



Using the Naive Bayes as a discriminative model

Elie Azeraf*
elie.azeraf@ibm.com
IBM GBS France
Paris, France

Emmanuel Monfrini
Institut Polytechnique de Paris,
Telecom SudParis
Paris, France

Wojciech Pieczynski
Institut Polytechnique de Paris,
Telecom SudParis
Paris, France

ABSTRACT

For classification tasks, probabilistic graphical models are usually categorized into two disjoint classes: generative or discriminative. It depends on the posterior probability $p(x|y)$ of the label x given the observation y computation. On the one hand, generative models, like the Naive Bayes or the Hidden Markov Model (HMM), need the computation of the joint probability $p(x, y)$, before using the Bayes rule to compute $p(x|y)$. On the other hand, discriminative models compute $p(x|y)$ directly, regardless of the observations' law. They are intensively used nowadays, with models as Logistic Regression or Conditional Random Fields (CRF). However, the recent Entropic Forward-Backward algorithm shows that the HMM, considered as a generative model, can also match the discriminative one's definition. This example leads to question if it is the case for other generative models. In this paper, we show that the Naive Bayes can also match the discriminative model definition, so it can be used in either a generative or a discriminative way. Moreover, this observation also discusses the notion of Generative-Discriminative pairs, linking, for example, Naive Bayes and Logistic Regression, or HMM and CRF. Related to this point, we show that the Logistic Regression can be viewed as a particular case of the Naive Bayes used in a discriminative way.

CCS CONCEPTS

• **Mathematics of computing** → **Computing most probable explanation.**

KEYWORDS

Probabilistic Graphical Models, Generative Model, Discriminative Model, Logistic Regression, Naive Bayes, Generative-Discriminative pair.

ACM Reference Format:

Elie Azeraf, Emmanuel Monfrini, and Wojciech Pieczynski. 2021. Using the Naive Bayes as a discriminative model. In *2021 13th International Conference on Machine Learning and Computing (ICMLC '21)*, February 26-March 1, 2021, Shenzhen, China. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3457682.3457697>

*Elie Azeraf is also a member of Institut Polytechnique de Paris, Telecom SudParis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICMLC '21, February 26-March 1, 2021, Shenzhen, China

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8931-0/21/02...\$15.00

<https://doi.org/10.1145/3457682.3457697>

1 INTRODUCTION

Given an input data y , also called observation, and a discrete finite output space $\Lambda_X = \{\lambda_1, \dots, \lambda_N\}$, a classification task consists in computing the different label probabilities $p(x = \lambda_i|y)$. They can be used in many areas as Natural Language Processing with Text Classification [15] and Sentiment Analysis [23], or Computer Vision with Image Classification [8] and Object Detection [22], among many others. These probabilistic models are usually categorized into two disjoint classes: generative or discriminative. From [28], a generative model learns the joint probability $p(x, y)$, then uses the Bayes rule to compute the posterior probability $p(x|y)$:

$$\forall \lambda_i \in \Lambda_X, p(x = \lambda_i|y) = \frac{p(x = \lambda_i, y)}{\sum_{\lambda_j \in \Lambda_X} p(x = \lambda_j, y)} \quad (1)$$

To train a generative model in a supervised context, one usually maximizes the joint likelihood $p(x, y)$ of the training data. For example, in the case where y is a discrete variable, it consists in counting the different patterns' occurrence. Among the most popular generative models, we can cite the Naive Bayes [21, 25, 34, 40], the Hidden Markov Model (HMM) [7, 30, 37], or the Gaussian Mixture Model [32, 33].

On the other hand, a discriminative model computes $p(x|y)$ directly, which means that $p(x, y)$ does not have to be known. We can cite the Logistic Regression [10, 12, 26, 29, 41], the Maximum Entropy Markov Model (MEMM) [24], the Conditional Random Fields (CRF) [16, 38], or even some Artificial Neural Networks (ANN) [11, 18], among the most popular ones. They are usually trained by fitting the conditional distribution $p(x|y)$, or minimizing a loss function, thanks to optimization algorithms like the gradient descent [36].

