In this case,

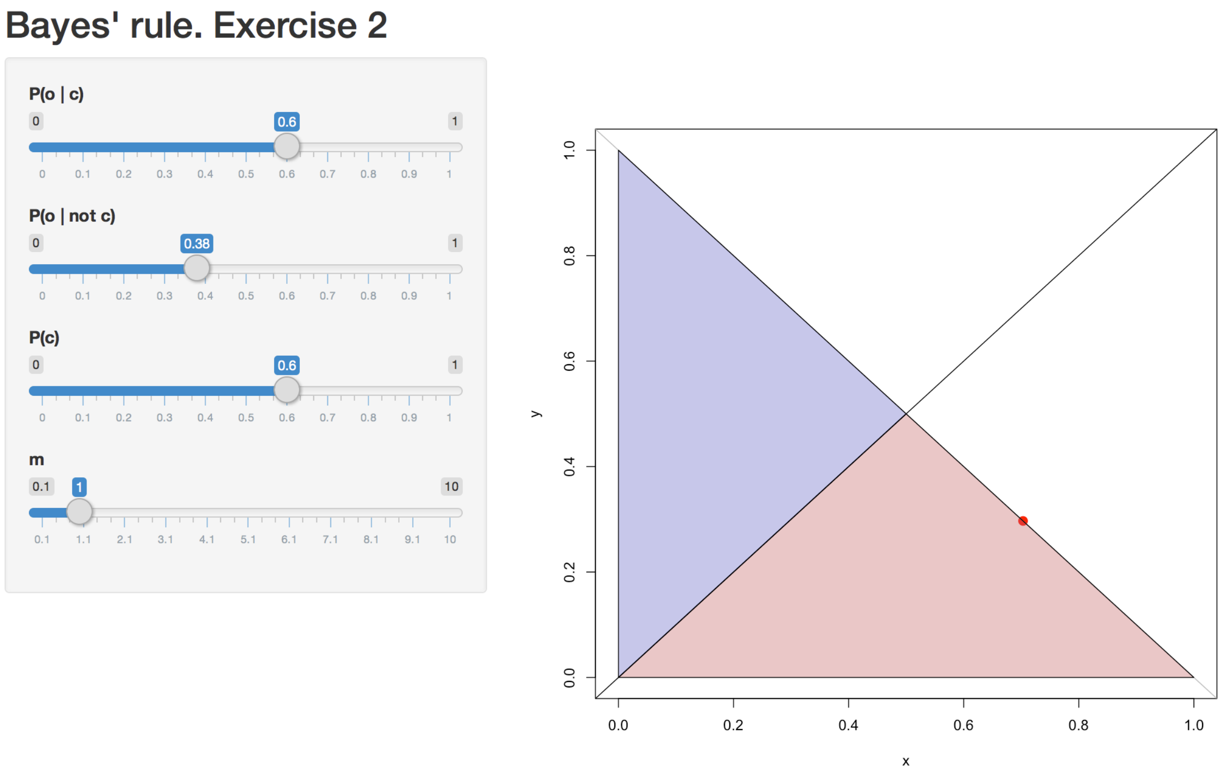
- Coordinates x and y are now computed via Bayes' rule.
- Priors on the categories $P(c)$ and $P(\bar{c})$ must sum to one $P(c) + P(\bar{c}) = 1$.
- Class conditional probabilities and likelihood functions can take any value between zero and one and the sum of the two likelihoods does not have to sum to one (that is, $P(o|c) + P(o|\bar{c}) \neq 1$, in general).

In Figure 3, we show the interface with the new sliders that allow users to directly interact with likelihood functions and priors. The first example shows a point that is below the zero-one loss function ($m = 1$) with the same coordinates as in the previous example. When a zero-one loss function is used, two situations are worth more thorough investigation:

1. if $P(c) = P(\bar{c})$, we assign the object to category c when $P(o|c) > P(o|\bar{c})$.
2. If $P(o|c) = P(o|\bar{c})$ then we assign o to category c when $P(c) > P(\bar{c})$.

In both cases, whenever we have a complete uncertainty (equal probabilities) the best thing to do in terms of minimizing the risk is to rely on the remaining information, either the

[8] This is the passage $P(o) = P(c, o) + P(\bar{c}, o) = P(o|c)P(c) + P(o|\bar{c})P(\bar{c})$



Bayes' rule. Exercise 2

P(o | c) 0.6

P(o | not c) 0.38

P(c) 0.6

m 1

Figure 3. We use Bayes' rule to compute the posterior probability in terms of priors and likelihood functions.

prior on a class or the likelihood of an object. Given some values of likelihoods and priors, it is always possible to find a loss function (angular coefficient m) that changes the decision of classification.

Exercise.

- Set the initial likelihood and priors. Find the value of m such that the object is classified under $\bar{c}$.

- Set the likelihoods, then describe what is the relation between the prior on class c and the angular coefficient m.

De-normalizing Probabilities

The posterior probabilities have a normalization factor given by the probability of the object $P(o)$ which is equal for both sides of the inequality. For this reason, it is very common to cancel it from both sides of the decision function of Eq. 7 and obtain:

$$P(o|\bar{c})P(\bar{c}) < \frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})} P(o|c)P(c) \tag{11}$$

The new coordinates of the point are $x' = \alpha x$ and $y' = \alpha y$ where $\alpha = P(o)$. This new interpretation of the probabilities is crucial for the effectiveness of the classification (it will be clear in the next sections why this small detail dramatically changes the decision of classification). We can describe some geometrical properties:

