Interactive Text Categorisation: The Geometry of Likelihood Spaces

Giorgio Maria Di Nunzio

Abstract In this chapter we present a two-dimensional representation of probabilities called Likelihood spaces. In particular, we show the geometrical properties of Bayes' rule when projected into this two-dimensional space and extend this concept to Naïve Bayes classifiers. We apply this geometrical interpretation to a real machine learning problem of text categorisation and present a Web application that implements all the concepts on a standard text categorisation benchmark.

1 Introduction

Classification is the task of learning a function that assigns a new unseen object to one or more predefined classes based on the features of the object [Tan et al(2005)Tan, Steinbach, and Kumar, Chapter 4]. Among the many different approaches presented in the literature, Naïve Bayes (NB) classifiers have been widely recognised as a good trade-off between efficiency and efficacy since they are easy to train and achieve satisfactory results [Mitchell(1997)]. A NB classifier is a type of probabilistic classifier that uses Bayes' rule to predict the class of the unknown object, and is based on the simplifying assumption that all the features of the object are conditionally independent given the class. Despite being comparable to other learning methods, these classifiers are rarely among the top performers when trained with default parameters [Caruana and Niculescu-Mizil(2006)]. Indeed, the optimisation of the parameters of NB classifiers is often not adequate, if not missing at all. The usual approach is to set default smoothing constants to avoid arithmetic anomalies given by zero probabilities [Yuan et al(2012)Yuan, Cong, and Thalmann]. Moreover, a probabilistic classifier could be greatly improved by taking into account misclas-

Giorgio Maria Di Nunzio
Dep. of Information Engineering, University of Padua, Via Gradenigo 6/a 35131 Padova Italy, e-mail: giorgiomaria.dinunzio@unipd.it

sification costs [Elkan(2001)]. The choice of these costs is not trivial and, as for the case of probability smoothing, default costs are used.

By involving users directly in the process of building a probabilistic model, as suggested by [Ankerst et al(2000a)Ankerst, Ester, and Kriegel], one can obtain a twofold result: first, the pattern recognition capabilities of the human can be used to increase the effectiveness of the classifier construction and understand why some parameters work better than others; second, visualisation of the model can be used to teach non-experts how probabilistic models work and improve the overall effectiveness of the classification task. Interactive machine learning is a relatively new area of machine learning where model updates are faster and more focused with respect to classical machine learning algorithms; moreover, the magnitude of the update is small; hence, the model does not change drastically with a single update. As a result, even non-expert users can solve machine learning problems through low-cost trial and error or focused experimentation with inputs and outputs. In this respect, the importance of the design of proper user interfaces for the interaction with machine learning models is crucial. Recently, an approach named "Explanatory Debugging" has been described and tested to help end users build useful mental models of a machine learning system while simultaneously allowing them to explain corrections back to the system [Kulesza et al(2015)Kulesza, Burnett, Wong, and Stumpf] . The authors found a significant correlation between how participants understood how the learning system operated and the performance of participants' classifiers.

Based on the idea of Likelihood Spaces [Singh and Raj(2004)], we present the geometric properties of the two-dimensional representation of probabilities [Di Nunzio(2009), Di Nunzio(2014a)] which allows us to provide an adequate data and knowledge visualisation for understanding how parameter optimisation and cost sensitive learning affect the performance of probabilistic classifiers in a real machine learning setting. We apply this geometrical interpretation to the problem of text categorisation [Sebastiani(2002)], in particular to a standard collection of newswires, the Reuters-21578 collection. [1]

The main objectives of this chapter are:

- A geometrical definition of the Bayes' rule and a discussion on the implications of the normalisation of posterior probabilities;
- An alternative derivation of the likelihood space from the definition of the logit function;
- A description of the link between Bayesian Decision Theory and Likelihood spaces;
- 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.

This chapter is organized as follows: in Section 2, we describe the mathematical background behind the idea of the two dimensional representation. In Section 3, we present the details of the likelihood space applied to the NB classifier, in particular

[1] `http://www.daviddlewis.com/resources/testcollections/`

the multivariate Bernoulli model. Section 4 is dedicated to the interactive text categorization application on a real machine learning problem. In Section 5, we give our final remarks and discuss future works and open research questions.

1.1 Related Works

The term "interactive machine learning" was probably coined around the very end of the 1990s. A work that paved the way for this research area was a paper on interactive decision tree construction by Ankerst et alii [Ankerst et al(1999)Ankerst, Elsen, Ester, and Kriegel]. The same authors also redefined the paradigm that "the user is the supervisor" in this cooperation between humans and machine learning algorithms, that is the system supports the user and the user always has the final decision [Ankerst et al(2000b)Ankerst, Ester, and Kriegel]. In the same years, Ware et alii demonstrated that even users who are not domain experts can often construct good classifiers using a simple two-dimensional visual interface, without any help from a learning algorithm [Ware et al(2002)Ware, Frank, Holmes, Hall, and Witten]. Ben Shneiderman (author of the "eight golden rules for user interfaces" [Shneiderman(1997)]) gives his impressions on the importance of the effective combination of information visualisation approaches and data mining algorithms in [Shneiderman(2002)]. The first paper that used "interactive machine learning" in the title was by Fails and Olsen [Fails and Olsen(2003)] in which the authors describe the difference between a classical and an interactive machine learning approach and show an interactive feature selection tool for image recognition. From the point of view of Machine Learning/Artificial Intelligence, an excellent survey on the methods and approaches used in the last fifteen years has been presented by Amershi et alii [Amershi et al(2014)Amershi, Cakmak, Knox, and Kulesza].

Information Visualisation is an important part of the research area of interactive machine learning, in particular for the parts relative to the design of appropriate user interfaces and the possible visualisation choices for classification tasks. For example, in [Behrisch et al(2014)Behrisch, Korkmaz, Shao, and Schreck], the authors present a framework for a feedback-driven view exploration, inspired by relevance feedback approaches used in Information Retrieval, that makes the exploration of large multidimensional datasets possible by means of visual classifiers. Although we focus less on this part in this paper, we suggest to refer to [Di Nunzio(2014b)] for a survey on visual classification approaches and to [Kucher and Kerren(2014)] for a survey on text visualisation techniques. [2]

[2] `http://textvis.lnu.se`

2 Mathematical Background

We suppose to work with a set of n classes $C = \{c_1, ..., c_i, ..., c_n\}$, and that an object can be assigned to (and may actually have) more than one class; this is also known as the problem of overlapping categories. Instead of building one single multi-class classifier, we split this multi-class categorisation into n binary problems; therefore, we have n binary classifiers [Sebastiani(2002)]. A binary classification problem is a special case of single-labels classification task in which each object belongs to one category or its complement. The usual notation to indicate these two classes is: c_i for the 'positive' class and $\bar{c}_i$ for the 'negative' class (we drop the index i and use c and $\bar{c}$ as long as there is no risk of misinterpreting the meaning).