Many papers use these definitions [4, 6, 17, 27, 35, 38, 39, 42] and compare both approaches, with a general preference for the second category where the task does not have unlabeled data. Indeed, generative models' main criticism concerns their learning strategy, imposing to learn the joint probability, and therefore the observation's distribution. This learning method constrains the observation's features during classification tasks. Indeed, it is considered impossible to handle arbitrary features with a generative model [13, 24, 38], except with a restrictive independence condition. The next section will illustrate this point.

We focus on the HMM, represented in figure 1, with an observed process $y_{1:T} = (y_1, \dots, y_T)$, and a hidden one $x_{1:T} = (x_1, \dots, x_T)$. This model is considered as a generative probabilistic one. Indeed, if one wants to compute $p(x_t = \lambda_i|y_{1:T})$, he can apply the Forward-Backward algorithm [30, 31] described in the appendices. In agreement with the generative model definition, this algorithm firstly computes the joint probability $p(x_t = \lambda_i, y_{1:T})$ before computing the posterior one. Moreover, all the parameters are learned with maximum likelihood. However, a recent work about HMM

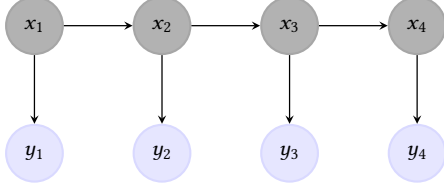


Figure 1: Probabilistic graph of the HMM

presents the Entropic Forward-Backward algorithm [3], also described in the appendices. It allows computing the posterior distribution $p(x_t = \lambda_i | y_{1:T})$ directly, with using neither the joint distribution, nor the observations' one. It also allows training the HMM with optimization algorithms as the gradient descent. Therefore, in this case, the HMM matches the discriminative model's definition.

The HMM example above leads to question the categorization of probabilistic models into generative and discriminative ones, which is the aim of the paper. We present two contributions:

- We show that the Naive Bayes, a popular generative model, can also match the definition of a discriminative one. It is another example showing that the traditional definitions of both discriminative and generative models do not necessarily lead to disjoint categories.
- We discuss the notion of Generative-Discriminative pairs [28, 38], linking a generative model with its discriminative counterpart, as for example “Naive Bayes - Logistic Regression” and “HMM - CRF”. Considering the first example, we show that the Logistic Regression is a particular case of the Naive Bayes used in a discriminative way.

This paper is organized as follows. In the next section, we recall the Naive Bayes model's law with its oriented probabilistic graph, and we present the usual generative way of computing the label probabilities. Then, we show how to compute these probabilities in a discriminative way, without using the observation's law. In the third section we show that Logistic Regression model can be viewed as a particular case of the Naive Bayes used in a discriminative way. Conclusion and perspectives lie at the end of the paper.

2 NAIVE BAYES AS A DISCRIMINATIVE MODEL

2.1 Naive Bayes model

The Naive Bayes is a probabilistic graphical model [14] considering the observations $y_{1:T} = (y_1, \dots, y_T)$, with y_t taking its value in Ω_{Y_t} , and a hidden variable x taking its value in Λ_X . The joint probability of $(x, y_{1:T})$ is given with:

$$\begin{aligned} p(x, y_{1:T}) &= p(x)p(y_1|x)p(y_2|x) \dots p(y_{T-1}|x)p(y_T|x) \\ &= p(x) \prod_{t=1}^T p(y_t|x) \end{aligned} \quad (2)$$

The oriented probabilistic graph is given in figure 2.