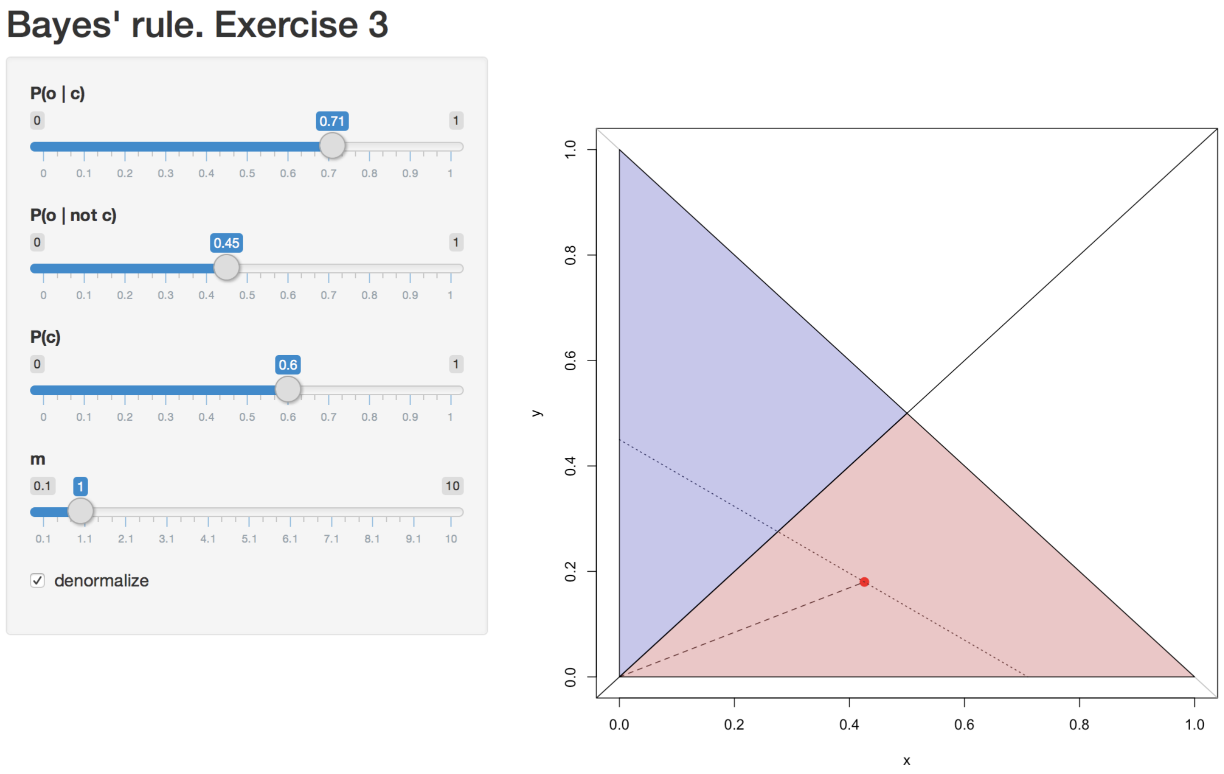
- The new coordinates x' and y' are the old ones multiplied by the same positive factor α which happens to be between zero and one. This means that the new coordinates lie on the segment with endpoints (0, 0) - (x, y).
- If the normalized point was below the decision line $y = mx$, the de-normalized point will remain below the same decision line, that is if $y < mx$ then $y' < mx'$. [9]
- Once the likelihoods are fixed, the first point moves along the line $y = -\frac{P(o|\bar{c})}{P(o|c)}x + P(o|\bar{c})$. This means that if we want to study the new coordinates in terms of the prior probability $P(c)$, the abscissa is $x = P(o|c)P(c)$ while the ordinate is $y = P(o|\bar{c})(1 - P(c))$.

In Fig. 4, we show an interface that allows users to de-normalize the posterior probability. The segment along which the point can move is highlighted with dotted lines.

The formulation of the decision function shown in Eq. 11 can also be rewritten in the following way:

$$\frac{P(o|\bar{c})}{P(o|c)} < \frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})} \frac{P(c)}{P(\bar{c})} \tag{12}$$

[9] This is true when we use a decision line that passes through the origin of the axis. As soon as we add an intercept q, $y = mx + q$, then the same point may be classified differently from the initial coordinates.



Bayes' rule. Exercise 3

P(o | c) 0.71

P(o | not c) 0.45

P(c) 0.6

m 1

☑ denormalize

Figure 4. The de-normalized coordinates lie on the segment with endpoints $(0, 0)$ - (x, y). Once the likelihoods are fixed, the de-normalized point moves along the line $y = -\frac{P(o|\bar{c})}{P(o|c)}x + P(o|\bar{c})$.

where the term on the left is called *likelihood ratio*. This formulation is important for two reasons:

- It is related to the formulation of classification in terms of the minimax criterion and the Neyman-Pearson criterion (Duda et al., 2000, Chapter 2).
- It shows that the loss function coefficients can be used to balance the ratio $P(c)/P(\bar{c})$ which, in cases of unbalanced classes, can be extremely high (or low) (Mladenic & Grobelnik, 1999).

Exercise.

- Consider the decision function shown in Eq. 11 and suppose that $P(c) = k$ with $k \ll 1$. What is the value of m that balances the disproportion between $P(c)$ and $P(\bar{c})$?.

- Fix the value of the two likelihood probabilities $P(o|c)$ and $P(o|\bar{c})$ and the angular coefficient m. Find the threshold of the probability $P(c)$ that changes the classification decision.

Naïve Bayes Approach

In real case scenarios, we estimate the likelihood function by means of the class conditional probability of the features of the object o. For example, let us assume that the objects we want to study are characterized by a set of three features $\mathcal{F} = \{F_1, F_2, F_3\}$. An object o is therefore a particular realization of these three features, and its likelihood for category c is:

$$P(o|c) = P(\{f_1, f_2, f_3\}|c) \tag{13}$$

The problem of computing this probability is that we need an amount of data that grows exponentially with the number of features (e.g. if the variables in F are binary, the probability table has $2^{|F|}$ entries). This is also called the *curse of dimensionality* (Hastie, Tibshirani, & Friedman, 2009). For this reason, it is very common to simplify the problem by means of a very strong assumption named *Naïve Bayes* assumption: all the features are conditionally independent given the class. In mathematical terms:

$$P(\{f_1, f_2, f_3\}|c) = \prod_{i=1}^{3} P(f_i|c) \tag{14}$$

For three features, the decision rule becomes:

$$\prod_{i=1}^{3} P(f_i|\bar{c})P(\bar{c}) < \frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})} \prod_{i=1}^{3} P(f_i|c)P(c) \tag{15}$$

Bernoulli Naïve Bayes. If the features that represent the object are binary (that is they can only assume a value equal to zero or one), the probability of each feature is described by a Bernoulli variable:

$$P(f_i|c) = \begin{cases} \theta & \text{if } f_i \text{ appears in the object} \\ 1-\theta & \text{if } f_i \text{ does not appear in the object} \end{cases} \tag{16}$$

where θ is the value of the probability of the feature being present or absent in the object. In Fig. 5 we show an example of an object that is represented by three features, f_1 and f_3 are present, while f_2 is absent.

