Bayesian Regularized Iterative Soft Thresholding Algorithm

Nicolas Cutrona[0000−0001−7673−1422] and Dominique Guillot[0000−0002−8589−8061]

University of Delaware, Newark DE 19711, USA
{ncutrona,dguillot}@udel.edu

Abstract. Weighted Naive Bayes methods have recently been developed to alleviate the strong conditional independence assumption of traditional Naive Bayes classifiers. In particular, class-specific attribute weighted Naive Bayes (CAWNB) has been shown to yield excellent performance on many modern datasets. Such methods, however, are prone to over-fitting on small sample, large feature space data. In this work, we propose a Bayesian Regularized Iterative Shrinkage-Thresholding Algorithm (`BARISTA`), which includes both ℓ_1 and ℓ_2 regularization to mitigate this problem. As we show, estimating the parameters of `BARISTA` via maximum likelihood yields a convex objective that can be efficiently optimized using Iterative Shrinkage-Thresholding Algorithms (ISTA). We prove the resulting method has many attractive theoretical and numerical properties, including a guaranteed linear rate of convergence. Using several standard benchmark datasets, we demonstrate how `BARISTA` can yield a significant increase in performance compared to many state-of-the-art weighted Naive Bayes methods. We also show how the Fast Iterative-Shrinkage Thresholding Algorithm (FISTA) can be used to further accelerate convergence.[1]

Keywords: Naive Bayes · Convex Optimization · Soft Thresholding.

1 Introduction

Naive Bayes (NB) is a popular white-box probabilistic classification algorithm that is used in a variety of machine learning applications. This method can produce accurate and interpretable results along with the ability to quantify the uncertainty of classifications. Given a set of m predictor attributes, $\{X_1, \dots, X_m\}$, and a class attribute, C, Naive Bayes efficiently estimates the joint distribution of the data as

$$P(C, X_1, X_2, \dots, X_m) = P(C) \cdot P(X_1|C) \cdot P(X_2|C) \cdot \dots \cdot P(X_m|C), \tag{1}$$

where $P(\cdot) \in [0, 1]$ is a probability measure. Using this joint distribution, the Naive Bayes estimator for the class attribute of a sample $\mathbf{x} = (x_1, \dots, x_m)$ is

[1] Our code and data are publicly available on this repository.

given by

$$\underset{c\in\mathcal{C}}{\operatorname{argmax}}\ P(C=c|X=\mathbf{x}) = \frac{P(c)\prod_{j=1}^{m} P(x_j|c)}{\sum_{c\in\mathcal{C}} P(c)\prod_{j=1}^{m} P(x_j|c)}, \tag{2}$$

where $P(c)$ is the prior probability, $P(x_j|c)$ is the likelihood of observing attribute x_j given the class, $P(c|\mathbf{x})$ is the posterior probability, and $\mathcal{C}$ denotes the set of possible attribute classes. Computing the posterior probability as in NB requires the assumption of conditional independence amongst the predictors. More formally, for any $i \neq j$, $X_i \perp X_j|C$, where the notation $\perp$ is used to denote independence. When this assumption does not hold, which is often the case for real world data, misclassifications become more prominent as the posterior probabilities can be severely biased [9]. This hindrance has motivated a variety of works aimed at offering techniques to mitigate the performance loss of NB, while preserving the simplicity and interpretability of the approach. Most notably, variations of attribute weighting have yielded the strongest results.