In this first part, we start building a probabilistic classifier which, given an object o and a category $c \in C$, classifies o in 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 is greater than the probability of its complement $\bar{c}$ given the object o.

2.1 The Geometry of Bayes' Rule

Bayes' rule gives a simple yet powerful link between prior and posterior probabilities of events. For example, assume that we have two classes c and $\bar{c}$ and we want to classify objects according to some measurable features. The probability that an object o belongs to c, $P(c|o)$, can be computed in the following way: [3]

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

Bayes' rule tells how, by starting from a prior probability on the category c, $P(c)$, we can update our belief on that category based on the likelihood of the object, $P(o|c)$, and obtain the so-called posterior probability $P(c|o)$. $P(o)$ is the probability of the object o, also known as the evidence of the data. The probability of the complementary category $\bar{c}$ is computed accordingly:

$$P(\bar{c}|o) = \frac{P(o|\bar{c})P(\bar{c})}{P(o)} \tag{3}$$

[3] We are intentionally simplifying the notation in order to have a cleaner description. In particular, when we write $P(c|o)$, we actually mean $P(C = c|O = o)$, where C and O are two random variables, and c and o two possible values, respectively.

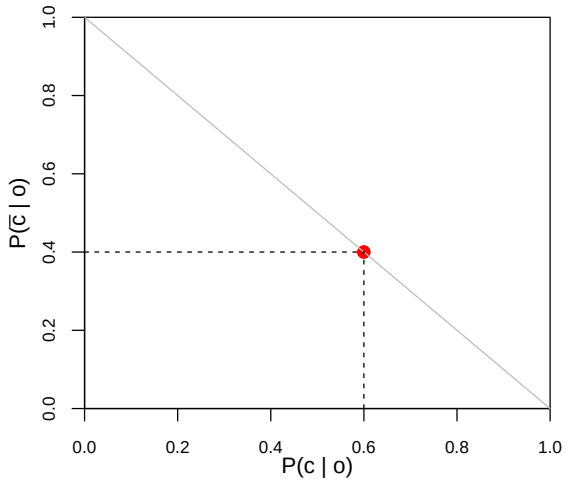

Fig. 1 Bayes' rule on a two-dimensional space. The probability of one class is complementary to the other, $P(\bar{c}|o) = 1 - P(c|o)$.

In the two-dimensional view of probabilities, we can imagine the posterior probabilities as the two coordinates of the object o in a Cartesian space, where $x = P(c|o)$ and $y = P(\bar{c}|o)$. Since the two classes are complementary, the two conditional probabilities sum to one, therefore:

$$P(\bar{c}|o) = 1 - P(c|o) \text{ , or} \tag{4}$$

$$y = 1 - x \tag{5}$$

which means that the point with coordinates (x, y) lies on the segment with endpoints $(1,0), (0,1)$ in the two dimensional space, as shown in Figure 1. When we want to classify the object, we compare the two probabilities as already shown in Equation 1. When we use Bayes' rule to calculate the posterior probabilities, we obtain:

$$\frac{P(o|c)P(c)}{P(o)} > \frac{P(o|\bar{c})P(\bar{c})}{P(o)} \tag{6}$$

It can be immediately seen that we assign o to class c when the probability $P(c|o)$ is greater than 0.5. Since $P(o)$ appears in both sides of the inequality, we can cancel it without changing the result of the classification:

$$P(o|c)P(c) > P(o|\bar{c})P(\bar{c}) \tag{7}$$

remembering that $P(o|c)P(c) \neq 1 - P(o|\bar{c})P(\bar{c})$ since we removed the normalisation factor. An alternative way to cancel $P(o)$ is considering the problem of classification in terms of the odds of the probability $P(c|o)$:

$$P(c|o) > P(\bar{c}|o) \tag{8}$$

$$\frac{P(c|o)}{P(\bar{c}|o)} > 1 \tag{9}$$

$$\frac{P(o|c)P(c)}{P(o|\bar{c})P(\bar{c})} > 1 \tag{10}$$

$$P(o|c)P(c) > P(o|\bar{c})P(\bar{c}) \tag{11}$$

In geometrical terms, the new coordinates x' and y' of the point of the object o are:

$$x' = xP(o) = P(o|c)P(c) \tag{12}$$

$$y' = yP(o) = P(o|\bar{c})P(\bar{c}) \tag{13}$$

The new coordinates are the old ones multiplied by $P(o)$ which means that we are actually 'pushing' the points towards the origin of the axis along the segment with endpoints $(0,0), (P(c|o), P(\bar{c}|o))$ since both coordinates are multiplied by the same positive number between 0 and 1, as shown in Figure 2.

Equation 11 can also be interpreted as a decision line with equation $y' = x'$. A more general classification line takes into account an angular coefficient m

$$mx' > y' \tag{14}$$

This non-negative parameter m comes from the introduction of misclassification costs of a Bayesian Decision Theory approach (see Section 2.3). Intuitively, when $m = 1$ we count every misclassification (false positives or false negatives) equally. If $m > 1$, we give more importance to the positive class and we are willing to accept more objects in this class; if $m < 1$, we increase the possibility that a point is above the line and classified under the negative category. An alternative, but equivalent, way of looking at this problem is to compare the value of the odds with a threshold k [Crestani et al(1998)Crestani, Lalmas, Van Rijsbergen, and Campbell]:

$$\frac{x'}{y'} > \frac{1}{m} \tag{15}$$

$$\frac{P(o|c)P(c)}{P(o|\bar{c})P(\bar{c})} > k \tag{16}$$

where k (inversely proportional to m in this formulation) can be set to optimise classification, and it is usually tuned to compensate for the unbalanced classes situation, that is when one of the two classes is much more frequent than the other [Mladenic and Grobelnik(1999)]. This is often the case for any multi-class problem, since the complementary category $\bar{c}$ is about $n-1$ times bigger than c. This is also the case for real two-class categorisation problems, like spam classification, where the dif-

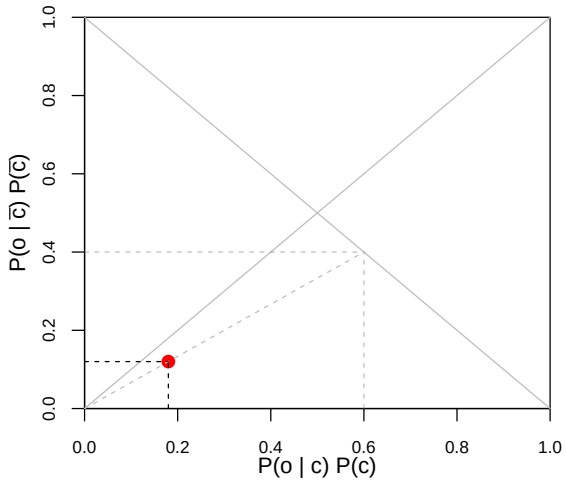

Fig. 2 Bayes' rule without normalisation. The point moves from the segment with endpoints $(0,1),(1,0)$ towards the origin. In this example, $P(o) = 0.3$. The ratio of the coordinates remain the same as well as the relative position with respect to the decision line with angular coefficient $m = 1$ (bisecting line of the first quadrant).