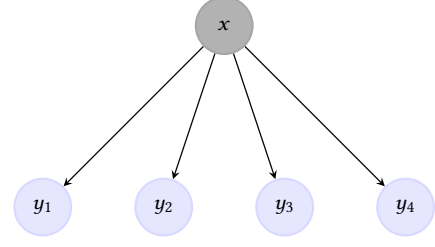


Figure 2: Probabilistic graph of the Naive Bayes

In the generative way, as usually used, the Naive Bayes computes, for each $\lambda_i \in \Lambda_X$, $p(x = \lambda_i | y_{1:T})$ as follows:

$$\begin{aligned} p(x = \lambda_i | y_{1:T}) &= \frac{p(x = \lambda_i, y_{1:T})}{\sum_{\lambda_j \in \Lambda_X} p(x = \lambda_j, y_{1:T})} \\ &= \frac{p(x = \lambda_i) \prod_{t=1}^T p(y_t | x = \lambda_i)}{\sum_{\lambda_j \in \Lambda_X} p(x = \lambda_j) \prod_{t=1}^T p(y_t | x = \lambda_j)} \\ &= \frac{\pi(i) \prod_{t=1}^T b_i^{(t)}(y_t)}{\sum_{\lambda_j \in \Lambda_X} \pi(j) \prod_{t=1}^T b_j^{(t)}(y_t)} \end{aligned} \quad (3)$$

with the notations:

- $\pi(i) = p(x = \lambda_i)$;
- $b_i^{(t)}(y) = p(y_t | x = \lambda_i)$.

With (3), the Naive Bayes effectively matches the generative model definition, as it first computes the joint probability $p(x = \lambda_i, y_{1:T})$, and then the posterior one $p(x = \lambda_i | y_{1:T})$.

Used in this way, it is facing difficulties to consider arbitrary features. Indeed, let us consider, for example, $\Omega_{Y_t} = \Omega_Y^2$, for each t , with $\Omega_Y = \{\omega_1, \dots, \omega_M\}$. Thus, each observation has two features: $y_t = (\omega_i, \omega_j)$. The different parameters are then learned by maximum likelihood estimation, which consists in counting the different patterns:

$$\pi(i) = \frac{f(i)}{L}, \quad b_i^{(t)}(y_t) = \frac{f_i^{(t)}(j, k)}{f(i)},$$

with L the number of training samples, $f(i)$ the number of times $x = \lambda_i$ in this training sample, and $f_i^{(t)}(j, k)$ the number of times $y_t = (\omega_j, \omega_k)$ when $x = \lambda_i$. Therefore, an observation y_t has its value $b_i^{(t)}(y_t)$ different from 0 if and only if there is an observation in the training set having the same features at position t . If this estimation method is possible with a small number of features, it quickly becomes intractable when the number of features increases, making mandatory to suppose them independent. It is especially the case for Natural Language Processing tasks, where features can be suffixes of any length, prefixes, some word's characteristics, or large numerical vector issued from an embedding method [2].

2.2 Computing directly $p(x = \lambda_i | y_{1:T})$ with the Naive Bayes model

In this section, we show that with the Naive Bayes it is possible to compute $p(x = \lambda_i | y_{1:T})$ directly, without previous computation of $p(x = \lambda_i, y_{1:T})$. To simplify notations, we set, for each $\lambda_i \in \Lambda_X$:

$$L_{y_t}^{(t)}(i) = p(x = \lambda_i | y_t).$$

Proposition 1. Let $p(x, y_{1:T})$ be a Naive Bayes distribution. Then for each $\lambda_i \in \Lambda_X$:

$$p(x = \lambda_i | y_{1:T}) = \frac{\pi(i)^{1-T} \prod_{t=1}^T L_{y_t}^{(t)}(i)}{\sum_{j=1}^N \pi(j)^{1-T} \prod_{t=1}^T L_{y_t}^{(t)}(j)} \quad (4)$$

Proof. For each $\lambda_i \in \Lambda_X$, we set:

$$\delta(i) = \pi(i)^{1-T} \prod_{t=1}^T L_{y_t}^{(t)}(i) \quad (5)$$

We have:

$$p(x = \lambda_i, y_{1:T}) = p(y_1)p(y_2) \dots p(y_{T-1})p(y_T)\delta(i). \quad (6)$$