Attribute weighting techniques are used to boost the discriminative power of Naive Bayes to increase classification accuracy. The most effective weighting methods learn optimal weights through an iterative optimization procedure to minimize a given loss function. In the attribute weighted Naive Bayes (WANBIA) framework, a weight is assigned to every likelihood, irrespective of the class [13]. Zhang *et al.* extended this idea by introducing a more complex class-specific attribute weighting Naive Bayes (CAWNB) approach [6]. This procedure assigns a weight to each likelihood with respect to each class c and each attribute x_j to capture class dependencies in the data. More specifically, let $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a sample of observations, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{im}) \in \mathcal{C}^m$. The estimated posterior probability of class $c \in \mathcal{C}$ given $\mathbf{x} = (x_1, \ldots, x_m)$ is given by

$$\widehat{P}(c|\mathbf{x}) = \frac{\pi_c \prod_{j=1}^{m} \theta_{c,j,\mathbf{x}}^{w_{c,j}}}{\sum_{c'=1}^{l} \pi_{c'} \prod_{j=1}^{m} \theta_{c',j,\mathbf{x}}^{w_{c',j}}}, \tag{3}$$

where $\pi = (\pi_1, \pi_2, \ldots, \pi_l)$ are the prior probabilities and where each π_c is the prior probability that a sample, $\mathbf{x}$, belongs to a class $c \in \mathcal{C}$. The $\theta_{c,j,\mathbf{x}}^{w_{c,j}}$ are the weighted likelihoods with respect to the j^{th} attribute and class c, and l is the number of possible class values. Each likelihood is assigned a weight $w_{c,j}$ that is class-dependent [6]. The prior probabilities and the likelihoods are estimated from the data as follows

$$\pi_c = \frac{\sum_{i=1}^{n} \delta(c_i, c) + \frac{1}{l}}{n+1}, \qquad \theta_{c,j,\mathbf{x}} = \frac{\sum_{i=1}^{n} \delta(x_{ij}, x_j)\delta(c_i, c) + \frac{1}{n_j}}{\sum_{i=1}^{n} \delta(c_i, c) + 1}, \tag{4}$$

where $\delta(\cdot, \cdot)$ is an indicator function that returns 1 if the inputs are the same and 0 otherwise. The $\frac{1}{l}$ and $\frac{1}{n_j}$ terms are smoothing constants to handle numerical instability during model learning, where n_j is the number of possible attribute values for the j^{th} attribute.

The CAWNB framework, along with a multitude of other work regarding attribute weighting methods, use the MSE loss function for classification feedback to iteratively find an optimal set of weights. In this setting, the MSE function is:

$$L(\mathbf{W}) = \frac{1}{2} \sum_{\mathbf{x}_i \in D} \sum_{c \in \mathcal{C}} (P(c|\mathbf{x}_i) - \widehat{P}(c|\mathbf{x}_i))^2, \tag{5}$$

where $\mathbf{W} = (w_{c,j})$ is the matrix of class-specific attribute weights, and $P(c|\mathbf{x}_i)$ is the ground truth of sample $\mathbf{x}_i$, given by

$$P(c|\mathbf{x}_i) = \begin{cases} 1 & \text{if } c = c_i \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where c_i denotes the class attribute of sample $\mathbf{x}_i$. Learning class-dependent weights significantly increases the complexity of this framework in comparison to simple attribute weighting methods. To handle over-fitting on small sample, large feature space data, Wang *et al.* propose to regularize the CAWNB framework by using the simpler WANBIA model as a constraint [9]. While their method does yield an increase in performance over CAWNB, Wang *et al.* state that ℓ_1 or ℓ_2 regularization would be a more robust approach, especially to handle noise and outliers in the data. Furthermore, they suggest that implementing these techniques would impose a severe computational burden.

