Naive Feature Selection: Sparsity in Naive Bayes

Armin Askari
UC Berkeley

Alex d'Aspremont
CNRS & École Normale Supérieure

Laurent El Ghaoui
UC Berkeley

Abstract

Due to its linear complexity, naive Bayes classification remains an attractive supervised learning method, especially in very large-scale settings. We propose a sparse version of naive Bayes, which can be used for feature selection. This leads to a combinatorial maximum-likelihood problem, for which we provide an exact solution in the case of binary data, or a bound in the multinomial case. We prove that our bound becomes tight as the marginal contribution of additional features decreases. Both binary and multinomial sparse models are solvable in time almost linear in problem size, representing a very small extra relative cost compared to the classical naive Bayes. Numerical experiments on text data show that the naive Bayes feature selection method is as statistically effective as state-of-the-art feature selection methods such as recursive feature elimination, l_1-penalized logistic regression and LASSO, while being orders of magnitude faster. For a large data set, having more than with 1.6 million training points and about 12 million features, and with a non-optimized CPU implementation, our model can be trained in less than 15 seconds [1].

1 Introduction

Modern, large-scale data sets call for classification methods that scale mildly (e.g. linearly) with problem size. In this context, the classical naive Bayes model remains a very competitive baseline, due to its linear complexity in the number of training points and features. In fact, it is sometimes the only feasible approach in very large-scale settings, particularly in text applications, where the number of features can easily be in the millions.

Feature selection, on the other hand, is a key component of machine learning pipelines, for two main reasons: i) to reduce effects of overfitting by eliminating noisy, non-informative features and ii) to provide interpretability. In essence, feature selection is a combinatorial problem, involving the selection of a few features in a potentially large population. State-of-the-art methods for feature selection employ some heuristic to address the combinatorial aspect, and the most effective ones are usually computationally costly. For example, LASSO (Tibshirani, 1996) or l_1-SVM models (Fan et al., 2008) are based on solving a l_1-penalized convex problem in order to achieve sparsity (at the expense of tuning a hyper parameter to attain a desired sparsity level).

Since naive Bayes corresponds to a linear classification rule, feature selection in this setting is directly related to the sparsity of the vector of classification coefficients. This work is devoted to a sparse variant of naive Bayes. Our main contributions are as follows.

- We formulate a sparse naive Bayes problem that involves a direct constraint on the cardinality of the vector of classification coefficients, leading to an interpretable naive Bayes model. No hyper-parameter tuning is required in order to achieve the target cardinality.
- We derive an exact solution of sparse naive Bayes in the case of binary data, and an approximate upper bound for general data, and show that it becomes increasingly tight as the marginal contribution of features decreases. Both models can be trained very efficiently, with an algorithm that scales almost linearly with the number of features and data points, just like classical naive Bayes.
- We show in experiments that our model significantly outperforms simple baselines (*e.g.*, thresholded naive Bayes, odds ratio), and achieves similar performance as more sophisticated feature selection methods, at a fraction of the computing cost.

[1] A python implementation of our model can be found at https://github.com/aspremon/NaiveFeatureSelection

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

Related Work on Naive Bayes Improvements. A large body of literature builds on the traditional naive Bayes classifier. A non-extensive list includes the seminal work by (Frank et al., 2002) introducing Weighted naive Bayes; Lazy Bayesian Learning by (Zheng and Webb, 2000); and the Tree-Augmented naive Bayes method by (Friedman et al., 1997). The paper (Webb et al., 2005) improves the computational complexity of the aforementioned methods, while maintaining the same accuracy. For a more complete discussion of modifications to naive Bayes, we refer the reader to (Jiang et al., 2007) and the references therein.

Related Work on Naive Bayes and Feature Selection. Of particular interest to this work are methods that employ feature selection. (Kim et al., 2006) use information-theoretic quantities for feature selection in text classification, while (Mladenic and Grobelnik, 1999) compare a host of different methods and shows the comparative efficacy of the Odds Ratio method. These methods often use ad hoc scoring functions to rank the importance of the different features. (Fleuret, 2004) uses the mutual information to select features in a fast way while (Zaidi et al., 2013) employs a weighting approach for selecting relevant features. (Boullé, 2007) achieve soft variable selection by introducing bayesian regularization into the training problem.

To our knowledge, the first work to directly address sparsity in the context of naive Bayes, with binary data only, is (Zheng et al., 2018). Their model does not directly address the requirement that the weight vector of the classification rule should be sparse, but does identify key features in the process. The method requires solving an approximation to the combinatorial feature selection problem via l_1-penalized logistic regression problem with non-negativity constraints, that has the same number of features and data points as the original one. Therefore the complexity of the method is the same as ordinary l_1-penalized logistic regression, which is relatively high. In contrast, our binary (Bernoulli) naive Bayes bound is exact, and has complexity almost linear in training problem size.

2 Background on Naive Bayes

In this paper, for simplicity only, we consider a two-class classification problem; the extension to the general multi-class case is straightforward.

Notation. For an integer m, $[m]$ is the set $\{1, \ldots, m\}$. The notation $\mathbf{1}$ denotes a vector of ones, with size inferred from context. The cardinality (number of nonzero elements) in a m-vector x is denoted $\|x\|_0$, whereas that of a finite set $\mathcal{I}$ is denoted $|\mathcal{I}|$. Unless otherwise specified, functional operations (such as $\max(0, \cdot)$) on vectors are performed element-wise. For $k \in [n]$, we say that a vector $w \in \mathbb{R}^n$ is k-sparse or has sparsity level $\alpha\%$ if at most k or $\alpha\%$ of its coefficients are nonzero respectively. For two vectors $f, g \in \mathbb{R}^m$, $f \circ g \in \mathbb{R}^m$ denotes the elementwise product. For a vector z, the notation $s_k(z)$ is the sum of the top k entries. Finally, $\mathbb{P}(A)$ denotes the probability of an event A.