Indeed,

$$\begin{aligned} p(x = \lambda_i, y_{1:T}) &= \pi(i) \prod_{t=1}^T b_i^{(t)}(y_t) = \pi(i) \prod_{t=1}^T \frac{p(y_t, x = \lambda_i)}{p(x = \lambda_i)} \\ &= \pi(i)^{1-T} \prod_{t=1}^T p(y_t) L_{y_t}^{(t)}(i) = \delta(i) \prod_{t=1}^T p(y_t). \end{aligned}$$

Therefore

$$p(x = \lambda_i | y_{1:T}) = \frac{p(x = \lambda_i, y_{1:T})}{\sum_{j=1}^N p(x = \lambda_j, y_{1:T})} = \frac{\delta(i)}{\sum_{j=1}^N \delta(j)}, \quad (7)$$

which ends the proof.

Finally, given (4), the Naive Bayes also matches the definition of a discriminative model: it allows computing $p(x | y_{1:T})$ directly and does use neither the observations' law nor the joint one. Therefore, through this example, we see that in some situations, "discriminative" and "generative" definitions do not describe the deep nature of a model, but rather the way it is used.

When differentiable functions model $L_{y_t}^{(t)}$, one can train the Naive Bayes used in a discriminative way with (4) using gradient descent algorithms. In this case, the training process consists in computing the gradient of a given loss function by applying the backpropagation [19, 20] algorithm, and updating the model's parameters. In this way, the Naive Bayes can be applied for any classification tasks and consider arbitrary observation features. For example, a homogeneous Naive Bayes used in a discriminative way, modelling $L_{y_t}^{(t)}$ with the same function independent of t , can be applied to classify text of various lengths. First of all, one has to use an embedding method like Flair [1], FastText [5], or BERT [9], to transform word into numerical vector. Then, he has to select a function to model $L_{y_t}^{(t)}$, for example, a feedforward neural network.

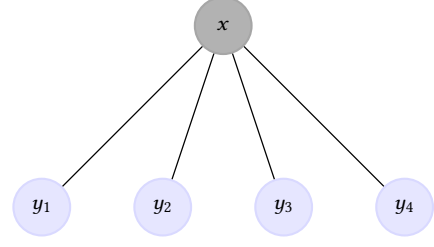


Figure 3: Probabilistic graph of the Logistic Regression

3 THE LOGISTIC REGRESSION: A PARTICULAR CASE OF THE NAIVE BAYES USED IN A DISCRIMINATIVE WAY

3.1 Logistic Regression model

Let us consider multinomial Logistic Regression. For each $t \in \mathbb{N}$, $\Omega_{Y_t} = \mathbb{R}$, $p(x = \lambda_i | y_{1:T})$ is given with

$$p(x = \lambda_i | y_{1:T}) = \frac{\exp(W_i \cdot y_{1:T} + b_i)}{\sum_{j=1}^N \exp(W_j \cdot y_{1:T} + b_j)}, \quad (8)$$

where for each $i \in \{1, \dots, N\}$, $W_i \in \mathbb{R}^T$, $b_i \in \mathbb{R}$, and \cdot denotes the matrix product. Figure 3 represents the probabilistic graph of this model.

The Logistic Regression is a probabilistic graphical model used in numerous applications. One can train it with a gradient descent algorithm. It is considered as a discriminative model as it computes directly $p(x | y_{1:T})$. With this model, we notice that each probability computed is strictly greater than 0.

3.2 Logistic Regression and Naive Bayes

Logistic Regression is frequently linked with the Naive Bayes with the notion of Generative – Discriminative pairs [28, 38], assuming it is the discriminative counterpart of the Naive Bayes. However, we are going to show that the Logistic Regression can also be considered as a particular case of the Naive Bayes.