In this paper, we show that incorporating ℓ_1 and ℓ_2 regularization within the CAWNB framework significantly increases performance and can be achieved with little impact on computational complexity. Moreover, in contrast to most previous work in the literature, we investigate the properties of the objective function to optimize and derive theoretical convergence guarantees of our algorithm. In particular, we first replace the MSE loss function (5) by a negative log-likelihood. With a careful analysis of the associated Hessian, we prove that the resulting objective function is convex, contrary to the MSE used in previous work. We then exhibit how the negative log-likelihood with added ℓ_1 and ℓ_2 penalties can be efficiently minimized by leveraging iterative shrinkage-thresholding algorithms (ISTA). We call the resulting approach `BARISTA`. While ISTA and FISTA are well known optimization techniques in the literature, verifying the assumptions of these methods to justify their use in `BARISTA` requires several non-trivial calculations, which constitutes the main contribution of our work. This analysis allows us to derive theoretical results to guarantee the linear convergence of `BARISTA` to its optimal value. We also illustrate the linear convergence of our algorithm in concrete experiments. Finally, we compare the performance of `BARISTA` with several state-of-the-art weighted Naive Bayes methods on eight classical real world datasets from the UCI machine learning repository that were previously used in the weighted Naive Bayes literature. Our results demonstrate how `BARISTA` can yield a significant increase in performance compared to competing methods, while maintaining computational efficiency and interpretability.

The rest of this paper is organized as follows. Section 2 describes prior work on attribute weighting techniques for Naive Bayes. The BARISTA method is formulated in Section 3, along with the necessary results to show convexity of the negative log-likelihood function. Section 4 details the theoretical convergence results of BARISTA. Performance results are presented in Section 5, and concluding remarks are made in Section 6.

2 Prior Work

Naive Bayes performs well on many machine learning and data mining tasks. However, the strong conditional independence assumption often hinders its classification capabilities in real world applications. There is thus interest in developing new robust Naive Bayes variations [13]. Attribute weighting is a popular strategy that has been shown to work well in practice. In this work, we focus on wrapper-based methods [9].

Wrapper-based methods [5,6,8,9,11,13,14,15] iteratively optimize over attribute weights to find an acceptable solution as opposed to non-iterative filter-based methods. Because wrapper-based methods are iterative in nature, they are computationally demanding but often yield stronger results [9]. Taheri *et al.* design an attribute weighted NB framework that learns optimal attribute weights by a local optimization quasi-secant method [8]. Zheng *et al.* also make use of attribute weighting by utilizing conjugate gradient descent and implementing an ℓ_1 penalty to regularize attribute weights [15]. In [13], Zaidi *et al.* give a detailed overview of different attribute weighting schemes for NB, as well as comparisons to other data mining methods. Works [8,13,15] represent simple attribute weighting for NB (WANBIA) [13]. Class-specific attribute weighted NB is a more discriminative version of WANBIA. CAWNB determines an optimal weight for an attribute with respect to each class to learn deeper patterns in the data, as opposed to disregarding class dependencies [9].

Jiang *et al.* argue that optimizing over class specific attribute weights should boost performance of wrapper-based methods [6]. While this has shown to be an accurate assessment, Wang *et al.* state that CAWNB methods introduce more computational overhead than previously discussed weighting methods. Wang *et al.* propose a regularized attribute weighted framework for NB (RNB) that is designed to handle the over-fitting issue that can occur using the CAWNB approach [9]. While the RNB method provides excellent performance on benchmark datasets, Wang *et al.* speculate that ℓ_1 or ℓ_2 regularization could further improve performance; however, they suggest implementing such a scheme would drastically increase the computational complexity of the learning algorithm.

3 Methodology

We now provide the theory and methodology of BARISTA. We begin by addressing the non-convexity of the MSE loss function. We then discuss how to optimize the penalized negative log-likelihood.

3.1 Convexity Analysis and Optimization

It is not difficult to show that, in general, the MSE (5) is not convex under the CAWNB framework described in (3). To yield an easier optimization problem and guarantee the uniqueness of its solution, we estimate the weights $w_{c,j}$ in (3) via maximum likelihood. Let $\widehat{P}(c|\mathbf{x})$ be given as in (3). The negative log-likelihood function associated to this problem is

$$\begin{aligned} f(\mathbf{W}) &= -\sum_{i=1}^{n} \log\left(\widehat{P}(c_i|\mathbf{x}_i)\right) &(7)\\ &= -\sum_{i=1}^{n}\left(\log\left(\pi_{c_i}\prod_{j=1}^{m}\theta_{c_i,j,\mathbf{x}_i}^{w_{c_i,j}}\right) - \log\left(\sum_{c'=1}^{l}\pi_{c'}\prod_{j=1}^{m}\theta_{c',j,\mathbf{x}_i}^{w_{c',j}}\right)\right), \end{aligned}$$

where $\mathbf{W} = (w_{c,j}) \in \mathbb{R}^{l\times m}$ is the matrix of class specific attribute weights, m is the number of attributes in the data, and the log given is the natural logarithm.