Data Setup. We are given a non-negative data matrix $X \in \mathbb{R}_+^{n \times m} = [x^{(1)}, x^{(2)}, \ldots, x^{(n)}]^\top$ consisting of n data points, each with m dimensions (features), and a vector $y \in \{-1, 1\}^n$ that encodes the class information for the n data points, with C_+ and C_- referring to the positive and negative classes respectively. We define index sets corresponding to each class C_+, C_-, and their respective cardinality, and data averages:

$$\begin{aligned} \mathcal{I}_\pm &:= \{i \in [n] \ : \ y_i = \pm 1\}, \\ n_\pm &= |\mathcal{I}_\pm|, \\ f_\pm &:= \sum_{i \in \mathcal{I}_\pm} x^{(i)} = \pm(1/2) X^\top (y \pm \mathbf{1}) \end{aligned}$$

Naive Bayes. We are interested in predicting the class label of a test point $x \in \mathbb{R}^m$ via $\hat{y}(x) = \arg\max_{\epsilon \in \{-1,1\}} \mathbb{P}(C_\epsilon \mid x)$. To calculate the latter posterior probability, we employ Bayes' rule and then use the "naive" assumption that features are independent of each other: $\mathbb{P}(x \mid C_\epsilon) = \prod_{j=1}^m \mathbb{P}(x_j \mid C_\epsilon)$, leading to

$$\hat{y}(x) = \arg \max_{\epsilon \in \{-1,1\}} \log \mathbb{P}(C_\epsilon) + \sum_{j=1}^m \log \mathbb{P}(x_j | C_\epsilon). \quad (1)$$

In (1), we need to have an explicit model for $\mathbb{P}(x_j|C_i)$; in the case of binary or integer-valued features, we use Bernoulli or categorical distributions, while in the case of real-valued features we can use a Gaussian distribution. We then use the maximum likelihood principle (MLE) to determine the parameters of those distributions. Using a categorical distribution, $\mathbb{P}(C_\pm)$ simply becomes $n_\pm/n$.

Bernoulli Naive Bayes. With binary features, that is, $X \in \{0, 1\}^{n \times m}$, we choose the following conditional probability distributions parameterized by two non-negative vectors $\theta^+, \theta^- \in [0, 1]^m$. For a given vector $x \in \{0, 1\}^m$,

$$\mathbb{P}(x_j \mid C_\pm) = (\theta_j^\pm)^{x_j} (1 - \theta_j^\pm)^{1 - x_j}, \ \ j \in [m],$$

hence

$$\sum_{j=1}^m \log \mathbb{P}(x_j \mid C_\pm) = x^\top \log \theta^\pm + (\mathbf{1} - x)^\top \log(\mathbf{1} - \theta^\pm).$$

Training a classical Bernoulli naive Bayes model reduces to the problem

$$(\theta_*^+, \theta_*^-) = \arg \max_{\theta^+, \theta^- \in [0,1]^m} \mathcal{L}_{\text{bnb}}(\theta^+, \theta^-; X) \quad (2)$$

where the loss is a concave function

$$\begin{aligned}\mathcal{L}_{\text{bnb}}(\theta^+,\theta^-) = &\sum_{i\in\mathcal{I}_+}\log\mathbb{P}(x^{(i)}\mid C_+)\\ &+\sum_{i\in\mathcal{I}_-}\log\mathbb{P}(x^{(i)}\mid C_-) \qquad (3)\\ =&f^{+\top}\log\theta^+ + (n_+\mathbf{1}-f^+)^\top\log(\mathbf{1}-\theta^+)\\ &+f^{-\top}\log\theta^- + (n_-\mathbf{1}-f^-)^\top\log(\mathbf{1}-\theta^-)\end{aligned}$$

Note that problem (2) is decomposable across features and the optimal solution is simply the MLE estimate, that is, $\theta_*^\pm = f^\pm/n_\pm$. From (1), we get a linear classification rule: for a given test point $x\in\mathbb{R}^m$, we set $\hat{y}(x) = \mathbf{sign}(v + w_b^\top x)$, where

$$\begin{aligned}v &:= \log\frac{\mathbb{P}(C_+)}{\mathbb{P}(C_-)} + \mathbf{1}^\top\Big(\log(\mathbf{1}-\theta_*^+) - \log(\mathbf{1}-\theta_*^-)\Big)\\ w_b &:= \log\frac{\theta_*^+\circ(\mathbf{1}-\theta_*^-)}{\theta_*^-\circ(\mathbf{1}-\theta_*^+)}. \qquad (4)\end{aligned}$$

Multinomial naive Bayes. With integer-valued features, that is, $X\in\mathbb{N}^{n\times m}$, we choose the following conditional probability distribution, again parameterized by two non-negative m-vectors $\theta^\pm\in[0,1]^m$, but now with the constraints $\mathbf{1}^\top\theta^\pm = 1$: for given $x\in\mathbb{N}^m$,

$$\begin{aligned}\mathbb{P}(x\mid C_\pm) &= \frac{(\sum_{j=1}^m x_j)!}{\prod_{j=1}^m x_j!}\prod_{j=1}^m(\theta_j^\pm)^{x_j}\\ \Rightarrow\log\mathbb{P}(x\mid C_\pm) &= x^\top\log\theta^\pm + \log\left(\frac{(\sum_{j=1}^m x_j)!}{\prod_{j=1}^m x_j!}\right)\end{aligned}$$