ference in proportion of the number of objects in the two classes 'spam' and 'ham' is very large. We can incorporate this disproportion between the two classes in the angular coefficient m of the two-dimensional space in the following way:

$$mx' > y' \tag{17}$$

$$mP(o|c)P(c) > P(o|\bar{c})P(\bar{c}) \tag{18}$$

$$m\frac{P(c)}{P(\bar{c})}P(o|c) > P(o|\bar{c}) \tag{19}$$

$$m'x'' > y'' \tag{20}$$

where $m' = m\frac{P(c)}{P(\bar{c})}$ is the new angular coefficient of the decision line $y'' = m'x''$, and $x'' = P(o|c)$ and $y'' = P(o|\bar{c})$. At this point we have defined the coordinates of an object in terms of the two likelihood functions $P(o|c)$ and $P(o|\bar{c})$ as shown in Figure 3.

All the alternatives presented so far are equivalent in terms of classification decisions. There are two connections with two relevant works in the literature that we want to stress: one with the Neyman-Pearson approach [Neyman and Pearson(1933)], and the other with the work of Pazzani and colleagues on the opti-

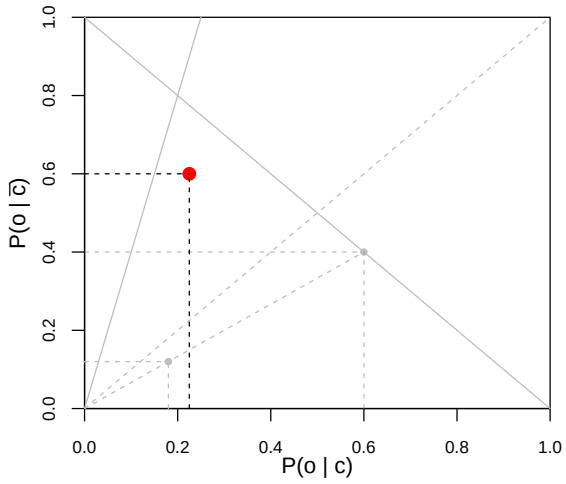

Fig. 3 Data space formed by the coordinates $P(o|c)$ and $P(o|\bar{c})$. This is an example of an unbalanced class situation where the prior $P(c) = 0.7$ is so high that the object is classified under c (in accordance with the earlier examples) despite the likelihood of the object of the negative class is almost three times the one of the positive class. In this example, $m = 1$ and $m' = \frac{P(c)}{P(\bar{c})}$. If we set $m = \frac{P(\bar{c})}{P(c)}$, we would get $m' = 1$ and rebalance the proportion of classes (and change the classification decision). The points of the previous figures are shown in light grey for comparison.

mality of NB classifiers [Domingos and Pazzani(1997), Webb and Pazzani(1998)]. The Neyman-Pearson lemma states that the likelihood ratio test defines the most powerful region of acceptance, which is exactly what we have in Equation 20:

$$\frac{P(o|c)}{P(o|\bar{c})} > M \tag{21}$$

where M is a threshold that defines the region of acceptance. In the optimality of NB classifiers, the authors find an adjustment of the probabilities of the classes $P(c)$ and $P(\bar{c})$ which is again exactly the same idea since we are actually changing the angular coefficient m'.

2.2 Bayes' Rule on Likelihood Space

So far, we have described the two-dimensional representation of the Bayes' rule in the so-called 'data space' which is the space in which the original data resides. The likelihood space, however, is the space formed by the log-likelihood probabilities [Singh and Raj(2004)]. The likelihood space can be derived directly by applying the logs of Equation 20. In this section, we present an alternative way, which is different from the original paper, to obtain the likelihood space which starts from the classification decision given by the log-odds, or logit function, compared to the logarithm of the threshold k:

$$\log\left(\frac{P(c|o)}{P(\bar{c}|o)}\right) > \log(k) \tag{22}$$

$$\log\left(\frac{P(o|c)P(c)}{P(o|\bar{c})P(\bar{c})}\right) > \log(k) \tag{23}$$

$$\log\left(\frac{P(o|c)}{P(o|\bar{c})}\right) + \log\left(\frac{P(c)}{P(\bar{c})}\right) > \log(k) \tag{24}$$

$$\log\left(P(o|c)\right) + \log\left(\frac{P(c)}{P(\bar{c})}\frac{1}{k}\right) > \log\left(P(o|\bar{c})\right) \tag{25}$$

$$x_L + q_L > y_L \tag{26}$$

The likelihood space coordinates of an object o, $x_L = \log(x'')$ and $y_L = \log(y'')$, are the logarithms of the coordinates of the data space. An interesting relation between the data space and the likelihood space is that, while in the data space we 'rotate' the decision line around the origin of the axis ($y'' = m'x''$), the same decision line in the likelihood space correspond to a parallel line to the bisecting line of the first and third quadrant $y_L = x_L + q_L$ where $q_L = \log(m')$ is the intercept of this line. In Figure 4, we show an example of the likelihood space relative to the point of Figure 3.

2.3 Bayesian Decision Theory on Likelihood Spaces

In Bayesian decision theory, the objective is to quantify the trade-off between various classification decisions using probabilities and the costs that accompany such decisions [Duda et al(2000)Duda, Hart, and Stork, Chapter 2]. Whenever we have an object to classify, if we take the decision to classify it under c, we are actually "taking a risk" because we may choose the wrong category. In this framework, the classification of an object becomes the problem of choosing the 'less risky' category; for a binary classification problem, the Bayes decision rule corresponds to selecting the action for which the risk is minimum:

$$R(c|o) < R(\bar{c}|o) \tag{27}$$

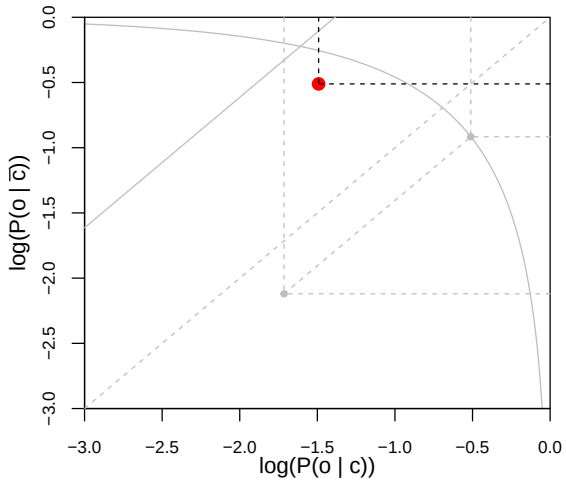

Fig. 4 Bayes' rule on likelihood space. The red point corresponds to the one shown in Figure 3. Note that the decision line (solid grey line) is above the red point as expected. The decision line moves parallel to the bisecting line of the third quadrant. In light grey, the points relative to Figures 1 and 2. Non normalised points move parallel to the bisecting lines and towards minus infinity, instead of going towards the origin. The segment with endpoints $(0,1),(1,0)$ becomes a logarithmic curve in the likelihood space.