Theorem 1. *The negative log-likelihood function* (7) *is convex.*

Proof. In order to prove Theorem 1, we compute the Hessian matrix of f. The calculations are non-trivial and are broken down into several lemmas. We direct readers to the supplementary material[2] for the details.

Instead of directly optimizing (7), we append ℓ_1 and ℓ_2 regularization terms to penalize the weights to form the following loss function

$$F(\mathbf{W}) = -\sum_{i=1}^{n}\log\left(\widehat{P}(c_i|\mathbf{x}_i)\right) + \rho_2||\mathbf{W}||_2^2 + \rho_1||\mathbf{W}||_1, \quad (8)$$

where $\rho_1, \rho_2 > 0$ are penalization constants. In general, ℓ_1 regularization can be used for attribute selection by setting parameters associated with irrelevant attributes to zero. The ℓ_2 regularization is used to achieve better numerical stability, decreased parameter variance, and a better conditioned design matrix. In the same spirit as the Elastic Net used in the context of regression [16], using ℓ_1 and ℓ_2 regularization together allows `BARISTA` to produce accurate, and if necessary, sparse solutions. Because the ℓ_1 regularization term is non-differentiable, conventional first order optimization techniques such as gradient descent cannot be used in this framework. In Section 3.1, we proved that the negative log-likelihood function is convex (Theorem 1). This allows us to handle the non-differentiable ℓ_1 term using an iterative shrinkage-thresholding algorithm (ISTA), a special case of proximal gradient descent.

Proximal gradient descent is an efficient first-order technique for solving problems of the form

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}}\ F(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}), \quad (9)$$

[2] Please visit https://github.com/ncutrona/BARISTA to view the supplementary material.

where $\mathcal{X}$ is a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and corresponding norm $|| \cdot ||$, $g : \mathcal{X} \to \mathbb{R}$ is a continuously differentiable convex function, and $h : \mathcal{X} \to \mathbb{R}$ is a lower semi-continuous, convex, and not necessarily smooth function [7]. The proximal operator associated to h, denoted by $\text{prox}_h : \mathcal{X} \to \mathcal{X}$, is given by

$$\text{prox}_h(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{X}}{\text{argmin}} \frac{1}{2}||\mathbf{x} - \mathbf{y}||^2 + h(\mathbf{y}). \tag{10}$$

If h were differentiable, we could simply use the gradient descent update $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \alpha \nabla F(\mathbf{x}^{(i)})$ to minimize F, where $\alpha > 0$ is a given step size. This update minimizes the following quadratic approximation of F at $\mathbf{x}^{(i)}$

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{y}}{\text{argmin}}\ F(\mathbf{x}^{(i)}) + \nabla F(\mathbf{x}^{(i)})(\mathbf{y} - \mathbf{x}^{(i)}) + \frac{1}{2\alpha}\|\mathbf{x}^{(i)} - \mathbf{y}\|^2. \tag{11}$$