While it is essential that the data be binary in the Bernoulli model seen above, the multinomial one can still be used if x is non-negative real-valued, and not integer-valued. Training the classical multinomial model reduces to the problem

$$\begin{aligned}(\theta_*^+,\theta_*^-) = \arg\max_{\theta^+,\theta^-\in[0,1]^m}\ &\mathcal{L}_{\text{mnb}}(\theta^+,\theta^-)\\ &\mathbf{1}^\top\theta^+ = \mathbf{1}^\top\theta^- = 1 \qquad (5)\end{aligned}$$

where the loss is a again a concave function

$$\begin{aligned}\mathcal{L}_{\text{mnb}}(\theta^+,\theta^-) = &\sum_{i\in\mathcal{I}_+}\log\mathbb{P}(x^{(i)}\mid C_+)\\ &+\sum_{i\in\mathcal{I}_-}\log\mathbb{P}(x^{(i)}\mid C_-)\\ =&f^{+\top}\log\theta^+ + f^{-\top}\log\theta^- \qquad (6)\end{aligned}$$

Again, problem (5) is decomposable across features, with the added complexity of equality constraints on $\theta^\pm$. The optimal solution is the MLE estimate $\theta_*^\pm = f^\pm/(\mathbf{1}^\top f^\pm)$. As before, we get a linear classification rule: for a given test point $x\in\mathbb{R}^m$, we set $\hat{y}(x) = \mathbf{sign}(v + w_m^\top x)$, where

$$v := \log\mathbb{P}(C_+) - \log\mathbb{P}(C_-),\ \ w_m := \log\theta_*^+ - \log\theta_*^- \qquad (7)$$

3 Naive Feature Selection

In this section, we incorporate sparsity constraints into the aforementioned models.

3.1 Naive Bayes with Sparsity Constraints

For a given integer $k\in[m]$, with $k<m$, we seek to obtain a naive Bayes classifier that uses at most k features in its decision rule. For this to happen, we need the corresponding coefficient vector, denoted w_b and w_m for the Bernoulli and multinomial cases, and defined in (4) and (7) respectively, to be k-sparse. For both Bernoulli and multinomial models, this happens if and only if the difference vector $\theta_*^+ - \theta_*^-$ is sparse. By enforcing k-sparsity on the difference vector, the classifier uses less than m features for classification, making the model more interpretable.

Sparse Bernoulli Naive Bayes. In the Bernoulli case, the sparsity-constrained problem becomes

$$\begin{aligned}(\theta_*^+,\theta_*^-) = \arg\max_{\theta^+,\theta^-\in[0,1]^m}\ &\mathcal{L}_{\text{bnb}}(\theta^+,\theta^-;X)\\ &\|\theta^+-\theta^-\|_0\le k \qquad \text{(SBNB)}\end{aligned}$$

where $\mathcal{L}_{\text{bnb}}$ is defined in (3). Here, $\|\cdot\|_0$ denotes the l_0-norm, or cardinality (number of non-zero entries) of its vector argument, and $k<m$ is the user-defined upper bound on the desired cardinality.

Sparse Multinomial Naive Bayes. In the multinomial case, in light of (5), our model is written

$$\begin{aligned}(\theta_*^+,\theta_*^-) = \arg\max_{\theta^+,\theta^-\in[0,1]^m}\ &\mathcal{L}_{\text{mnb}}(\theta^+,\theta^-;X)\\ &\mathbf{1}^\top\theta^+ = \mathbf{1}^\top\theta^- = 1\\ &\|\theta^+-\theta^-\|_0\le k \qquad \text{(SMNB)}\end{aligned}$$

where $\mathcal{L}_{\text{mnb}}$ is defined in (6).

3.2 Main Results

Due to the inherent combinatorial and non-convex nature of the cardinality constraint, and the fact that it couples the variables $\theta^\pm$, the above sparse training problems look much more challenging to solve when compared to their classical counterparts, (2) and (5). We will see in what follows that this is not the case.

Sparse Bernoulli Case. The sparse counterpart to the Bernoulli model, (SBNB), can be solved efficiently in *closed form*, with complexity comparable to that of the classical Bernoulli problem (2).

Theorem 1 (Sparse Bernoulli naive Bayes)**.** *Consider the sparse Bernoulli naive Bayes training problem* (SBNB)*, with binary data matrix $X \in \{0,1\}^{n\times m}$. The optimal values of the variables are obtained as follows. Set*

$$\begin{aligned} v &:= (f^+ + f^-) \circ \log\Big(\frac{f^+ + f^-}{n}\Big) \\ &\quad + (n\mathbf{1} - f^+ - f^-) \circ \log\Big(\mathbf{1} - \frac{f^+ + f^-}{n}\Big) \qquad (8)\\ w &:= w^+ + w^- \qquad (9)\\ w^\pm &:= f^\pm \circ \log\frac{f^+}{n_\pm} + (n_\pm \mathbf{1} - f^\pm) \circ \log\Big(\mathbf{1} - \frac{f^\pm}{n_\pm}\Big). \end{aligned}$$

Then identify a set $\mathcal{I}$ of indices with the k largest elements in $w - v$, and set θ_*^+, θ_*^- according to

$$\theta^+_{*_i} = \theta^-_{*_i} = \frac{1}{n}(f_i^+ + f_i^-),\ \forall i \in \mathcal{I}, \quad \theta^\pm_{*_i} = \frac{f_i^\pm}{n_\pm},\ \forall i \notin \mathcal{I}. \qquad (10)$$