$R(c|o)$ and $R(\bar{c}|o)$ are the conditional risks defined as:

$$R(c|o) = \lambda(c|c)P(c|o) + \lambda(c|\bar{c})P(\bar{c}|o) \tag{28}$$

$$R(\bar{c}|o) = \lambda(\bar{c}|c)P(c|o) + \lambda(\bar{c}|\bar{c})P(\bar{c}|o) \tag{29}$$

where $\lambda(\cdot|\cdot)$ is the loss function of an action given the true classification. For example, $\lambda(c|\bar{c})$ quantifies the loss in taking the decision c when the 'true' decision is $\bar{c}$. The new classification decision becomes:

$$\lambda(c|c)P(c|o) + \lambda(c|\bar{c})P(\bar{c}|o) < \lambda(\bar{c}|c)P(c|o) + \lambda(\bar{c}|\bar{c})P(\bar{c}|o) \tag{30}$$

We can group common terms and obtain:

$$[\lambda(c|\bar{c}) - \lambda(\bar{c}|\bar{c})]P(\bar{c}|o) < [\lambda(\bar{c}|c) - \lambda(c|c)]P(c|o) \tag{31}$$

$$P(\bar{c}|o) < \frac{[\lambda(\bar{c}|c) - \lambda(c|c)]}{[\lambda(c|\bar{c}) - \lambda(\bar{c}|\bar{c})]}P(c|o) \tag{32}$$

$$P(o|\bar{c})P(\bar{c}) < \frac{[\lambda(\bar{c}|c) - \lambda(c|c)]}{[\lambda(c|\bar{c}) - \lambda(\bar{c}|\bar{c})]}P(o|c)P(c) \tag{33}$$

$$P(o|\bar{c}) < \frac{[\lambda(\bar{c}|c) - \lambda(c|c)]}{[\lambda(c|\bar{c}) - \lambda(\bar{c}|\bar{c})]}\frac{P(c)}{P(\bar{c})}P(o|c) \tag{34}$$

$$y'' < m'x'' \tag{35}$$

So the ratio of the costs can be interpreted as the angular coefficient m included in m' of Equation 20. When a zero-one loss function is used, we have $\lambda(c|c) = \lambda(\bar{c}|\bar{c}) = 0$ which means that we have no loss when we give the correct answer, and $\lambda(c|\bar{c}) = \lambda(\bar{c}|c) = 1$ which means that we have a cost equal to one every time we assign the object to the wrong category.

3 Naïve Bayes on Likelihood Space

In real case scenarios, projecting objects into likelihood spaces becomes a necessity since the conditional probabilities $P(o|c)$ and $P(o|\bar{c})$ rapidly go to zero. This problem becomes evident when we use a Naïve Bayes assumption. For example, if o is represented by a set of k features $F = \{f_1, \ldots, f_j, \ldots, f_k\}$, a Naïve Bayes approach allows us to factorize $P(o|c)$ as:

$$P(o|c) = \prod_{j=1}^{k} P(f_j|c) \tag{36}$$

where features are independent from each other given the class. 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. Instead, in likelihood spaces, the product becomes a sum of logarithms of probabilities:

$$\log(P(o|c)) = \log\left(\prod_{j=1}^{k} P(f_j|c)\right) = \sum_{j=1}^{k} \log(P(f_j|c)) \tag{37}$$

In the following section, we derive the mathematical formulation of a NB model that represents features with binary variables, known as multivariate Bernoulli NB model. In Figure 5, we show a screenshot of an interactive demo of this type of

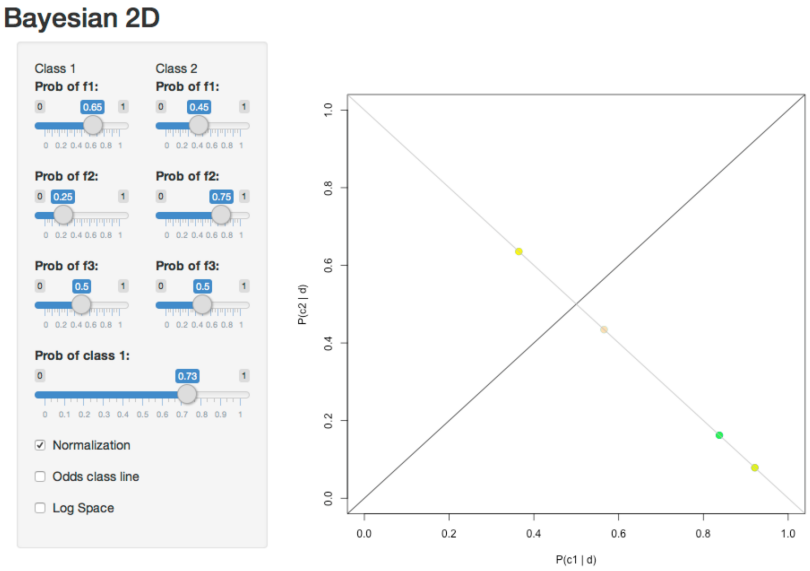

Fig. 5 An interactive demo to show how a multivariate Bernoulli NB model works on a two-dimensional space.

NB classifier. [4] The aim of this toy example is to show the geometric interpretation of this classifier rather than study the optimal parameters for classification. The user can change the conditional probability of each single feature (f_1, f_2, and f_3) and the prior probability of class c_1 (the positive class). The points represent the eight possible combinations (three binary features, hence $2^3 = 8$ objects); when the conditional probability of a feature given the positive class equals that of the negative class, some points overlap in the data space (because we are not able to use that feature to discriminate the objects of one class from the others). The selection widgets allow for choosing normalised probabilities and working in the likelihood space ('log space').