In practice, there are at least two problems with this new assumption:

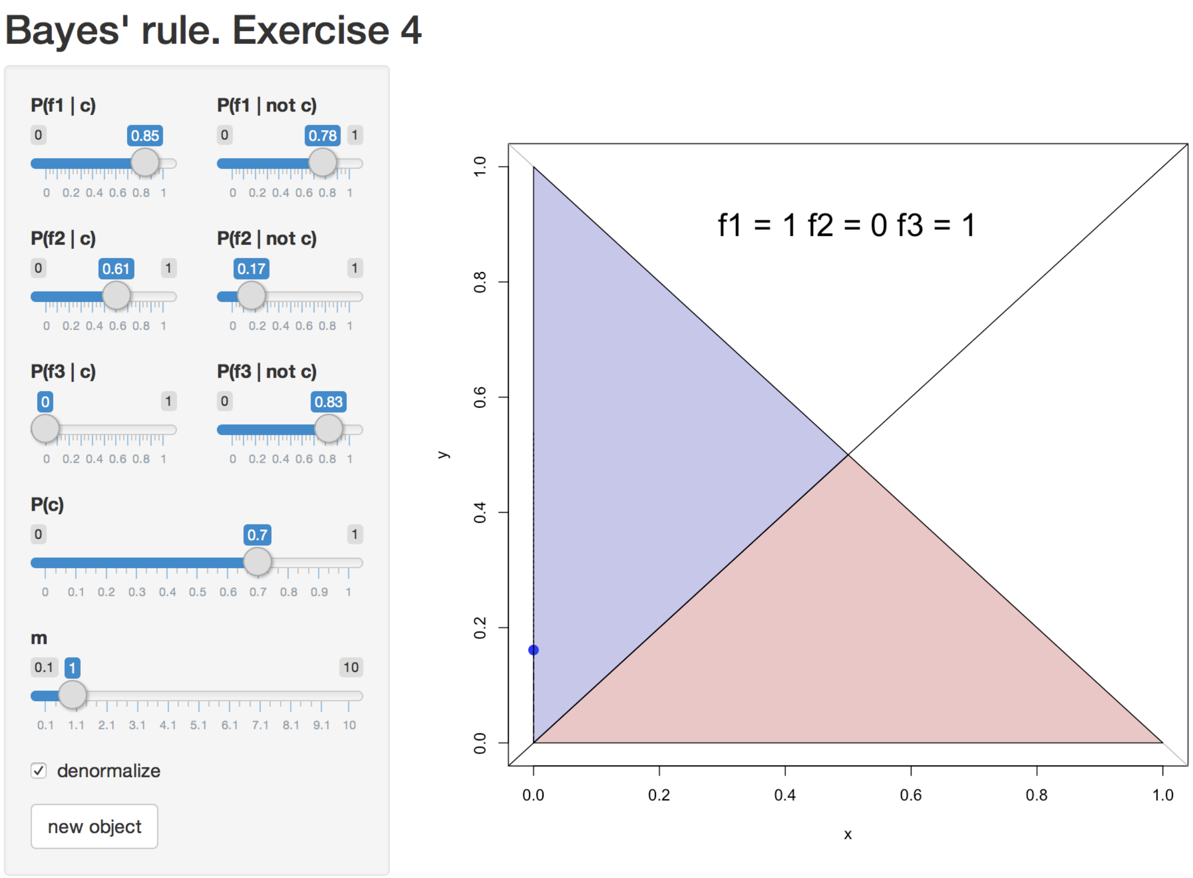
- When one of the features has probability equal to zero (or one), the whole likelihood goes to zero if the feature is present (or absent) in the object. This situation is shown in Fig. 5, where the coordinate x of the point is zero and feature f_3 has $P(f_3|c) = 0$.
- Since the likelihood of an object is the product of n conditional probabilities (where n is the number of features), the value of the probability $P(o|c)$ is very small. In general, not only are the points very close to the origin of the axes, but they are also equal to zero by approximation. [10]

The first problem can be solved by means of a probability smoothing approach (Hiemstra, 2009). The second problem requires the application of a monotonic function which preserves classification (we describe this passage later in this work). Before solving this last problem, we show an extension of the classic risk of Bayesian Decision Theory that allows us to draw a decision line that will perform better.

Exercise.

- Suppose that one of the features that describes the object o has probability equal to one. What happens if that feature is not present in the object?

[10] Suppose that, on average, the probability of a feature given a class is $P(f_j|c) \simeq 10^{-2}$ and all the features have a probability greater than zero to avoid $P(o|c) = 0$. With 100 features, the likelihood of an object will be, on average, $P(o|c) \simeq 10^{-200}$ which is very close to the limit of the representation of a 64 bit floating point number. In real situations, probabilities are much smaller than 10^{-2} and features can be easily tens of thousands; hence, all the likelihood functions would be equal to zero by approximation.



Bayes' rule. Exercise 4

P(f1 | c): 0.85

P(f1 | not c): 0.78

P(f2 | c): 0.61

P(f2 | not c): 0.17

P(f3 | c): 0

P(f3 | not c): 0.83

P(c): 0.7

m: 1

☑ denormalize

Figure 5. When one of the features has probability equal to zero (or one), the whole likelihood goes to zero if the feature is present (or absent) in the object.

- Suppose that one coordinate is equal to zero. Is there any value of m that can change the classification decision?

A New Decision Line: Far From the Origin

So far, we have used a decision line that passes through the origin of the axes. This solution is a consequence of a geometrical interpretation of the classical Bayesian Decision Theory approach according to the definition of risk given by Equation 3. A more sophisticated approach consists of assigning to each class a cost, independently of the posterior probability of the object o we are about to classify. As suggested by Di Nunzio

(2014), we can imagine a new risk that adds one element:

$$R(c|o) = \lambda_{cc} P(c|o) + \lambda_{c\bar{c}} P(\bar{c}|o) + \frac{\lambda_c}{P(o)} \tag{17}$$

where λ_c is constant for each object o and represents the cost of choosing c independently of the posterior probability $P(c|o)$ and $P(\bar{c}|o)$. With this new definition, we can rewrite the decision function in the following way:

$$\underbrace{P(o|\bar{c})P(\bar{c})}_{y} < \underbrace{\frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}}_{m} \underbrace{P(o|c)P(c)}_{x} + \underbrace{\frac{\lambda_{\bar{c}} - \lambda_c}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}}_{q} \tag{18}$$

The intercept q of the decision line is the new coefficient that we can use to optimize the classification decision. In particular, when the two costs are equal $\lambda_{\bar{c}} = \lambda_c$, the coefficient is zero and we return to a decision line that passes through the origin. When $\lambda_{\bar{c}} > \lambda_c$, the cost of choosing $\bar{c}$ is higher, and the decision line moves upwards reducing the area of classification for $\bar{c}$.