Proof. For completeness we also include the following proof in Appendix A. First note that an ℓ_0-norm constraint on a m-vector q can be reformulated as

$$\|q\|_0 \le k \Longleftrightarrow \exists\, \mathcal{I} \subseteq [m],\ \ |\mathcal{I}| \le k\ :\ \forall\, i \notin \mathcal{I},\quad q_i = 0.$$

Hence problem (SBNB) is equivalent to

$$\begin{aligned} \max_{\theta^+,\theta^- \in [0,1]^m, \mathcal{I}} & \mathcal{L}_{\mathrm{bnb}}(\theta^+, \theta^-; X) \\ \text{s.t. } & \theta_i^+ = \theta_i^- \ \ \forall i \notin \mathcal{I},\ \ \mathcal{I} \subseteq [m],\ \ |\mathcal{I}| \le k \end{aligned} \qquad (11)$$

where the complement of the index set $\mathcal{I}$ encodes the indices where variables θ^+, θ^- agree. Then (11) becomes

$$p^* := \max_{\mathcal{I}\subseteq[m],\ |\mathcal{I}|\le k} \ \Big(\sum_{i\notin\mathcal{I}} h_i^\pm\Big) + \Big(\sum_{i\in\mathcal{I}} h_i^+ + h_i^-\Big) \qquad (12)$$

where

$$\begin{aligned} h_i^\pm &= \max_{\theta_i\in[0,1]} (f_i^+ + f_i^-)\log\theta_i + (n - f_i^+ - f_i^-)\log(1-\theta_i)\\ h_i^+ &= \max_{\theta_i^+\in[0,1]} f_i^+ \log\theta_i^+ + (n_+ - f_i^+)\log(1-\theta_i^+)\\ h_i^- &= \max_{\theta_i^-\in[0,1]} f_i^- \log\theta_i^- + (n_- - f_i^-)\log(1-\theta_i^-) \end{aligned}$$

and where we use the fact that $n_+ + n_- = n$. All the above expressions for $h_i^\pm, h_i^+, h_i^-$ have closed form values and solutions

$$\begin{aligned} \theta_i = \theta^+_{*\,i} = \theta^-_{*\,i} &= \frac{1}{n}(f_i^+ + f_i^-),\ \ \forall i \notin \mathcal{I}\\ \theta^\pm_{*_i} &= \frac{f_i^\pm}{n_\pm},\ \ \forall i \in \mathcal{I} \end{aligned} \qquad (13)$$

Plugging the above inside the objective of (11) results in a Boolean formulation, with a Boolean vector u of cardinality $\le k$ such that $\mathbf{1} - u$ encodes indices for which entries of θ^+, θ^- agree:

$$p^* := \max_{u\in\mathcal{C}_k} (\mathbf{1} - u)^\top v + u^\top w,$$

where, for $k \in [m]$:

$$\mathcal{C}_k := \{u\ :\ u \in \{0,1\}^m,\ \mathbf{1}^\top u \le k\},$$

and vectors v, w are as defined in (8):

$$\begin{aligned} v &:= (f^+ + f^-) \circ \log\Big(\frac{f^+ + f^-}{n}\Big)\\ &\quad + (n\mathbf{1} - f^+ - f^-) \circ \log\Big(\mathbf{1} - \frac{f^+ + f^-}{n}\Big)\\ w &:= w^+ + w^-\\ w^\pm &:= f^\pm \circ \log\frac{f^+}{n_\pm} + (n_\pm \mathbf{1} - f^\pm) \circ \log\Big(\mathbf{1} - \frac{f^\pm}{n_\pm}\Big) \end{aligned}$$

We obtain

$$p^* = \mathbf{1}^\top v + \max_{u\in\mathcal{C}_k}\ u^\top(w - v) = \mathbf{1}^\top v + s_k(w - v),$$

where $s_k(\cdot)$ denotes the sum of the k largest elements in its vector argument. Here we have exploited the fact that the map $z := w - v \ge 0$, which in turn implies that

$$s_k(z) = \max_{u\in\{0,1\}^m\,:\,\mathbf{1}^\top u = k}\ u^\top z = \max_{u\in\mathcal{C}_k}\ u^\top z.$$

In order to recover an optimal pair (θ_*^+, θ_*^-), we simply identify the set $\mathcal{I}$ of indices with the $m - k$ smallest elements in $w - v$, and set θ_*^+, θ_*^- according to (20). ■

Note that the complexity of the computation (including forming the vectors $f^\pm$, and finding the k largest elements in the appropriate m-vector) grows as $O(mn\log(k))$. This represents a very moderate extra cost compared to the cost of the classical naive Bayes problem, which is $O(mn)$.

Multinomial Case. In the multinomial case, the sparse problem (SMNB) does not admit a closed-form solution. However, we can obtain an upper bound.