Proposition 2. Let us consider a Naive Bayes with $p(x = \lambda_i | y_{1:T})$ given by (4). If, for each $i \in \{1, \dots, N\}$, $t \in \{1, \dots, T\}$:

$$L_{y_t}^{(t)}(i) = \frac{\exp(a_i^{(t)} y_t + c_i^{(t)})}{\sum_{k=1}^N \exp(a_k^{(t)} y_t + c_k^{(t)})}, \quad (9)$$

with $a_i^{(t)} \in \mathbb{R}$ and $c_i^{(t)} \in \mathbb{R}$, and $\pi(i) > 0$, then it is a Logistic Regression.

Proof. We start from (4), for each $\lambda_i \in \Lambda_X$:

$$\begin{aligned}
 p(x = \lambda_i | y_{1:T}) &= \frac{\pi(i)^{1-T} \prod_{t=1}^T L_{y_t}^{(t)}(i)}{\sum_{j=1}^N \pi(j)^{1-T} \prod_{t=1}^T L_{y_t}^{(t)}(j)} \\
 &= \frac{\pi(i)^{1-T} \prod_{t=1}^T \frac{\exp(a_i^{(t)} y_t + c_i^{(t)})}{\sum_{k=1}^N \exp(a_k^{(t)} y_t + c_k^{(t)})}}{\sum_{j=1}^N \pi(j)^{1-T} \prod_{t=1}^T \frac{\exp(a_j^{(t)} y_t + c_j^{(t)})}{\sum_{k=1}^N \exp(a_k^{(t)} y_t + c_k^{(t)})}} \\
 &= \frac{\pi(i)^{1-T} \prod_{t=1}^T \exp(a_i^{(t)} y_t + c_i^{(t)})}{\sum_{j=1}^N \pi(j)^{1-T} \prod_{t=1}^T \exp(a_j^{(t)} y_t + c_j^{(t)})} \\
 &= \frac{\exp\left((1-T) \log(\pi(i)) + \sum_{t=1}^T a_i^{(t)} y_t + c_i^{(t)}\right)}{\sum_{j=1}^N \exp\left((1-T) \log(\pi(j)) + \sum_{t=1}^T a_j^{(t)} y_t + c_j^{(t)}\right)}. \tag{10}
 \end{aligned}$$

Setting:

- $b_i = (1-T) \log(\pi(i)) + \sum_{t=1}^T c_i^{(t)}$;
- $W_i = [a_i^{(1)}, a_i^{(2)}, \dots, a_i^{(T)}]$.

We verify that $b_i \in \mathbb{R}$ and $W_i \in \mathbb{R}^T$.

Then:

$$p(x = \lambda_i | y_{1:T}) = \frac{\exp(W_i \cdot y_{1:T} + b_i)}{\sum_{j=1}^N \exp(W_j \cdot y_{1:T} + b_j)}, \tag{11}$$

which is the equation (8) defining a Logistic Regression and ends the proof.

Proposition 3. Let us consider a Logistic Regression (8). This model can be viewed as a particular Naive Bayes used in the discriminative way with the parameter $L^{(t)}$ defined as in (9).

Proof. We start from (8), thanks to (10) and the proof of Proposition 2, we set:

- $[a_i^{(1)}, a_i^{(2)}, \dots, a_i^{(T)}] = W_i$
- $\pi(i) = \frac{1}{N}$
- $c_i^{(1)} = -(1-T) \log(\frac{1}{N}) + b_i$
- For $t = 2, \dots, T, c_i^{(t)} = 0$.

Therefore, (8) becomes:

$$\begin{aligned}
 p(x = \lambda_i | y_{1:T}) &= \frac{\exp(W_i \cdot y_{1:T} + b_i)}{\sum_{j=1}^N \exp(W_j \cdot y_{1:T} + b_j)} \\
 &= \frac{\exp\left(a_i \cdot y_{1:T} + (1-T) \log(\frac{1}{N}) + c_i^{(1)} + \sum_{t=2}^T c_i^{(t)}\right)}{\sum_{j=1}^N \exp\left(a_j \cdot y_{1:T} + (1-T) \log(\frac{1}{N}) + c_j^{(1)} + \sum_{t=2}^T c_j^{(t)}\right)} \\
 &= \frac{\exp\left(\sum_{t=1}^T a_i^{(t)} y_t + (1-T) \log(\pi(i)) + \sum_{t=1}^T c_i^{(t)}\right)}{\sum_{j=1}^N \exp\left(\sum_{t=1}^T a_j^{(t)} y_t + (1-T) \log(\pi(j)) + \sum_{t=1}^T c_j^{(t)}\right)}
 \end{aligned}$$