3.1 Multivariate Bernoulli NB model

In the multivariate Bernoulli NB model, an object is a binary vector over the space of features. Given a set of features F, each object o of the class c is represented as a vector of k Bernoulli random variables $o \equiv (f_1, ..., f_j, ..., f_k)$ such that:

$$f_j \sim \text{Bern}(\theta_{f_j|c}) \ . \tag{38}$$

[4] `http://gmdn.shinyapps.io/bayes2d/`

where $\theta_{f_j|c}$ is the parameter of the Bernoulli distribution for the j-th feature of class c. We can re-write the probability of an object by using the NB conditional independence assumption, this time by considering the parameter θ of the distribution:[5]

$$P(o|c;\theta) = \prod_{j=1}^{k} P(f_j|c;\theta) = \prod_{j=1}^{k} \theta_{f_j|c}^{h_j}(1-\theta_{f_j|c})^{1-h_j} = \prod_{j=1}^{k} \left(\frac{\theta_{f_j|c}}{1-\theta_{f_j|c}}\right)^{h_j}(1-\theta_{f_j|c}) \,, \tag{39}$$

where h_j is either 1 or 0 indicating whether feature f_j is present or absent in object o. When we project this probability into the likelihood space, we obtain:

$$\log(P(o|c;\theta)) = \sum_{j=1}^{k} h_j \log\left(\frac{\theta_{f_j|c}}{1-\theta_{f_j|c}}\right) + \sum_{j=1}^{k} \log(1-\theta_{f_j|c}) \,, \tag{40}$$

In terms of the likelihood projections, each object of class c has a coordinate composed by: i) a variable part, the first sum, that depends on the features that are present in the object, and ii) a fixed part, the second sum, that considers all the features F independently from the features that appear in the object. This second part is very important because, in many works, it is ignored (actually canceled) with the justification that it is a constant independent from the object and, therefore, it does not change the classification decision. This is true only if we do not fix q_L in advance but, on the contrary, we find the optimal parameter q_L of the decision line of Equation 26. In fact, once q_L is fixed, including or excluding the second sum in the computation of the coordinates would result in a different decision since the points would have different coordinates. The two solutions are equivalent 'only' when we choose an appropriate threshold:

$$\log\left(x''\right) + \log\left(m'\right) > \log(y'') \tag{41}$$

$$\log(x_1'') + \log(x_2'') + \log\left(m'\right) > \log(y_1'') + \log(y_2'') \tag{42}$$

$$\log(x_1'') + \log\left(\frac{m'x_2''}{y_2''}\right) > \log(y_1'') \tag{43}$$

where $x_1'' = \sum_{j=1}^{k} h_j \log\left(\frac{\theta_{f_j|c}}{1-\theta_{f_j|c}}\right)$ and $x_2'' = \sum_{j=1}^{k} \log(1-\theta_{f_j|c})$ (y_1'' and y_2'' are defined accordingly). For example if we set $m' = 1$ in Equation 41, then we must set $m' = y_2''/x_2''$ to obtain the same classification in Equation 43.

3.2 Probability smoothing

The parameter $\theta_{f|c}$ of each Bernoulli random variable can be estimated in different ways. A common solution is a Maximum Likelihood approach:

[5] We use the notation $P(o|c;\theta)$ to indicate the probability parametrised by θ.

$$\theta_{f|c} = \frac{n_{f,c}}{n_c} \tag{44}$$

where $n_{f,c}$ is the number of objects of category c in which feature f appears, and n_c is the number of objects of category c. However, this approach generates arithmetical anomalies; in particular, a probability equal to zero when the feature is absent in category c, $n_{f,c} = 0$ (or a probability equal to one when $n_{f,c} = n_c$ but it is less frequent). A zero in one of the features of the objects corresponds to a likelihood equal to zero (or a minus infinity in the log space). To avoid these arithmetical problems, smoothing is usually applied. For example, Laplacian smoothing or add-one smoothing:

$$\theta_{f|c} = \frac{n_{f,c} + 1}{n_c + 2} \tag{45}$$

In this chapter, instead of a Maximum Likelihood approach, we estimate the parameter $\theta_{f|c}$ by using a conjugate prior approach which, in this case, corresponds to finding a *beta* function with parameters α and β [Di Nunzio and Sordoni(2012)]:

$$beta_{f|c} = \theta_{f|c}^{\alpha-1}(1-\theta_{f|c})^{\beta-1} \ . \tag{46}$$

The result of this choice is that the estimate $\theta_{f|c}$ is now governed by the two hyperparameters α and β in the following way:

$$\theta_{f|c} = \frac{n_{f,c} + \alpha}{n_c + \alpha + \beta} \tag{47}$$

note that for $\alpha = \beta = 1$, we obtain the Laplacian smoothing. It is possible to optimise α and β for each feature, but in this work we choose to use the same parameters for all the features.

3.3 Decision Line in Likelihood Spaces

As suggested by the authors of the original paper of likelihood spaces [Singh and Raj(2004)], one advantage with working in likelihood spaces is that we can devise new strategies for classifying objects. In fact, if we do not limit ourselves to the Bayesian Decision Theory, we can find other linear or non-linear solutions that work much better in terms of classification. The first improvement would be to add a 'rotation' to the decision line in the likelihood space. The authors of the seminal paper discuss this problem and show that polynomial decision lines in the likelihood space can obtain a significant improvement in terms of classification accuracy. However, a polynomial line in the likelihood space corresponds to a complex curve in the data space. Suppose that we find a decision function of this type

$$y_L < m_L x_L + q_L \tag{48}$$

where y_L, x_L, q_L are the same as Equation 26 and m_L is the angular coefficient of the new decision line. This corresponds to:

$$e^{y_L} < e^{m_L x_L + q_L} \tag{49}$$

$$e^{\log(P(o|\bar{c}))} < e^{m_L \log(P(o|c)) + q_L} \tag{50}$$

$$P(o|\bar{c}) < P(o|c)^{m_L} e^{q_L} \tag{51}$$

which is a sort of exponential curve in the data space. Alternatively, it is also possibile to show that a rotation and a shift of the decision function in the data space corresponds to a non-linear curve in the likelihood space [Di Nunzio(2014a)]. However, it is not our main objective to discuss the possible extension of Bayesian Decision Theory in this chapter. However, we want to stress the fact that, for the interactive text categorisation problem, we use a decision line in the likelihood space like the one shown Equation 48 but this choice does not have an immediate interpretation in the data space in terms of Bayesian Decision Theory.