De-normalization Makes (Some) Problems Linearly Separable

By using the classic definition of risk, normalizing or de-normalizing coordinates does not change the classification decision. With the new decision function, this is not true anymore. The advantage of this new situation is evident when two non-linearly separable classes become linearly separable in the de-normalized version of the problem. In Fig. 6a, we show an example where three objects, one belonging to class c and the other two to class $\bar{c}$, cannot be separated by the classic linear decision. This is true also for the de-normalized version of the same problem, as shown in Fig. 6b. Instead, when the decision line of Eq. 18 is used, the intercept allows us to move from the origin and find the correct separation between the two classes.

Exercise.

- Set the values of the probabilities of the features given the class. Compute the de-normalized coordinates and find the parameters m and q of the decision line that optimize classification (when possible).

- In some cases, it may be possible to find decision lines with a negative angular coefficient. Find what are the costs of the loss function that produce these values and discuss whether these values are sensible or not (Elkan, 2001).

Likelihood Spaces, when Logarithms Make a Difference (or a Sum)

In a NB classifier, the likelihood of an object is the product of the conditional probabilities of the features that describe the object. This makes the value of the probability $P(o|c)$ so small that it is usually approximated with a value equal to zero. In order to avoid this arithmetical anomaly, we apply the logarithm to Eq. 6, a monotonic transformation of the probabilities, and we obtain:

$$\begin{aligned} \log(P(\bar{c}|o)) &< \log\left(\frac{\lambda_{\bar{c}c} - \lambda_{cc}}{\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}}}\right) + \log(P(c|o)) && (19) \\ \log(y) &< \log(m) + \log(x) && (20) \\ y_L &< q_L + x_L && (21) \end{aligned}$$

where $x_L = \log(P(c|o))$ and $y_L = \log(P(\bar{c}|o))$ are the coordinates in the logarithmic space. Note that when the logarithm is applied to the classic decision line, the rotation of the decision line in the data space corresponds to a shift in the logarithmic space. In Fig. 7, we show the same point of Fig. 1 projected into the logarithmic space. Note that the segment with endpoint (0,1)-(1,0) where normalized points lie becomes a sort of hyperbola in the logarithmic space.

De-normalizing in Likelihood Spaces

When the normalization factor $P(o)$ is canceled, we obtain the following coordinates in the likelihood space:

$$\begin{aligned} \log(P(o|\bar{c})P(\bar{c})) &< \log\left(\frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}\right) + \log(P(o|c)P(c)) && (22) \\ \log(P(o|\bar{c})) + \log(P(\bar{c})) &< \log\left(\frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}\right) + \log(P(o|c)) + \log(P(c)) && (23) \end{aligned}$$

Therefore, while in the original data space we have $x' = \alpha x$ where $\alpha = P(o)$, in the logarithmic space we obtain:

$$x_L = \log(x') = \log(\alpha x) = \log(P(o|c)) + \log(P(c)) - \log(P(o)) + \log(P(o)) \quad (24)$$

$$y_L = \log(y') = \log(\alpha y) = \log(P(o|\bar{c})) + \log(P(\bar{c})) - \log(P(o)) + \log(P(o)) \quad (25)$$

This means that the de-normalized coordinates in the logarithmic space are shifted by the same quantity $\log(P(o))$ towards minus infinity and parallel to the bisecting line of the third quadrant. In Fig. 8, we show an example of a de-normalized point in the likelihood space.

A New Decision in Likelihood Spaces

When we work in likelihood spaces, the decision line presented in Eq. 18 takes a particular form:

$$\log(P(o|\bar{c})P(\bar{c})) < \log\left(\frac{(\lambda_{\bar{c}c} - \lambda_{cc})}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})} P(o|c)P(c) + \frac{\lambda_{\bar{c}} - \lambda_{c}}{(\lambda_{c\bar{c}} - \lambda_{\bar{c}\bar{c}})}\right) \quad (26)$$

the logarithm on the right hand side of the inequality cannot be factorized into the sum of logarithms. Therefore, we have this type of logarithmic curve

$$\log(y') < \log(mx' + q) \quad (27)$$

that, given $m > 0$, for positive values of q is convex, while for $q < 0$ is concave. For $q = 0$ we obtain the classical decision line ($log(y) = log(m) + log(x)$). This curve allows us to separate points that have been denormalized in the likelihood space. For example, in Fig. 9 we show three points that cannot be separated when normalized, but they can be separated by the logarithmic curve in the likelihood space.

A Real Case Scenario: Text Categorization

In the previous sections, we presented the geometric interpretation of probabilistic classifiers on a two-dimensional space, and we described a set of parameters that can be tuned to optimize classification. In a real machine-learning setting these parameters need to be trained and validated using portions of the dataset available to train the classifier. For example, a k-fold cross validation can be used to find the parameters that minimize the error of the classifier (Duda et al., 2000, Chapter 9).

In Fig. 10, we show a real machine-learning scenario that uses a standard benchmark for text classification: the Reuters-21578 dataset [11]. The top 10 most frequent categories of the corpus were chosen as a benchmark. This Web application applies all the concepts presented in this paper. The idea is that even a non-expert can easily find a solution by visual inspection. The only difference is that we have two more parameters α and β that are used to change how probabilities are smoothed. [12] Moreover, the user has two windows: one dedicated to the training phase on the left, and one to check the performance on the validation set on the right. Performance measures are shown to give numerical feedback to the user, in addition to the visual feedback.

Final Remarks

EDM exploits DM algorithms over the different types of educational data. The application of DM techniques to these specific educational datasets that come from educational environments allows researchers to address important educational questions. However, most of the current DM tools are too complex for educators to use and their features go well beyond the scope of what an educator may want to do. Moreover, EDM tools should be open source and/or freely downloadable.

In this chapter, we have presented an approach to represent probabilities on a two-dimensional space with two goals: i) to teach probabilities that makes use of visual primitives that are very intuitive and exploit the capability of humans to find regular patterns; ii) to build a visual tool that makes use of a standard classification algorithm that can be used for real tasks. This algorithm can be optimized very efficiently even by

[11] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[12] https://gmdn.shinyapps.io/shinyK/

lay people by means of interactive tools.

The Web applications have been developed with a package of the R language that allows for rapid prototyping and can be easily embedded in other larger projects. It is completely open source and the aim is to embed this approach in frameworks like the Interactive and Classification (ICE) approach presented by Amershi et al. (2015).