Theorem 2 (Sparse multinomial naive Bayes)**.** *Let $\phi(k)$ be the optimal value of* (SMNB)*. Then $\phi(k) \le \psi(k)$, where $\psi(k)$ is the optimal value of the following one-dimensional convex optimization problem*

$$\psi(k) := C + \min_{\alpha\in[0,1]}\ s_k(h(\alpha)), \qquad \text{(USMNB)}$$

where C is a constant, $s_k(\cdot)$ is the sum of the top k entries of its vector argument, and for $\alpha \in (0,1)$,

$$h(\alpha) = \tilde{C} - f^+\log\alpha - f^-\log(1-\alpha).$$

where $\tilde{C} = f^+ \circ \log f^+ + f^- \circ \log f^- - (f^+ + f^-) \circ \log(f^+ + f^-)$. *Furthermore, given an optimal dual variable* α_* *that solves* (USMNB), *we can reconstruct a primal feasible (sub-optimal) point* (θ^+, θ^-) *for* (SMNB) *as follows. For* α^* *optimal for* (USMNB), *let* $\mathcal{I}$ *be complement of the set of indices corresponding to the top* k *entries of* $h(\alpha_*)$; *then set* $B_\pm := \sum_{i \notin \mathcal{I}} f_i^\pm$, *and*

$$\theta^+_{*\,i} = \theta^-_{*\,i} = \frac{f_i^+ + f_i^-}{\mathbf{1}^\top (f^+ + f^-)}, \ \forall i \in \mathcal{I}$$
$$\theta^\pm_{*i} = \frac{B_+ + B_-}{B_\pm} \frac{f_i^\pm}{\mathbf{1}^\top (f^+ + f^-)}, \ \forall i \notin \mathcal{I} \tag{14}$$

Proof. See Appendix B. ■

The key point here is that, while problem (SMNB) is nonconvex and potentially hard, the dual problem is a one-dimensional convex optimization problem which can be solved very efficiently, using bisection. The number of iterations to localize an optimal α^* with absolute accuracy ϵ grows slowly, as $O(\log(1/\epsilon))$; each step involves the evaluation of a sub-gradient of the objective function, which requires finding the k largest elements in a m-vector, and costs $O(m \log k)$. As before in the Bernoulli case, the complexity of the sparse variant in the multinomial case is $O(mn \log k)$, versus $O(mn)$ for the classical naive Bayes.

Quality estimate. The quality of the bound in the multinomial case can be analysed using bounds on the duality gap based on the Shapley-Folkman theorem.

Theorem 3 (Quality of Sparse Multinomial Naive Bayes Relaxation)**.** *Let* $\phi(k)$ *be the optimal value of* (SMNB) *and* $\psi(k)$ *that of the convex relaxation in* (USMNB), *we have, for* $k \geq 4$,

$$\psi(k-4) \leq \phi(k) \leq \psi(k) \leq \phi(k+4). \tag{15}$$

Proof. See Appendix C. ■

While we defer details of the proof of Theorem 3 to the Appendix, we provide a high level discussion of how to bound the duality gap. The proof follows from results by (Aubin and Ekeland, 1976) (see (Ekeland and Temam, 1999; Kerdreux et al., 2017) for a more recent discussion) which are summarized below. Given functions f_i, a vector $b \in \mathbb{R}^m$, and vector-valued functions g_i, $i \in [n]$ that take values in $\mathbb{R}^m$, we consider the following problem:

$$\mathrm{h}_P(u) := \min_x \sum_{i=1}^n f_i(x_i) \ : \ \sum_{i=1}^n g_i(x_i) \leq b + u \tag{P}$$

in the variables $x_i \in \mathbb{R}^{d_i}$, with perturbation parameter $u \in \mathbb{R}^m$. Let $\mathrm{h}_P(u)^{**}$ be the biconjugate of $\mathrm{h}_P(u)$ defined in (P), then $\mathrm{h}_P(0)^{**}$ is the optimal value of the dual to (P) (Ekeland and Temam, 1999, Lem. 2.3), and (Ekeland and Temam, 1999, Th. I.3) shows the following result.

Theorem 4. *Suppose the functions* f_i, g_{ji} *in problem* (P) *are proper, 1-coercive, lower semicontinuous and there exists affine minorants for* $i = 1, \ldots, n$, $j = 1, \ldots, m$. *Let*

$$\bar{p}_j = (m+1) \max_i \rho(g_{ji}), \quad \text{for } j = 1, \ldots, m \tag{16}$$

then

$$\mathrm{h}_P(\bar{p}) \leq \mathrm{h}_P(0)^{**} + (m+1) \max_i \rho(f_i). \tag{17}$$

where $\rho(f) \triangleq \sup_{x \in \mathbf{dom}(f)} \{f(x) - f^{**}(x)\}$.

Hence by bounding the non-convexity of the ℓ_0 constraint, we are able to bound the overall duality gap.

The bound in Theorem 3 implies in particular

$$\psi(k-4) \leq \phi(k) \leq \psi(k-4) + \Delta(k), \text{ for } k \geq 4,$$

where $\Delta(k) := \psi(k) - \psi(k-4)$. This means that if $\psi(k)$ does not vary too fast with k, so that $\Delta(k)$ is small, then the duality gap in problem (SMNB) is itself small, bounded by $\Delta(k)$; then solving the convex problem (USMNB) will yield a good approximate solution to (SMNB). This means that when the marginal contribution of additional features, i.e. $\Delta(k)/\psi(k)$ becomes small, our bound becomes increasingly tight. The "elbow heuristic" is often used to infer the number of relevant features k^*, with $\psi(k)$ increasing fast when $k < k^*$ and much more slowly when $k \geq k^*$. In this scenario, our bound becomes tight for $k \geq k^*$.

4 Experiments

In this section, we compare our sparse multinomial model (SMNB) against other feature selection methods (Experiments 1-3) we empirically show the quality of our relaxation on a synthethic dataset (Experiment 4). For the former experiments, we do not use deep learning methods since we want to compare the features selected rather than the end-to-end training accuracy. For this reason, we compare (SMNB) against traditional ℓ_1 methods, recursive feature elimination (RFE) methods, and other sparsity-inducing methods.