4 An Example of Interactive 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 optimise classification. In particular:

- we can change the estimates of the probability of the features by modifying the values α and β of the prior beta function.
- we can adjust the classification line by changing the intercept q_L and the angular coefficient m_L in the likelihood space;

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 minimise the error of the classifier[Duda et al(2000)Duda, Hart, and Stork, Chapter 9]. For this reason, we have developed an interactive application that allows users to see how the tuning of these parameters affects classification on a real text classification problem. [6]

The top 10 most frequent categories of the Reuters-21578 [7] corpus were chosen as a benchmark. In particular, we chose the 6,494 training documents. Table 1 shows the number of training documents for each category. Some text preprocessing was done: a first cleaning was done to remove all the punctuation marks and convert all the letters to lowercase. A stoplist of 571 words and contractions (that is, 're, don't, etc.) was used to remove the most frequent words of the English language. [8] Finally,

[6] http://gmdn.shinyapps.io/shinyK

[7] http://www.daviddlewis.com/resources/testcollections/

[8] http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/

Table 1 Number of training documents for each class of the Reuters-21578 collection

category	training
acq	1,650
corn	182
crude	391
earn	2,877
grain	434
interest	347
money-fx	539
ship	198
trade	369
wheat	212
total	6,494

the English Porter stemmer [9] was used as the only method to reduce the space of terms.

Standard classification measures are calculated for the k-fold cross validation and shown real time as parameters are tuned [Sokolova and Lapalme(2009)].

4.1 Description of the Interface

The main window is split into two parts: the sidebar on the left and the main panel on the right, as shown in Figure 6. On the left side, the user can interact with the classifier and see the results on the right in terms of both the accuracy of the classification and the visualisation.

4.1.1 Interaction

The user can interact with the classifier by adjusting and changing the values of the following widgets (we describe them from top to bottom, but the user can interact in any order):

1. The user chooses the category of documents to classify with a selection input menu.
2. The number of k-folds, between 2 and 10, to train and validate on are selected by a slider; the user can also switch from one k-fold to the other (for example, with five folds, the first fold is used for validation while the other four folds are used to train the classifier), or re-sample the folds (documents are randomly sampled to create a new k-fold cross validation) by using the two buttons below the slider of the number of folds.

[9] `http://www.tartarus.org/~martin/PorterStemmer/`

3. The number of features (terms) can be selected with a slider from 5 up to 30,000 features.
4. The parameters of the beta prior can be adjusted by the two sliders *Alpha*, from 10^{-5} to 2, and *Beta*, from 0.5 to 300.
5. The decision line can be adjusted with the two sliders *Angular coefficient*, values from 0.5 to 2, and *Intercept*, values in the range -300, 300.
6. The user can reset all the parameters to the default values, or go back to the best settings found for the training set or the validation set by using one of the three buttons.

4.1.2 Visualization

The main panel is divided into two columns: the first column shows the results on the training set, the second column the results on the validation set. Both columns contain the following information (from top to bottom):

1. The text box shows the total number of objects and the number of positive examples (red points, the documents of the chosen category). The box in the validation column also tells the user on what fold we are validating.
2. The table shows performance measures in terms of Recall, Precision and F1. The first row displays the performance of the classifier when only the parameter of the priors are used, while the second row gives the results when both the prior and the coefficient of the decision line are taken into account.
3. The two-dimensional plot shows in red the documents of the chosen class and in black all the other documents of the collection. The blue line changes according to the parameters *Angular coefficient* and *Intercept*, m and q respectively, while the green line (visible only when the previous parameters are not the default ones) remains fixed to the bisecting line of the third quadrant.

4.2 Example of usage

Figure 6 shows an example of one category '*corn*' that is quite unbalanced, since the number of positive examples of this category is around 180 and the total number of training examples is about 6,400. In order to recover this disproportion, we can change the value of the intercept of the decision line and increase it to 200. In this way, we get an almost perfect recall but the precision is low, as shown in Figure 7. This situation shows how the intervention of the loss function (which influences the shift of the line in the likelihood space) is good but not optimal. A rotation of the line can significantly improve the situation as shown in Figure 8.

This optimisation can continue iteratively by slightly changing the intercept and the angular coefficient. Additionally (or alternatively), the user can change the smoothing of the probabilities with the sliders alpha and beta. As surprising as it

Fig. 6 Interactive Text Categorisation. Default values of a multivariate Bernoulli NB classifier on the Reuters-21578 dataset.

may seem, for small values of alpha and high values of beta, the points in the likelihood space change their distribution and 'move' around the zero-one loss decision function (bisecting line third quadrant, green line). This particular behaviour can be explained by the fact that for $\alpha = \beta = 1$ we are actually giving as input a uniform distributed prior which is very unlikely in real situations; in other words, we are saying that any value for the parameter $\theta_{f|c}$ is equally probable. Instead, it is much more likely to observe a very small value close to zero. This is expressed by a beta function whose parameters have the values suggested in the Figures.

This incremental process, as the interactive machine learning approach suggests, can significantly improve the initial results of the classifier. With this interactive application, we can also show how overfitting may generate very poor classifiers. This situation is shown in Figure 10, where we set the alpha and beta values to their extremes and slightly adjusted the intercept and the angular coefficient to obtain an almost perfect score on the training data ($F_1 = 0.956$). With these parameters, the performance on the validation set is very low. Compared to Figure 9, the F_1 score decreased from 0.7 to 0.4.

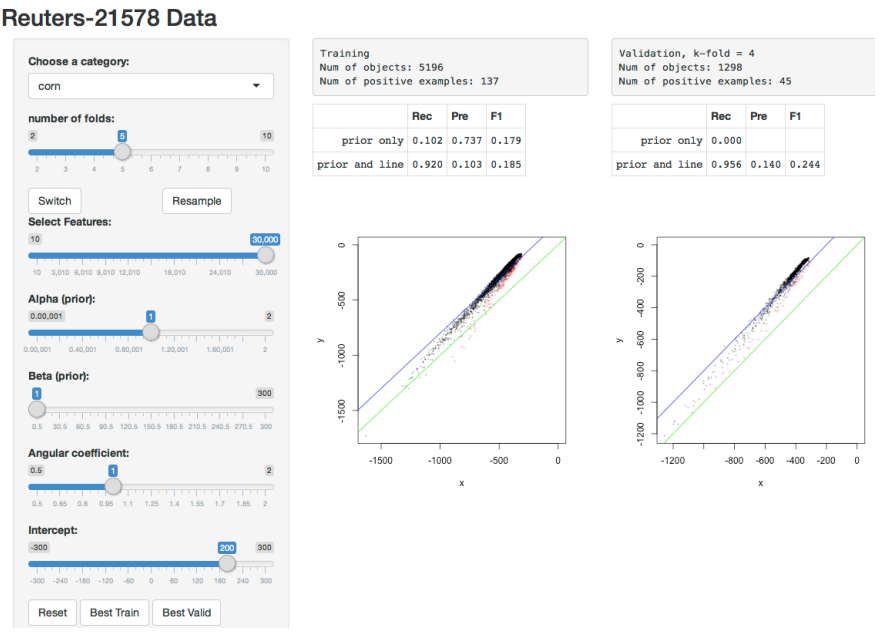

Fig. 7 Interactive Text Categorisation. Increase the value of intercept to recover the disproportion of the two classes.