Ultimately, this approach can be adapted to more complex analysis like the work by van de Sande (2013) where the Knowledge Tracing algorithm uses student performance at each opportunity to apply a skill to update the conditional probability that the student has learned that skill. In this case, the algorithm can be optimized by means of the visualization tool on a two-dimensional space.

References

Amershi, S., Cakmak, M., Knox, W. B., & Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, *35*(4), 105–120. Retrieved from http://www.aaai.org/ojs/index.php/aimagazine/article/view/2513

Amershi, S., Chickering, M., Drucker, S. M., Lee, B., Simard, P., & Suh, J. (2015). ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 337–346). CHI ’15. Seoul, Republic of Korea: ACM. doi:10.1145/2702123.2702509

Amershi, S. & Conati, C. (2009). Combining Unsupervised and Supervised Classification to Build User Models for Exploratory. *Journal of Educational Data Mining (JEDM)*, *1*(1), 18–71.

Ankerst, M., Ester, M., & Kriegel, H.-P. (2000). Towards an Effective Cooperation of the User and the Computer for Classification. In *Proceedings of the sixth acm sigkdd international conference on knowledge discovery and data mining* (pp. 179–188). KDD ’00. Boston, Massachusetts, USA: ACM. doi:10.1145/347090.347124

Baker, R. S. & Yacef, K. (2009). The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining (JEDM)*, *1*, 3–17. Retrieved from http://educationaldatamining.org/JEDM/images/articles/vol1/issue1/JEDMVol1Issue1_BakerYacef.pdf

Chang, W. (2015). *Shiny: Web Application Framework for R.* R package version 0.11. Retrieved from http://CRAN.R-project.org/package=shiny

Dangi, A. & Srivastava, S. (2014, December). Educational Data Classification Using Selective Naïve Bayes for Quota Categorization. In *MOOC, Innovation and Technology in Education (MITE), 2014 IEEE International Conference on* (pp. 118–121). doi:10.1109/MITE.2014.7020253

Di Nunzio, G. (2009). Using Scatterplots to Understand and Improve Probabilistic Models for Text Categorization and Retrieval. *Int. J. Approx. Reasoning*, *50*(7), 945–956.

Di Nunzio, G. (2014). A New Decision to Take for Cost-Sensitive Naïve Bayes Classifiers. *Information Processing and Management*, *50*(5), 653–674. doi:10.1016/j.ipm.2014.04.008

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience.

Duval, E. & Verbert, K. (2012). Learning Analytics. *E-Learning and Education (ELEED)*, *8*(1). Retrieved from http://nbn-resolving.de/urn:nbn:de:0009-5-33367

Elkan, C. (2001). The Foundations of Cost-sensitive Learning. In *Proceedings of the 17th international joint conference on artificial intelligence - volume 2* (pp. 973–978). IJCAI’01. Seattle, WA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1642194.1642224

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (third edition). Morgan Kaufmann.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer. Retrieved from http://books.google.it/books?id=tVIjmNS3Ob8C

Hiemstra, D. (2009). Probability Smoothing. In L. LIU & M. ÃZSU (Eds.), *Encyclopedia of database systems* (pp. 2169–2170). Springer US. doi:10.1007/978-0-387-39940-9_936

Keim, D. A., Kohlhammer, J., Ellis, G., & Mansmann, F. (Eds.). (2010, November). *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics. Retrieved from http://www.vismaster.eu/book/

Kruschke, J. K. (2014). *Doing Bayesian Data Analysis. A Tutorial with R, JAGS, and Stan* (2nd Edition). Academic Press, Inc.

Mladenic, D. & Grobelnik, M. (1999). Feature Selection for Unbalanced Class Distribution and Naive Bayes. In *Proceedings of the Sixteenth International Conference on*

Machine Learning (pp. 258–267). ICML ’99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=645528.657649

Ramaswami, M. & Bhaskaran, R. (2009). A Study on Feature Selection Techniques in Educational Data Mining. *CoRR*, *abs/0912.3924*. Retrieved from http://arxiv.org/abs/0912.3924

Rocha, A. & Goldenstein, S. K. (2014). Multiclass From Binary: Expanding One-Versus-All, One-Versus-One and ECOC-Based Approaches. *IEEE Trans. Neural Netw. Learning Syst. 25*(2), 289–302. doi:10.1109/TNNLS.2013.2274735

Romero, C. & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *40*(6), 601–618. doi:10.1109/TSMCC.2010.2053532

Romero, C. & Ventura, S. (2013). Data mining in education. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, *3*(1), 12–27. doi:10.1002/widm.1075

Sebastiani, F. (2002, March). Machine Learning in Automated Text Categorization. *ACM Comput. Surv. 34*(1), 1–47. doi:10.1145/505282.505283

Siemens, G. & Baker, R. (2012). Learning Analytics and Educational Data Mining: Towards Communication and Collaboration. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 252–254). LAK ’12. Vancouver, British Columbia, Canada: ACM. doi:10.1145/2330601.2330661

Singh, R. & Raj, B. (2004). Classification in Likelihood Spaces. *Technometrics*, *46*(3), 318–329. doi:10.1198/004017004000000347. eprint: http://www.tandfonline.com/doi/pdf/10.1198/004017004000000347