4.1 Experiment 1: Feature Selection

In the next three experiments, we compare (SMNB) with other feature selection methods for sentiment classification on five different text data sets. Some details on the data sets sizes are given in Table 1. More

Feature Vectors	Amazon	IMDB	Twitter	MPQA	SST2
Count Vector	31,666	103,124	273,779	6,208	16,599
tf-idf	5000	5000	5000	5000	5000
tf-idf wrd bigram	5000	5000	5000	5000	5000
tf-idf char bigram	5000	5000	5000	4838	5000
n_{train}	8000	25,000	1,600,000	8484	76,961
n_{test}	2000	25,000	498	2122	1821

Table 1: **Experiment 1 data**: Number of features for each type of feature vector for each data set. For tf-idf feature vectors, we fix the maximum number of features to 5000 for all data sets. The last two rows show the number of training and test samples.

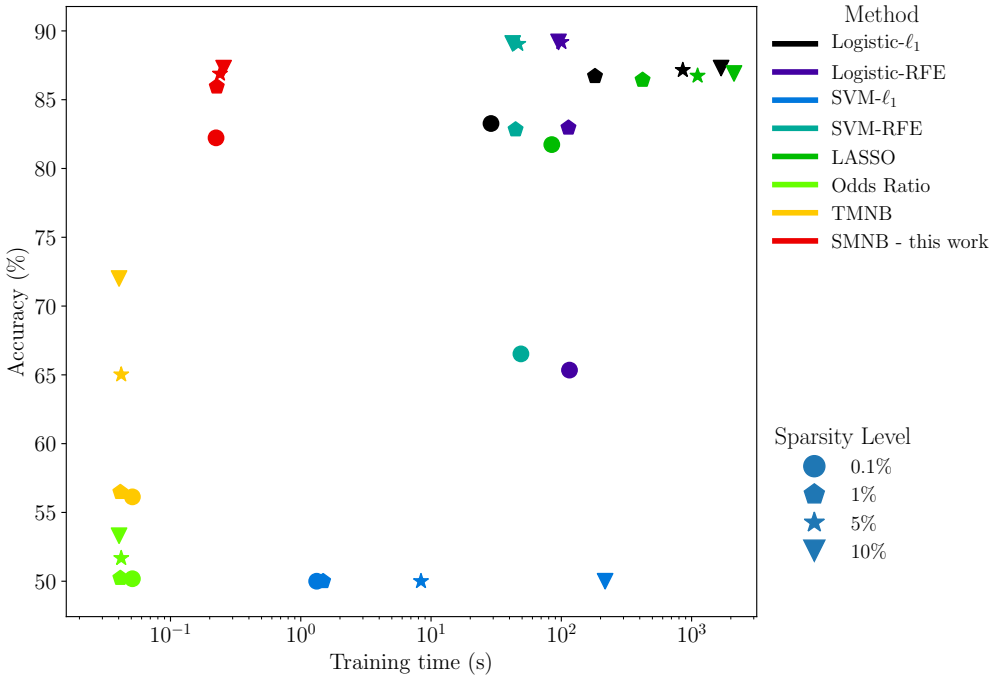

Figure 1: **Experiment 1:** Accuracy versus run time with the IMDB dataset/Count Vector with MNB in stage 2, showing performance on par with the best feature selection methods, at fraction of computing cost. Times *do not* include the cost of grid search to reach the target cardinality for ℓ_1-based methods. For more details on the experiment, see Appendix D.

information on these data sets and how they were preprocessed are given in Appendix D.

For each data set and each type of feature vector, we perform the following two-stage procedure. In the first step, we employ a feature selection method to attain a desired sparsity level of (0.1%, 1%, 5%, 10%); in the second step, we train a classifier based on the selected features. Specifically, we use ℓ_1-regularized logistic regression, logistic regression with recursive feature elimination (RFE), ℓ_1-regularized support vector machine (SVM), SVM with RFE, LASSO, thresholded Multinomial naive Bayes (TMNB), the Odds Ratio metric described by Mladenic and Grobelnik (1999) and (SMNB) in the first step. Then using the selected features, in the second step we train a logistic model, a MNB model, and a SVM. Thresholded multinomial naive Bayes (TMNB) means we train a multinomial naive Bayes model and then select the features corresponding to indices of the largest absolute value entries of the vector of classification coefficients w_m, as defined in (7). For each desired sparsity level and each data set

Feature Vectors	Amazon	IMDB	Twitter	MPQA	SST2
Count Vector	31,666	103,124	273,779	6,208	16,599
tf-idf	31,666	103,124	273,779	6,208	16,599
tf-idf wrd bigram	870,536	8,950,169	12,082,555	27,603	227,012
tf-idf char bigram	25,019	48,420	17,812	4838	7762

Table 2: **Experiment 2 data**: Number of features for each type of feature vector for each data set with no limit on the number of features for the tf-idf vectors. The train/test split is the same as in Table 1.