5 Final Remarks and Future Works

In this chapter we have presented a geometrical interpretation of likelihood spaces and an interactive text categorisation problem that makes use of this interpretation. We have explained the possible relations that exist between likelihood spaces and Bayesian Decision Theory; moreover, we have derived the same interpretation of the two-dimensional logarithmic space from the definition of classification in terms of the logit function. The interactive application shows, in a real machine learning setting, how human pattern recognition capabilities can immediately steer the learning algorithm towards one possible solution.

The importance of the visualisation approach becomes more evident when the result is used as input for the optimisation of a classifier. Theoretically, we could find the solution found with the interactive approach (if not a better one) by means of a classical full-automatic machine learning approach that searches for the best combination of parameters. The problem is that the space of the vector of parameters is huge. Although a reduction of the space can be obtained with a correct interpretation of the problem in geometrical terms [Di Nunzio(2009), Di Nunzio and Micarelli(2004)], the interactive approach can be crucial in setting the initial parameters of the function that optimises the automatic classifier.

Fig. 8 Interactive Text Categorisation with R. Adjust angular coefficient to decrease the number of false positives.

From a theoretical point of view, there are interesting open questions about the meaning of the decision line found in the likelihood space. In particular, whether the solution has an equivalent form in the data space and in Decision Theory in general, or whether the new solution defines a completely new decision theory in the data space. Another important aspect that was not discussed in this chapter is that the smoothing parameters α and β should be optimised for each single feature instead of being equal for all the features. This problem alone would require a completely different user interface, or, in terms of classical machine learning, a study on how to choose parameters individually in an efficient way.

References

[Amershi et al(2014)Amershi, Cakmak, Knox, and Kulesza] Amershi S, Cakmak M, Knox WB, Kulesza T (2014) Power to the people: The role of humans in interactive machine learning. AI Mag 35(4):105–120, URL `http://www.aaai.org/ojs/index.php/aimagazine/article/view/2513`

[Ankerst et al(1999)Ankerst, Elsen, Ester, and Kriegel] Ankerst M, Elsen C, Ester M, Kriegel HP (1999) Visual classification: An interactive approach to decision tree construction. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, KDD '99, pp 392–396, DOI 10.1145/312129.

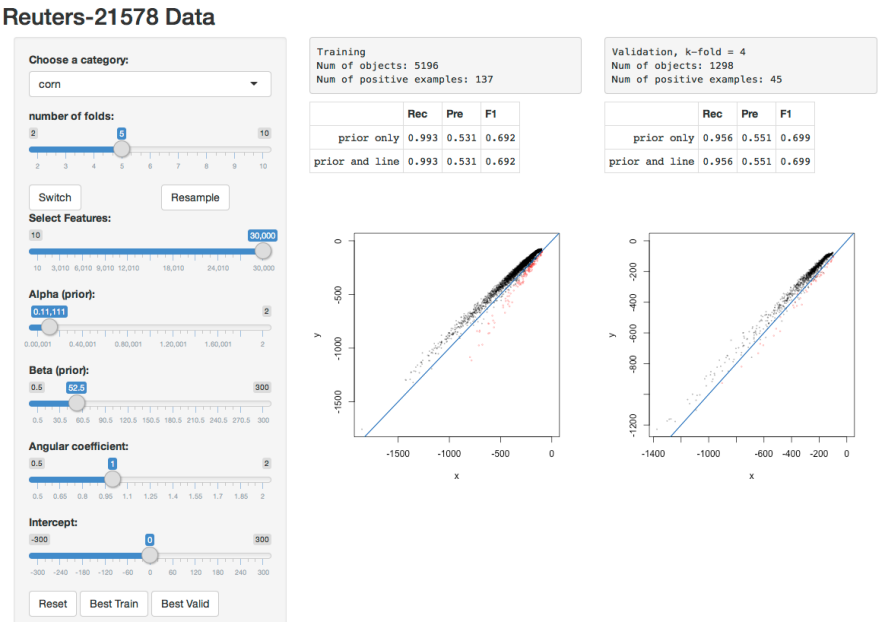

Fig. 9 Interactive Text Categorisation with R. Change the value of the smoothing parameters to see how points move around the zero-one loss function.

312298, URL http://doi.acm.org/10.1145/312129.312298

[Ankerst et al(2000a)Ankerst, Ester, and Kriegel] Ankerst M, Ester M, Kriegel HP (2000a) 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, Boston, MA, USA, August 20-23, 2000, pp 179–188

[Ankerst et al(2000b)Ankerst, Ester, and Kriegel] Ankerst M, Ester M, Kriegel HP (2000b) 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, ACM, New York, NY, USA, KDD '00, pp 179–188, DOI 10.1145/347090.347124, URL http://doi.acm.org/10.1145/347090.347124

[Behrisch et al(2014)Behrisch, Korkmaz, Shao, and Schreck] Behrisch M, Korkmaz F, Shao L, Schreck T (2014) Feedback-Driven Interactive Exploration of Large Multidimensional Data Supported by Visual Classifier. In: Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on, IEEE CS Press, pp 43–52, DOI 10.1109/VAST.2014.7042480

[Caruana and Niculescu-Mizil(2006)] Caruana R, Niculescu-Mizil A (2006) An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning, ACM, New York, NY, USA, ICML '06, pp 161–168, DOI 10.1145/1143844.1143865, URL http://doi.acm.org/10.1145/1143844.1143865

[Crestani et al(1998)Crestani, Lalmas, Van Rijsbergen, and Campbell] Crestani F, Lalmas M, Van Rijsbergen CJ, Campbell I (1998) Is this document relevant? probably. a survey of probabilistic models in information retrieval. ACM Comput Surv 30(4):528–552, DOI 10.1145/299917.299920, URL http://doi.acm.org/10.1145/299917.299920

[Di Nunzio(2009)] Di Nunzio G (2009) Using scatterplots to understand and improve probabilistic models for text categorization and retrieval. Int J Approx Reason 50(7):945–956