Which is the equation (10) defining a Naive Bayes used in the discriminative way with $L^{(t)}$ as in (9), ending the proof. We notice that the possible definition of $\pi(i)$ and $c_i^{(t)}$ is not unique.

Proposition 3 shows that one can view the Logistic Regression as a particular case of the Naive Bayes used in a discriminative way. Added to Proposition 2, we have shown the equivalence between both models. However, this equivalence exists only for Naive Bayes with probabilities strictly greater than 0, showing that the latter is more general than the Logistic Regression.

4 CONCLUSION

This paper presents how to use the Naive Bayes as a discriminative model. A practical consequence is that it can be used in both generative and discriminative ways; the latter allowing arbitrary feature consideration. Moreover, we show that the Logistic Regression, usually presented as the discriminative counterpart of the Naive Bayes, can be seen as a particular case of the latter used in a discriminative way. A general conclusion is that the usual definitions of generative and discriminative models do not lead to disjoint families; and there are models that simultaneously satisfy both of them. An interesting perspective would consist in extending ideas of the paper to other popular generative models and examining whether they can also be used in a discriminative way.

REFERENCES

- [1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1638–1649.
- [2] Felipe Almeida and Geraldo Xexéo. 2019. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069* (2019).
- [3] Elie Azeraf, Emmanuel Monfrini, Emmanuel Vignon, and Wojciech Pieczynski. 2020. Hidden Markov Chains, Entropic Forward-Backward, and Part-Of-Speech Tagging. *arXiv preprint arXiv:2005.10629* (2020).
- [4] Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.
- [5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
- [6] Guillaume Bouchard and Bill Triggs. 2004. The tradeoff between generative and discriminative classifiers.
- [7] Olivier Cappé, Eric Moulines, and Tobias Rydén. 2009. Inference in hidden markov models.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

- [10] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York.
- [11] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
- [12] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression*. Vol. 398. John Wiley & Sons.
- [13] Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.
- [14] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.
- [15] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information* 10, 4 (2019), 150.
- [16] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).
- [17] Julia A Lasserre, Christopher M Bishop, and Thomas P Minka. 2006. Principled hybrids of generative and discriminative models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 1. IEEE, 87–94.
- [18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
- [19] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1, 4 (1989), 541–551.
- [20] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. 396–404.
- [21] David D Lewis. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*. Springer, 4–15.
- [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [23] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 142–150. <https://www.aclweb.org/anthology/P11-1015>
- [24] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. [n.d.]. Maximum Entropy Markov Models for Information Extraction and Segmentation.
- [25] Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Vol. 752. Citeseer, 41–48.
- [26] Scott Menard. 2002. *Applied logistic regression analysis*. Vol. 106. Sage.
- [27] Tom Minka. 2005. *Discriminative models, not discriminative training*. Technical Report. Technical Report MSR-TR-2005-144, Microsoft Research.
- [28] Andrew Y Ng and Michael I Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*. 841–848.
- [29] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. 2002. An introduction to logistic regression analysis and reporting. *The journal of educational research* 96, 1 (2002), 3–14.
- [30] Lawrence Rabiner and B Juang. 1986. An introduction to hidden Markov models. *ieee assp magazine* 3, 1 (1986), 4–16.
- [31] Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.
- [32] Carl Edward Rasmussen. 2000. The infinite Gaussian mixture model. In *Advances in neural information processing systems*. 554–560.
- [33] Douglas Reynolds. 2009. *Gaussian Mixture Models*. Springer US, Boston, MA, 659–663. https://doi.org/10.1007/978-0-387-73003-5_196
- [34] Irina Rish et al. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3. 41–46.
- [35] Wolfgang Roth, Robert Peharz, Sebastian Tschiatschek, and Franz Pernkopf. 2018. Hybrid generative-discriminative training of Gaussian mixture models. *Pattern recognition letters* 112 (2018), 131–137.
- [36] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).
- [37] Ruslan Leont'evich Stratonovich. 1965. Conditional markov processes. In *Non-linear transformations of stochastic processes*. Elsevier, 427–453.
- [38] Charles Sutton and Andrew McCallum. [n.d.]. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning* 2 ([n. d.]), 93–128.
- [39] Ilkay Ulusoy and Christopher M Bishop. 2005. Generative versus discriminative methods for object recognition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 258–265.
- [40] Geoffrey I. Webb. 2010. *Naive Bayes*. Springer US, Boston, MA, 713–714. https://doi.org/10.1007/978-0-387-30164-8_576
- [41] Raymond E Wright. 1995. Logistic regression. (1995).
- [42] Oksana Yakhnenko, Adrian Silvescu, and Vasant Honavar. 2005. Discriminatively trained markov model for sequence classification. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 8–pp.