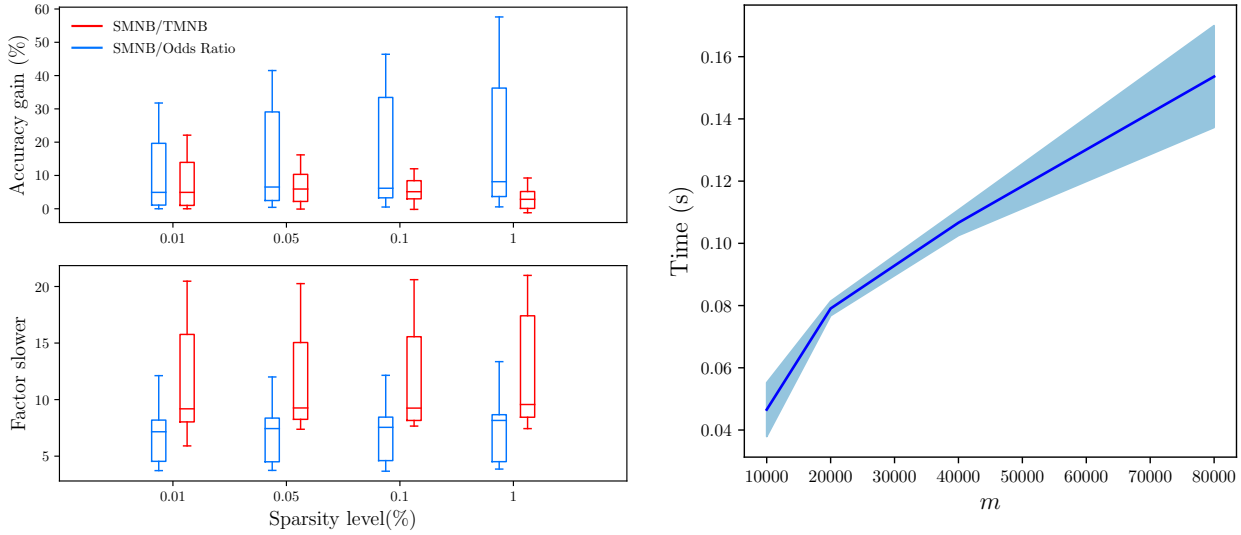

Figure 2: **Experiment 2 (Left):** Accuracy gain for our method (top panel) and factor slower (bottom panel) over all data sets listed in Table 2 with MNB in stage 2, showing substantial performance increase with a constant increase in computational cost. **Experiment 3 (Right):** Run time with IMDB dataset/tf-idf vector data set, with increasing m, k with fixed ratio k/m, empirically showing (sub-) linear complexity.

in the first step, we do a grid search over the optimal Laplace smoothing parameter for MNB for each type of feature vector. We use this same parameter in (SMNB). All models were implemented using Scikit-learn (Pedregosa et al., 2011). Figure 1 shows that (SMNB) is competitive with other feature selection methods, consistently maintaining a high test set accuracy, while only taking a fraction of the time to train; for a sparsity level of 5%, a logistic regression model with ℓ_1 penalty takes more than 1000 times longer to train.

4.2 Experiment 2: large-scale feature selection

For this experiment, we consider the same data sets as before, but do not put any limit on the number of features for the tf-idf vectors. Due to the large size of the data sets, most of the feature selection methods in Experiment 1 are not feasible. We use the same two-stage procedure as before: 1) do feature selection using TMNB, the Odds Ratio method and our method (USMNB), and 2) train a MNB model using the features selected in stage 1. We tune the hyperparameters for MNB and (USMNB) the same way as in Experiment 2. In this experiment, we focus on sparsity levels of $0.01\%, 0.05\%, 0.1\%, 1\%$. Table 2 summarizes the data used in Experiment 2 and in Table 3 we display the average training time for (USMNB).

Figure 2 shows that, even for large datasets with millions of features and data points, our method, implemented on a standard CPU with a non-optimized solver, takes at most a few seconds, while providing a significant improvement in performance. See Appendix D for the accuracy versus sparsity plot for each data set and each type of feature vector.

4.3 Experiment 3: complexity

Using the IMDB dataset in Table 1, we perform the following experiment: we fix a sparsity pattern $k/m =$

	Amazon	IMDB	Twitter	MPQA	SST2
F_C	0.043	0.22	1.15	0.0082	0.037
F_{t_1}	0.033	0.16	0.89	0.0080	0.027
F_{t_2}	0.68	9.38	13.25	0.024	0.21
F_{t_3}	0.076	0.47	4.07	0.0084	0.082

Table 3: **Experiment 2 run times**: Average run time (in seconds, with a standard CPU and a non-optimized implementation) over $4 \times 30 = 120$ values for different sparsity levels and 30 randomized train/test splits per sparsity level for each data set and each type of feature vector. On the largest data set (Twitter, $\sim$ 12M features, $\sim$ 1.6M data points), the computation takes less than 15 seconds. For the full distribution of run times, see Appendix D. $F_C, F_{t_1}, F_{t_2}, F_{t_3}$ refer to the count vector, tf-idf, tf-idf word bigram, and tf-idf character bigram feature vectors respectively.

0.05 and then increase k and m. Where we artificially set the number of tf-idf features to 5000 in Experiment 1, here we let the number of tf-idf features vary from $10,000$ to $80,000$. We then plot the the time it takes to train (SMNB) at a the fixed 5% sparsity level. Figure 2 shows that for a fixed sparsity level, the complexity of our method appears to be sub-linear.

4.4 Experiment 4: Duality Gap

In this experiment, we generate random synthetic data with uniform independent entries: $f^{\pm} \sim U[0,1]^m$, where $m \in \{30, 3000\}$. We then normalize $f^{\pm}$ and compute $\psi(k)$ and $\psi(k-4)$ for $4 \leq k \leq m$ and plot how this gap evolves as k increases. For each value of k, we also plot the value of the reconstructed primal feasible point, as detailed in Theorem 2. The latter serves as a lower bound on the true value $\phi(k)$, which can be used to test *a posteriori* if our bound is accurate.

Figure 3 shows that, as the number of features m or the sparsity parameter k increases, the duality gap bound decreases. Figure 3 also shows that the *a posteriori* gap is almost always zero, implying strong duality. In particular, as shown in Figure 3(b), as the number of features increases, the gap between the bounds and the primal feasible point's value becomes negligible for all values of k. This indicates that we can solve the original, non-convex problem (SBNB) by instead solving a 1-dimensional dual problem and constructing a primal feasible solution in closed form.