When h is not differentiable, the proximal gradient descent proceeds in a similar manner by solving

$$\begin{aligned}
\mathbf{x}^{(i+1)} &= \underset{\mathbf{y}}{\text{argmin}}\ g(\mathbf{x}^{(i)}) + \nabla g(\mathbf{x}^{(i)})(\mathbf{y} - \mathbf{x}^{(i)}) + \frac{1}{2\alpha}\|\mathbf{x}^{(i)} - \mathbf{y}\|^2 + h(\mathbf{y}) \\
&= \underset{\mathbf{y}}{\text{argmin}}\ \frac{1}{2\alpha}||\mathbf{y} - (\mathbf{x}^{(i)} - \alpha\nabla g(\mathbf{x}^{(i)}))||^2 + h(\mathbf{y}) \\
&= \text{prox}_{\alpha h}\left(\mathbf{x}^{(i)} - \alpha\nabla g(\mathbf{x}^{(i)})\right).
\end{aligned} \tag{12}$$

This technique is a cheap alternative compared to other optimization algorithms, especially when the proximal operator can be computed in closed form and can be efficiently evaluated. With respect to `BARISTA`, each term in (8) is separable, allowing us to express the problem in the form (9) with

$$g(\mathbf{W}) = -\sum_{i=1}^{n} \log\left(\widehat{P}(c_i|\mathbf{x}_i)\right) + \rho_2||\mathbf{W}||_2^2, \qquad h(\mathbf{W}) = \rho_1||\mathbf{W}||_1. \tag{13}$$

Using Theorem 1, we obtain

$$\nabla g(\mathbf{W}) = -\sum_{i=1}^{n}(v_{c_i,i} - v_i) + 2\rho_2\mathbf{W}, \tag{14}$$

where

$$v_{c,i} = \sum_{k=1}^{m} \log\theta_{c,k,\mathbf{x}_i}\mathbf{e}_{c,k}, \qquad v_i = \sum_{c=1}^{l}\widehat{P}(c|\mathbf{x}_i)v_{c,i} \qquad (c = 1, \ldots, l). \tag{15}$$

The proximal operator of $\rho_1||\mathbf{W}||_1$ with $\rho_1 > 0$ is the entrywise soft thresholding operator $\eta_{\alpha\rho_1}(w_{i,j})$, given by

$$\left(\eta_{\alpha\rho_1}(\mathbf{W})\right)_{c,j} = \text{sgn}(w_{c,j})\big(|w_{c,j}| - \alpha\rho_1\big)_+, \tag{16}$$

where $(x)_+ := \max(x, 0)$. The resulting proximal gradient update for the objective (8) is therefore

$$\mathbf{W}^{(i+1)} = \eta_{\alpha\rho_1}\left(\mathbf{W}^{(i)} - \alpha\nabla g(\mathbf{W}^{(i)})\right), \tag{17}$$

with $\nabla g(\mathbf{W})$ as in (14). Since each update involves a soft-thresholding operation, the method is known as the iterative shrinkage-thresholding algorithm (ISTA) in the literature. See, e.g., [2] and [1, Chapter 10] for more details. We now discuss our implementation of `BARISTA`, which uses an accelerated version of ISTA to further improve convergence.

3.2 Bayesian Regularized Iterative Shrinkage-Thresholding Algorithm (`BARISTA`)

`BARISTA` implements both the ISTA approach given in (17) and the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) of Beck and Teboulle [2] to minimize (8). FISTA is an accelerated variation of the iterative shrinkage-thresholding algorithm. The method maintains the computational simplicity of ISTA, while providing a better global rate of convergence. Instead of using only the previous iterate $\mathbf{W}^{(k-1)}$ to find an updated set of parameters through the soft thresholding operator, a combination of the two previous iterates, $\mathbf{W}^{(k-2)}, \mathbf{W}^{(k-1)}$ is used. With either ISTA or FISTA, we use a backtracking line search to find an appropriate step size that decreases a quadratic approximation of (8):

Algorithm 1 FISTA with Backtracking Line Search

Step 0. Take $\alpha > 0$ and $\mathbf{W}^{(0)} \in \mathbb{R}^{l\times m}$. Set $\mathbf{Y}^{(1)} = \mathbf{W}^{(0)}$, $t_1 = 1$.