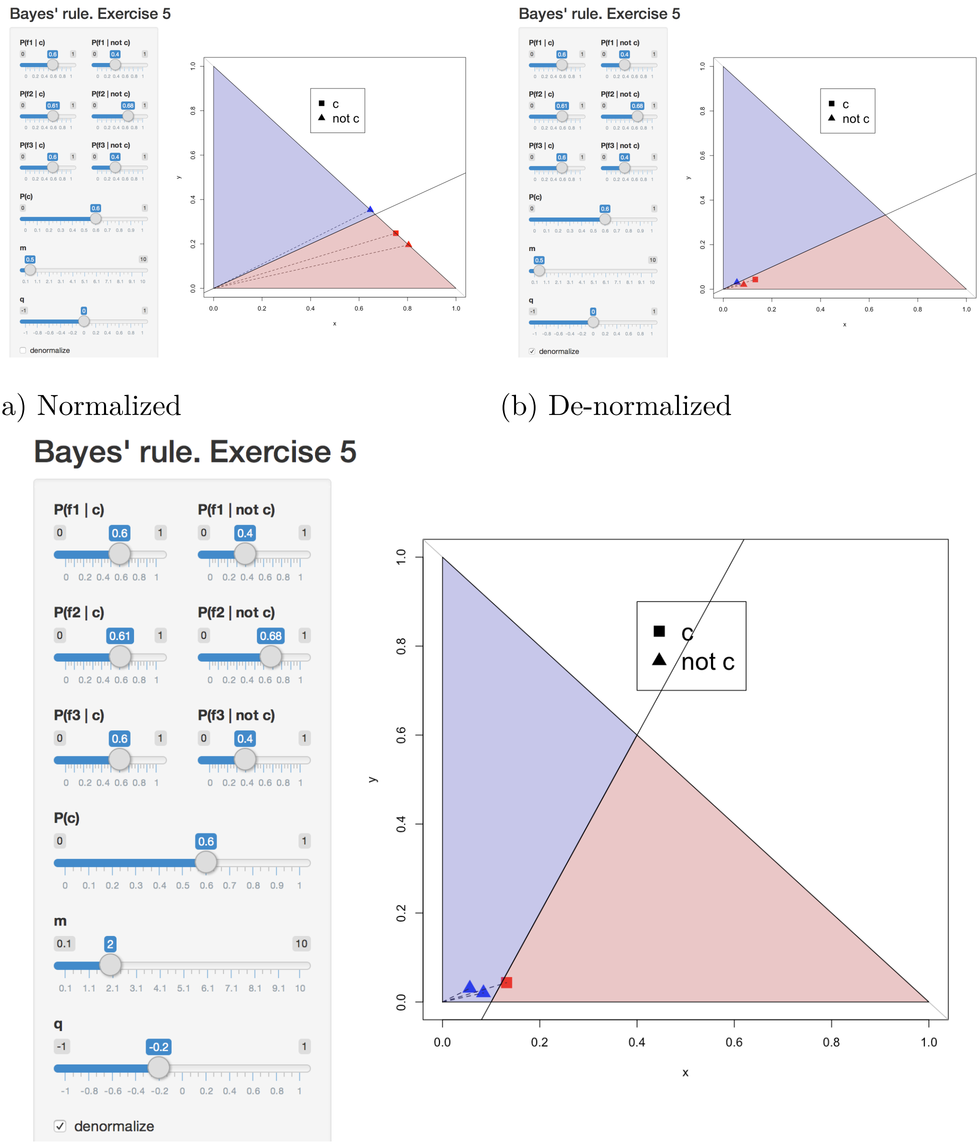
U.S. Department of Education. (2012, October). Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics. https://tech.ed.gov/learning-analytics/.

Valiente, J. A. R., Merino, P. J. M., Leony, D., & Kloos, C. D. (2015). ALAS-KA: A learning analytics extension for better understanding the learning process in the khan academy platform. *Computers in Human Behavior*, *47*, 139–148. doi:10.1016/j.chb.2014.07.002

van de Sande, B. (2013). Properties of the Bayesian Knowledge Tracing Model. *Journal of Educational Data Mining (JEDM)*, *5*(2), 1–10.

Ware, M., Frank, E., Holmes, G., Hall, M., & Witten, I. H. (2002, March). Interactive Machine Learning: Letting Users Build Classifiers. *Int. J. Hum.-Comput. Stud. 56*(3), 281–292. Retrieved from http://dl.acm.org/citation.cfm?id=514412.514417

Xing, W. & Goggins, S. P. (2015, March). Learning Analytics in Outer Space: A Hidden Naïve Bayes Model for Automatic Student Off-Task Behavior Detection. In *Proceedings of the fifth international conference on learning analytics and knowledge, LAK '15* (pp. 176–183). Poughkeepsie, NY, USA. doi:10.1145/2723576.2723602