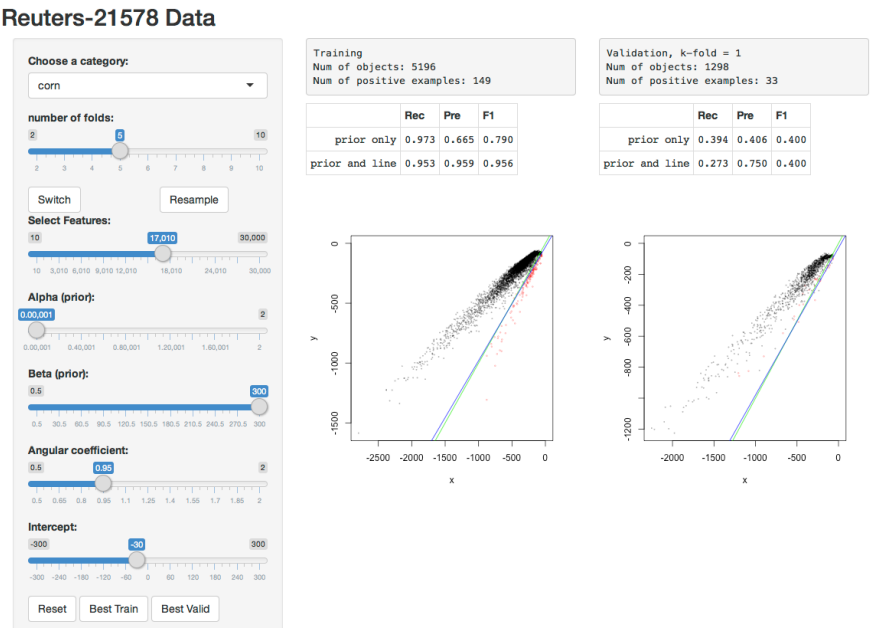

Fig. 10 Interactive Text Categorisation. Example of overfitting with an almost perfect score on the training data.

[Di Nunzio(2014a)] Di Nunzio G (2014a) A new decision to take for cost-sensitive Naïve Bayes classifiers. Inf Proc & Manag 50(5):653 – 674, DOI http://dx.doi.org/10.1016/j.ipm.2014.04.008, URL http://www.sciencedirect.com/science/article/pii/S0306457314000363

[Di Nunzio(2014b)] Di Nunzio G (2014b) Visual classification. In: Aggarwal CC (ed) Data Classification: Algorithms and Applications, CRC Press, pp 607–632, URL http://www.crcnetbase.com/doi/abs/10.1201/b17320-24

[Di Nunzio and Micarelli(2004)] Di Nunzio G, Micarelli A (2004) Pushing "underfitting" to the limit: Learning in bidimensional text categorization. In: Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004, pp 465–469

[Di Nunzio and Sordoni(2012)] Di Nunzio G, Sordoni A (2012) How well do we know Bernoulli? In: Proceedings of the 3rd Italian Information Retrieval Workshop, Bari, Italy, January 26-27, 2012, pp 38–44, URL http://ceur-ws.org/Vol-835/paper5.pdf

[Domingos and Pazzani(1997)] Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29(2-3):103–130, DOI 10.1023/A:1007413511361, URL http://dx.doi.org/10.1023/A:1007413511361

[Duda et al(2000)Duda, Hart, and Stork] Duda RO, Hart PE, Stork DG (2000) Pattern Classification, 2nd edn. Wiley-Interscience

[Elkan(2001)] Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'01, pp 973–978, URL http://dl.acm.org/citation.cfm?id=1642194.1642224

[Fails and Olsen(2003)] Fails JA, Olsen DR Jr (2003) Interactive machine learning. In: Proceedings of the 8th International Conference on Intelligent User Interfaces, ACM, New York, NY,

USA, IUI '03, pp 39–45, DOI 10.1145/604045.604056, URL http://doi.acm.org/10.1145/604045.604056

[Kucher and Kerren(2014)] Kucher K, Kerren A (2014) Text visualization browser: A visual survey of text visualization techniques. In: IEEE Information Visualization (InfoVis'14), Paris, p Poster Abstract

[Kulesza et al(2015)Kulesza, Burnett, Wong, and Stumpf] Kulesza T, Burnett M, Wong WK, Stumpf S (2015) Principles of explanatory debugging to personalize interactive machine learning. In: Proceedings of the 20th International Conference on Intelligent User Interfaces, ACM, New York, NY, USA, IUI '15, pp 126–137, DOI 10.1145/2678025.2701399, URL http://doi.acm.org/10.1145/2678025.2701399

[Mitchell(1997)] Mitchell TM (1997) Machine Learning, 1st edn. McGraw-Hill, Inc., New York, NY, USA

[Mladenic and Grobelnik(1999)] Mladenic D, Grobelnik M (1999) Feature selection for unbalanced class distribution and Naïve Bayes. In: Proceedings of the Sixteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '99, pp 258–267, URL http://dl.acm.org/citation.cfm?id=645528.657649

[Neyman and Pearson(1933)] Neyman J, Pearson ES (1933) On the problem of the most efficient tests of statistical hypotheses. Philosoph Trans of the R Soc of London Series A, Containing Papers of a Mathematical or Physical Character 231:pp. 289–337, URL http://www.jstor.org/stable/91247

[Sebastiani(2002)] Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47, DOI 10.1145/505282.505283, URL http://doi.acm.org/10.1145/505282.505283

[Shneiderman(1997)] Shneiderman B (1997) Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA

[Shneiderman(2002)] Shneiderman B (2002) Inventing discovery tools: Combining information visualization with data mining. Inf Vis 1(1):5–12, DOI 10.1057/palgrave/ivs/9500006, URL http://dx.doi.org/10.1057/palgrave/ivs/9500006

[Singh and Raj(2004)] Singh R, Raj B (2004) Classification in likelihood spaces. Technometrics 46(3):318–329, DOI 10.1198/004017004000000347, URL http://www.tandfonline.com/doi/abs/10.1198/004017004000000347, http://www.tandfonline.com/doi/pdf/10.1198/004017004000000347

[Sokolova and Lapalme(2009)] Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process & Manag 45(4):427–437, DOI 10.1016/j.ipm.2009.03.002, URL http://dx.doi.org/10.1016/j.ipm.2009.03.002

[Tan et al(2005)Tan, Steinbach, and Kumar] Tan PN, Steinbach M, Kumar V (2005) Introduction to Data Mining, first edition edn. Addison-Wesley, Boston, MA, USA

[Ware et al(2002)Ware, Frank, Holmes, Hall, and Witten] Ware M, Frank E, Holmes G, Hall M, Witten IH (2002) Interactive machine learning: Letting users build classifiers. Int J Hum-Comput Stud 56(3):281–292, URL http://dl.acm.org/citation.cfm?id=514412.514417

[Webb and Pazzani(1998)] Webb GI, Pazzani MJ (1998) Adjusted probability Naïve Bayesian induction. In: Advanced Topics in Artificial Intelligence, 11th Australian Joint Conference on Artificial Intelligence, AI '98, Brisbane, Australia, July 13-17, 1998, Selected Papers, pp 285–295, DOI 10.1007/BFb0095060, URL http://dx.doi.org/10.1007/BFb0095060

[Yuan et al(2012)Yuan, Cong, and Thalmann] Yuan Q, Cong G, Thalmann NM (2012) Enhancing Naïve Bayes with various smoothing methods for short text classification. In: Proceedings of the 21st International Conference Companion on World Wide Web, ACM, New York, NY, USA, WWW '12 Companion, pp 645–646, DOI 10.1145/2187980.2188169, URL http://doi.acm.org/10.1145/2187980.2188169