Step k. ($k \geq 1$) Find the smallest non-negative integer γ such that, for $\alpha^{(k)} = \alpha\left(\frac{1}{2}\right)^{\gamma}$,

$$F\left(\eta_{\alpha^{(k)}\rho_1}\left(\mathbf{Y}^{(k)}\right)\right) \leq Q_{\alpha^{(k)}}\left(\eta_{\alpha^{(k)}\rho_1}\left(\mathbf{Y}^{(k)}\right), \mathbf{Y}^{(k)}\right)$$

for Q as given in (18). Using $\alpha^{(k)}$, compute,

$$\mathbf{W}^{(k)} = \eta_{\alpha^{(k)}\rho_1}\left(\mathbf{Y}^{(k)}\right)$$

$$t^{(k+1)} = \frac{1 + \sqrt{1 + 4\left(t^{(k)}\right)^2}}{2},$$

$$\mathbf{Y}^{(k+1)} = \mathbf{W}^{(k)} + \left(\frac{t^{(k)} - 1}{t^{(k+1)}}\right)\left(\mathbf{W}^{(k)} - \mathbf{W}^{(k-1)}\right).$$

$$Q_\alpha(x, y) = g(y) + \langle x - y, \nabla g(y)\rangle + \frac{1}{2\alpha}\|x - y\|^2 + h(x), \tag{18}$$

where g, h are as in (13). FISTA, as applied to our framework, is given in Algorithm 1. The `BARISTA` algorithm is given in Algorithm 2. The initial choice of the weights is a matrix of 1's, which employs NB as the starting point. `BARISTA` computes the posterior distributions using the training data and finds $\mathbf{W}^*$ that minimizes (8) as in Algorithm 2. This solution is then used to compute the weighted posterior distributions of unseen data to make classifications.

Algorithm 2 BARISTA

```
input: Training Data $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, initial step size $\alpha$, tolerance $\epsilon$, $\ell_1$ penalty $\rho_1$, $\ell_2$ penalty $\rho_2$. Set initial iterate $\mathbf{W}_0 = \mathbf{1}$ with dimensions $l \times m$ and $\Delta = 2\epsilon$.
(1) Estimate the vector of priors $\pi \in \mathbb{R}^l$ from $D$ as in (4).
(2) Estimate the likelihood matrices $\{\mathbf{\Theta}\}_{i=1}^n$ for each $\mathbf{x}_i \in D$ as in (4).
while $\Delta > \epsilon$ do
    (3) Derive the posterior probabilities $\widehat{P}(C|D)$ from (3).
    (4) Compute the model loss of using the current iterate as in (8).
    (5) Update $\mathbf{W}$ using ISTA or FISTA as in (17) or Algorithm 1.
    (6) Set $\Delta = ||\mathbf{W}^{(k)} - \mathbf{W}^{(k-1)}||_1$.
end while
output: $\mathbf{W}^*$, where $\mathbf{W}^*$ is the solution found in by minimizing (8).
```

4 Convergence Analysis

We now examine the theoretical properties of our approach to minimize (8). We first show that the differentiable part of (8) has a Lipschitz continuous gradient.

Lemma 1. *Let $f(\mathbf{W})$ be as in Equation (7) and $v_{c,i}$ as in Equation (15). Then we have $\|\nabla^2 f(\mathbf{W})\|_2 \leq L$, where*

$$L := \sum_{i=1}^{n} \max_{c=1,\dots,l} \|v_{c,i}\|_2^2. \tag{19}$$

As a consequence, the function $g(\mathbf{W})$ in Equation (13) is $(L + 2\rho_2)$-smooth, i.e., its gradient is Lipschitz with constant $L + 2\rho_2$.

Proof. Please see supplementary material.

We next prove that the differentiable part of (8) is strongly convex.