Bayes' rule. Exercise 5

P(f1 | c) 0.6

P(f1 | not c) 0.4

P(f2 | c) 0.61

P(f2 | not c) 0.68

P(f3 | c) 0.6

P(f3 | not c) 0.4

P(c) 0.6

m 0.5

q 0

□ denormalize

(a) Normalized

Bayes' rule. Exercise 5

P(f1 | c) 0.6

P(f1 | not c) 0.4

P(f2 | c) 0.61

P(f2 | not c) 0.68

P(f3 | c) 0.6

P(f3 | not c) 0.4

P(c) 0.6

m 0.5

q 0

☑ denormalize

(b) De-normalized

Bayes' rule. Exercise 5

P(f1 | c) 0.6

P(f1 | not c) 0.4

P(f2 | c) 0.61

P(f2 | not c) 0.68

P(f3 | c) 0.6

P(f3 | not c) 0.4

P(c) 0.6

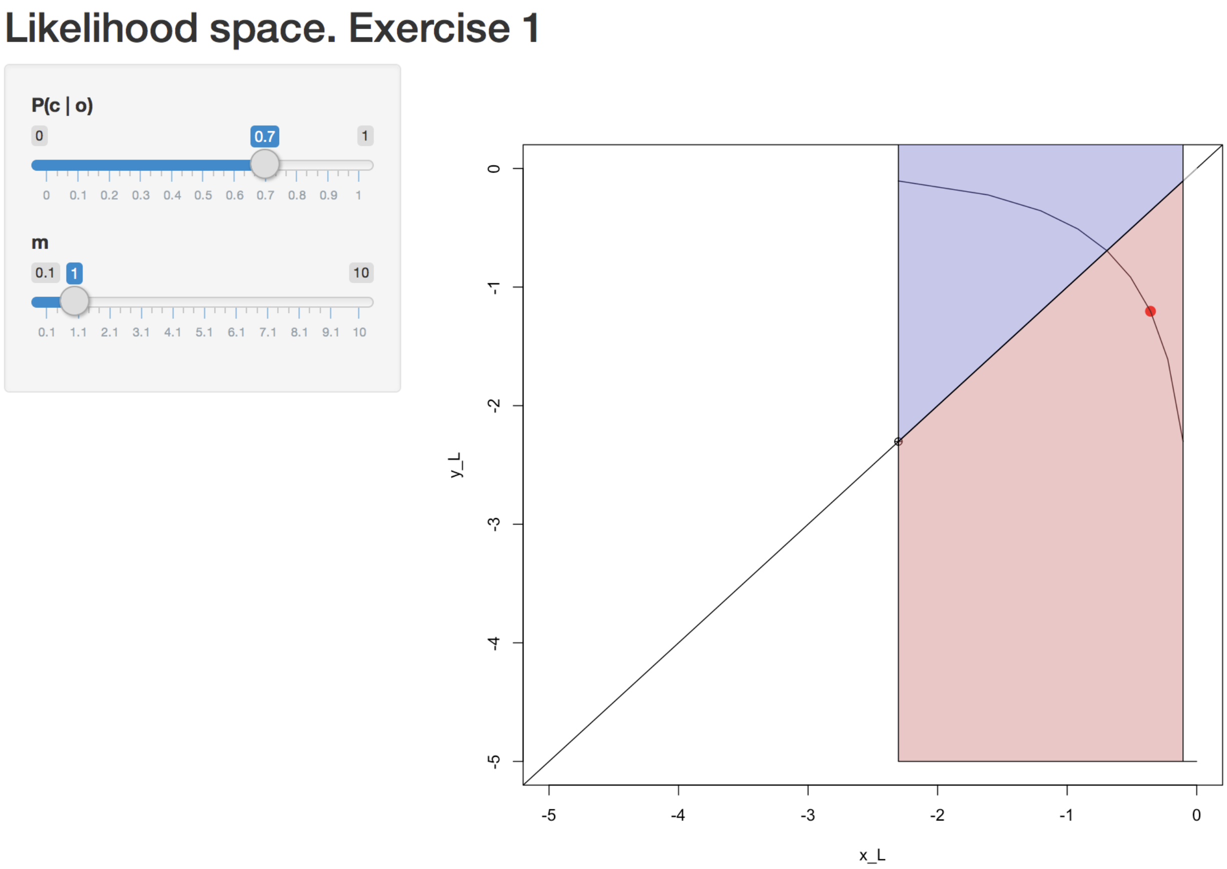
m 2

q -0.2

☑ denormalize

(c) De-normalized and decision line $y = mx + q$

Figure 6. The advantage of a de-normalization and decision line $y = mx + q$ is evident when two non-linearly separable classes become linearly separable.



Likelihood space. Exercise 1

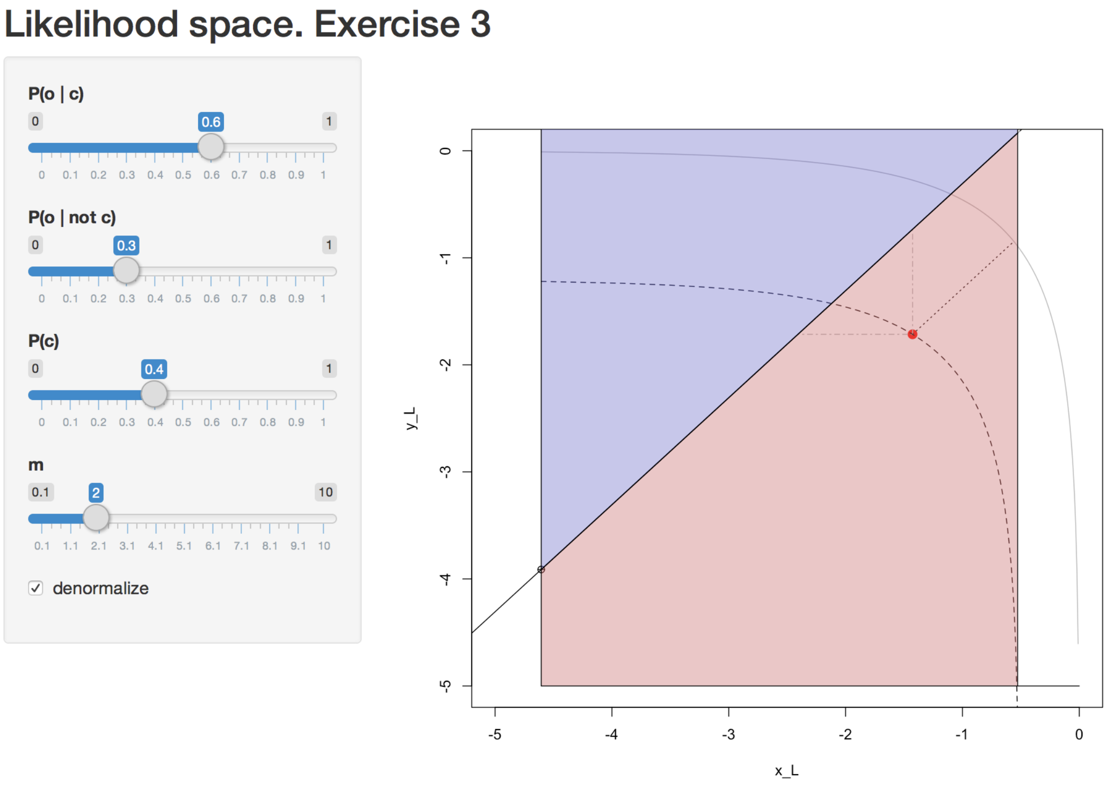
P(c | o)

0.7

m

1

Figure 7. In this example, we show the logarithm coordinates of the likelihood space.