A ALGORITHMS TO COMPUTE FOR HMM

In this appendix, we present both HMM's algorithms to restore, for each $\lambda_i \in \Lambda_X, t \in \{1, \dots, T\}, y_{1:T} \in (\Omega_Y)^T, p(x_t = \lambda_i | y_{1:T})$: the Forward-Backward algorithm, matching the generative definition, and the Entropic Forward-Backward, which matches the discriminative one. We consider a homogeneous HMM with the following notations:

- $\pi(i) = p(x_t = \lambda_i)$;
- $a_i(j) = p(x_{t+1} = \lambda_j | x_t = \lambda_i)$;
- $b_i(y_t) = p(y_t | x_t = \lambda_i)$;
- $L_{y_t}(i) = p(x_t = \lambda_i | y_t)$.

A.1 HMM as a generative model: The Forward-Backward algorithm

The Forward-Backward algorithm consists in computing $p(x_t = \lambda_i | y_{1:T})$ as follows:

$$p(x_t = \lambda_i | y_{1:T}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)}$$

with forward probabilities α computed with the following recursion:

$$\begin{aligned} \alpha_1(i) &= \pi(i)b_i(y_1) \\ \alpha_{t+1}(i) &= b_i(y_{t+1}) \sum_{j=1}^N \alpha_t(j)a_j(i) \end{aligned}$$

And the backward probabilities β :

$$\begin{aligned} \beta_T(i) &= 1 \\ \beta_t(i) &= \sum_{j=1}^N \beta_{t+1}(j)a_i(j)b_j(y_{t+1}) \end{aligned}$$

A.2 HMM as a discriminative model: The Entropic Forward-Backward algorithm

The Entropic Forward-Backward algorithm consists in computing $p(x_t = \lambda_i | y_{1:T})$ as follows:

$$p(x_t = \lambda_i | y_{1:T}) = \frac{\alpha_t^E(i)\beta_t^E(i)}{\sum_{j=1}^N \alpha_t^E(j)\beta_t^E(j)}$$

with entropic forward probabilities α^E computed as:

$$\begin{aligned} \alpha_1^E(i) &= L_{y_1}(i); \\ \alpha_{t+1}^E(i) &= \frac{L_{y_{t+1}}(i)}{\pi(i)} \sum_{j=1}^N \alpha_t^E(j)a_{ji}(i) \end{aligned}$$

And the entropic backward β^E :

$$\begin{aligned} \beta_T^E(i) &= 1; \\ \beta_t^E(i) &= \sum_{j=1}^N \frac{L_{y_{t+1}}(j)}{\pi(j)} \beta_{t+1}^E(j)a_{ji}(i) \end{aligned}$$