5 Conclusion

In this paper, we propose a sparse version of naive Bayes, leading to a combinatorial maximum likelihood

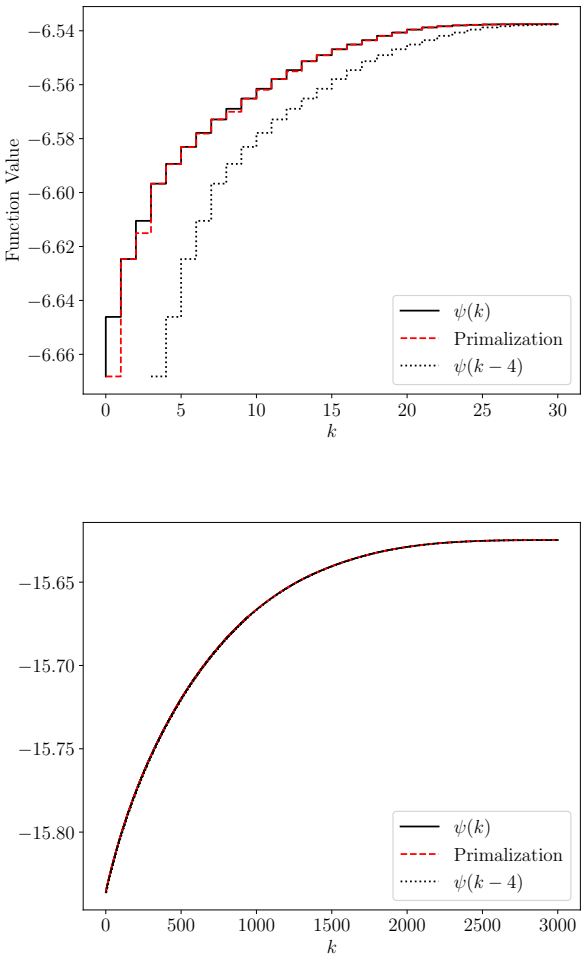

Figure 3: **Experiment 4:** Duality gap bound versus sparsity level for $m = 30$ (top panel) and $m = 3000$ (bottom panel), showing that the duality gap quickly closes as m or k increase.

problem that we show is more benign than it appears. In the case of binary data, we are able to solve the problem exactly, while in the multinomial case, we provide explicit bounds on the duality gap and show it decreases as as the marginal contribution of additional features decreases. Furthermore, we show empirically on synthetic data that this bound is quite loose and that our scheme appears to be tight (ie. strong duality holds). We test our method on different text data sets with other popular feature selection methods. On all the data sets, we are able to maintain the same performance on the test set while only taking a fraction of the time to train (in some cases our method is 1000x faster than other methods with specialized solvers).

Acknowledgements

A.A. is at the département d'informatique de l'ENS, École normale supérieure, UMR CNRS 8548, PSL Research University, 75005 Paris, France, and INRIA Sierra project-team. AA would like to acknowledge support from the *ML and Optimisation* joint research initiative with the *fonds AXA pour la recherche* and Kamet Ventures, a Google focused award, as well as funding by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). LEG would like to acknowledge support from Berkeley Artificial Intelligence Research (BAIR) and Tsinghua-Berkeley-Shenzhen Institute (TBSI).

References

Aubin, J.-P. and Ekeland, I. (1976). Estimates of the duality gap in nonconvex optimization. *Mathematics of Operations Research*, 1(3):225–245.

Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.

Boullé, M. (2007). Compression-based averaging of selective naive bayes classifiers. *Journal of Machine Learning Research*, 8(Jul):1659–1685.

Ekeland, I. and Temam, R. (1999). *Convex analysis and variational problems*. SIAM.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine learning research*, 5(Nov):1531–1555.

Frank, E., Hall, M., and Pfahringer, B. (2002). Locally weighted naive Bayes. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 249–256. Morgan Kaufmann Publishers Inc.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3):131–163.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (1993). *Convex Analysis and Minimization Algorithms*. Springer.

Jiang, L., Wang, D., Cai, Z., and Yan, X. (2007). Survey of improving naive Bayes for classification. In *International Conference on Advanced Data Mining and Applications*, pages 134–145. Springer.

Kerdreux, T., Colin, I., and d'Aspremont, A. (2017). An approximate shapley-folkman theorem. *arXiv preprint arXiv:1712.08559*.

Kim, S.-B., Han, K.-S., Rim, H.-C., and Myaeng, S. H. (2006). Some effective techniques for naive Bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466.

Mladenic, D. and Grobelnik, M. (1999). Feature selection for unbalanced class distribution and naive Bayes. In *ICML*, volume 99, pages 258–267.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press., Princeton.

Starr, R. M. (1969). Quasi-equilibria in markets with non-convex preferences. *Econometrica: journal of the Econometric Society*, pages 25–38.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Webb, G. I., Boughton, J. R., and Wang, Z. (2005). Not so naive Bayes: aggregating one-dependence estimators. *Machine learning*, 58(1):5–24.

Zaidi, N. A., Cerquides, J., Carman, M. J., and Webb, G. I. (2013). Alleviating naive bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, 14(1):1947–1988.

Zheng, Z., Cai, Y., Yang, Y., and Li, Y. (2018). Sparse weighted naive Bayes classifier for efficient classification of categorical data. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 691–696. IEEE.

Zheng, Z. and Webb, G. I. (2000). Lazy learning of Bayesian rules. *Machine Learning*, 41(1):53–84.