Likelihood space. Exercise 3

P(o | c): 0.6

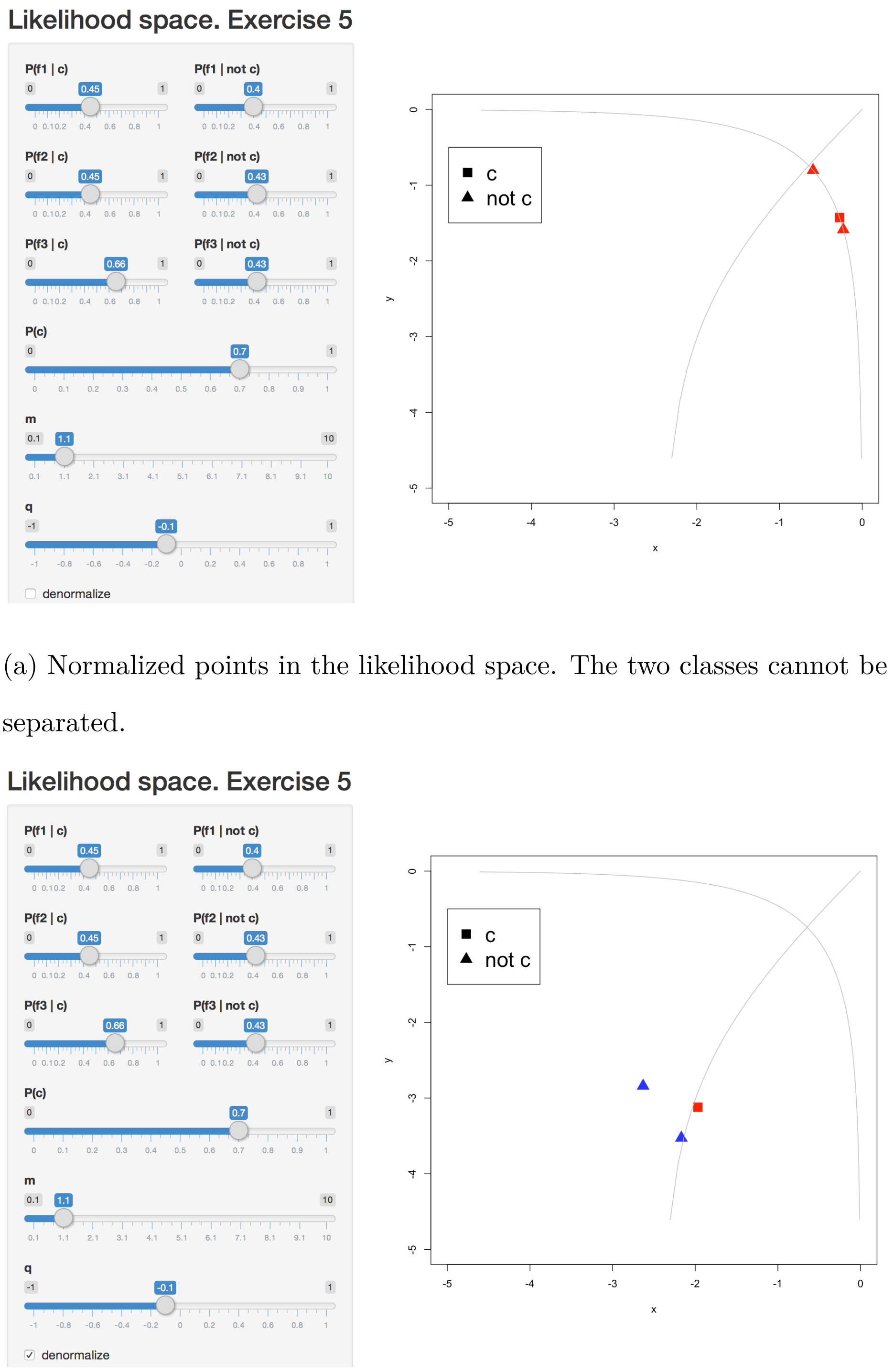
P(o | not c): 0.3

P(c): 0.4

m: 2

☑ denormalize

Figure 8. De-normalized point in the likelihood space.



Likelihood space. Exercise 5

P(f1 | c) 0.45

P(f1 | not c) 0.4

P(f2 | c) 0.45

P(f2 | not c) 0.43

P(f3 | c) 0.66

P(f3 | not c) 0.43

P(c) 0.7

m 1.1

q -0.1

□ denormalize

(a) Normalized points in the likelihood space. The two classes cannot be separated.

Likelihood space. Exercise 5

P(f1 | c) 0.45

P(f1 | not c) 0.4

P(f2 | c) 0.45

P(f2 | not c) 0.43

P(f3 | c) 0.66

P(f3 | not c) 0.43

P(c) 0.7

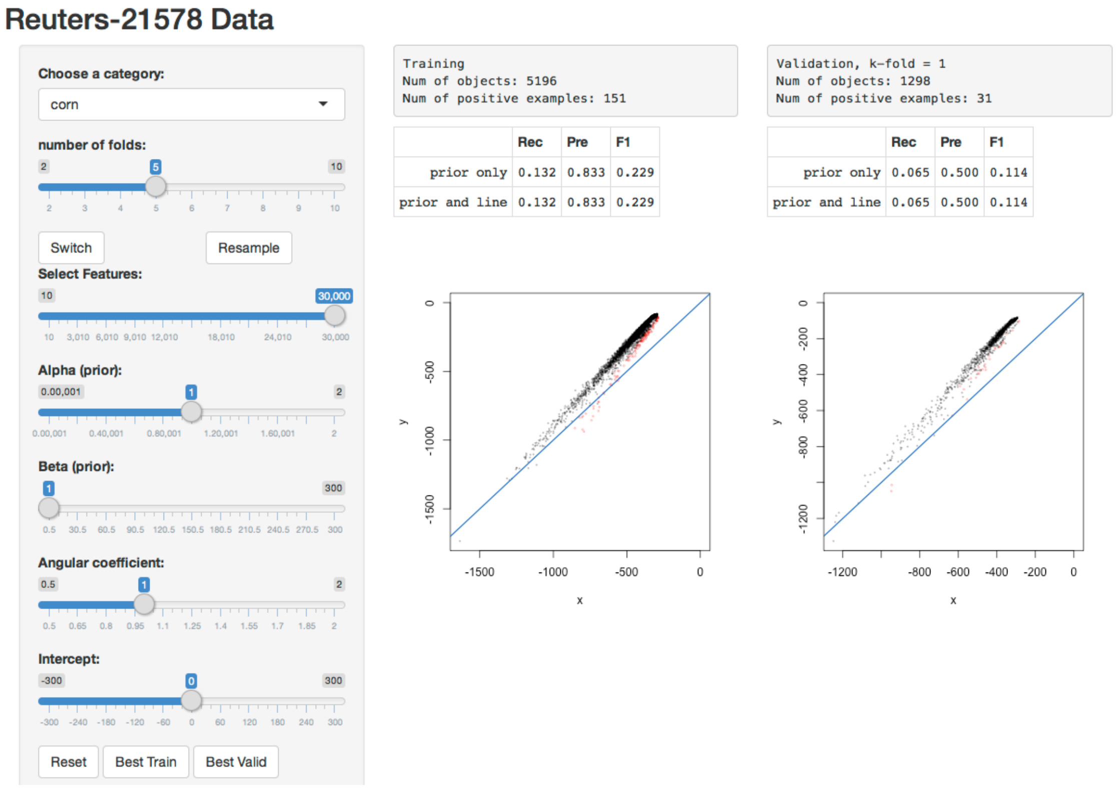
m 1.1

q -0.1

☑ denormalize

(b) De-normalized point in the likelihood space. The two classes are now separable with the decision function $\log(mx + q)$.

Figure 9. Separating two classes in likelihood spaces.



Reuters-21578 Data

Choose a category: corn

number of folds: 5

Select Features: 30,000

Alpha (prior): 1

Beta (prior): 1

Angular coefficient: 1

Intercept: 0

```
Training
Num of objects: 5196
Num of positive examples: 151
```

	Rec	Pre	F1
prior only	0.132	0.833	0.229
prior and line	0.132	0.833	0.229

```
Validation, k-fold = 1
Num of objects: 1298
Num of positive examples: 31
```

	Rec	Pre	F1
prior only	0.065	0.500	0.114
prior and line	0.065	0.500	0.114

Figure 10. Interactive Text categorization. Default values of a multivariate Bernoulli NB classifier on the Reuters-21578 dataset.