Lemma 2. *Let $\lambda := \lambda_{min}(\nabla^2 f(\mathbf{W})) = \lambda_{min}(-\sum_{i=1}^n H_i)$ denote the smallest eigenvalue of the positive semidefinite matrix $\nabla^2 f(\mathbf{W})$, where $H_i = v_i v_i^T - \sum_{c=1}^{l} \widehat{P}(c|\mathbf{x}_i) v_{c,i} v_{c,i}^T$. Then the function $g(\mathbf{W})$ in Equation (13) is σ-strongly convex with $0 < \lambda + 2\rho_2 \leq \sigma \leq L + 2\rho_2$, where L is given in (19).*

Proof. Please see supplementary material.

Our next result guarantees the convergence of the iterates (17) to minimize (8).

Theorem 2 (Convergence of ISTA in the BARISTA framework). *Let $\mathbf{W}^{(k)}$ denote the ISTA iterates (17) for minimizing Equation (8), with either a fixed step size $\alpha = 1/(L + 2\rho_2)$ or a step size chosen via a backtracking line search analogous to the procedure described in Algorithm 1. Assume $\mathbf{W}^*$ is the optimal solution with optimal value F^*. Then*

1. *For all $k \geq 1$, we have $F(\mathbf{W}^{(k)}) - F^* \leq \frac{c \cdot \|\mathbf{W}^{(0)} - \mathbf{W}^*\|_2^2}{2k}$.*

2. For all $k \geq 0$, we have $\|\mathbf{W}^{(k+1)} - \mathbf{W}^*\|_2^2 \leq \left(1 - \frac{\sigma}{c}\right) \cdot \|\mathbf{W}^{(k)} - \mathbf{W}^*\|_2^2$,

where $c = L + 2\rho_2$ in the constant step size case, $c = \max(2(L + 2\rho_2), \frac{1}{\alpha})$ if the backtracking line search is used, L is given in Equation (19), and σ is the strong convexity constant of $\nabla^2 g(\mathbf{W})$ given in Lemma 2.

Proof. See [1, Theorem 10.21] and [1, Theorem 10.29].

Remark 1. Notice that in Theorem 2(2), we always have $0 \leq 1 - \sigma/c < 1$ since $0 < \lambda + 2\rho_2 \leq \sigma \leq L + 2\rho_2$ (see Lemma 2). Moreover, the Hessian $\nabla^2 f(\mathbf{W})$ is a sum of n positive semidefinite matrices $-H_i$, where n denotes the sample size of the data. Since $\mathbf{W}$ is a $l \times m$ matrix, where l is the number of attribute classes and m is the number of features, we expect the Hessian to be positive definite when $n > l \times m$. In that case, $\lambda > 0$ and the ℓ_2 penalty in (8) is not necessary to guarantee linear convergence of the iterates.

Finally, a result analogous to Theorem 2(1) can be shown for FISTA, but with faster convergence.

Theorem 3 (Convergence of FISTA in the `BARISTA` framework). *Let $\mathbf{W}^{(k)}$ denote the FISTA iterates for minimizing Equation (8), with either a fixed step size $\alpha = 1/(L + 2\rho_2)$ or a step size chosen via the backtracking line search procedure described in Algorithm 1. Assume $\mathbf{W}^*$ is the optimal solution with optimal value F^*. Then for all $k \geq 0$*

$$F(\mathbf{W}^{(k)}) - F^* \leq \frac{2c \cdot \|\mathbf{W}^{(0)} - \mathbf{W}^*\|_2^2}{(k+1)^2},$$

where $c = L + 2\rho_2$ in the constant step size case, $c = \max(2(L + 2\rho_2), \frac{1}{\alpha})$ if the backtracking line search is used, and where L is given in Equation (19).

Proof. See [1, Theorem 10.34].

5 Performance Results

We now provide performance results to demonstrate the performance of `BARISTA` on real datasets. In section 5.1, we measure the classification performance of `BARISTA` and give semilog plots to visualize convergence. In section 5.2, we give details regarding experimental settings.

5.1 Benchmark Datasets

Table 1 provides the accuracy score of `BARISTA` on eight benchmark datasets (described in section 5.2) in comparison to several Naive Bayes variations. Our goal here is to illustrate how the added ℓ_1 and ℓ_2 regularization of `BARISTA` can yield a significant accuracy improvement to weighted Naive Bayes methods. In selecting datasets for our experiments, we aimed to create a heterogeneous test bed for the new method. The datasets were chosen based on attributes such as the number of features and sample size, ensuring diversity in terms of size, complexity, and variability. This approach allows for a comprehensive evaluation of the method's performance across a wide range of scenarios, and avoids the bias of selecting datasets that favorably highlight our method's performance.

Dataset	BARISTA	RNB [9]	CWANB [12]	CAWNB [6]	NB [10]	WANBIA [13]	FTAWNB [14]
breast-w	**97.86**	96.99	97.07	96.50	97.25	96.51	97.14
colic	**84.00**	83.42	82.83	83.07	81.20	83.72	81.75
heart-statlog	**88.15**	82.96	85.04	84.33	83.74	84.74	83.78
iris	**99.33**	97.33	94.60	94.67	94.33	94.33	95.53
kr-vs-kp	**95.40**	93.08	94.38	95.20	87.81	93.92	94.70
mushroom	99.90	**99.96**	99.84	**99.96**	98.03	99.90	99.85
segmentation	94.37	**95.84**	95.27	94.68	92.91	95.24	91.52
zoo	**100.00**	98.09	96.15	95.95	95.75	95.75	96.35
AVERAGE	94.88	93.46	93.15	93.04	91.38	93.01	92.58

Table 1. Experimental accuracy of `BARISTA` and other competing methods.

On average, `BARISTA` performs nearly 1.5% better than the next best method and brings an improvement of more than 5% in accuracy compared to some of the other competing methods, demonstrating the benefit of using both ℓ_1 and ℓ_2 regularization. Due to the attribute dependence in the selected datasets we see that `BARISTA` strongly outperforms vanilla Naive Bayes, demonstrating `BARISTA`'s ability to mitigate performance loss in the presence of attribute dependencies in the data. Furthermore, comparing `BARISTA` to the CAWNB frameworks acts as an ablation study to evaluate performance gains when using ℓ_1 and ℓ_2 regularization.

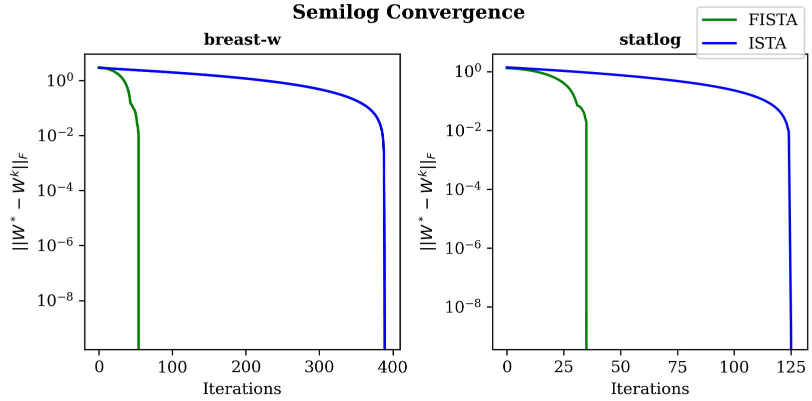

Fig. 1. Convergence rates of FISTA vs. ISTA for solving the `BARISTA` problem.

Figure 1 illustrates the linear convergence rates of ISTA and the dramatically faster convergence of FISTA. Both techniques achieve super-linear convergence near a solution.

5.2 Experimental Settings

Each dataset listed in Table 1 can be found on the UCI machine learning repository [3]. With respect to pre-processing, missing values were replaced with the mean value or maximum frequency of the attribute. Numerical attributes were discretized using the MDL method [4]. To get an average accuracy score, a 5-fold cross validation procedure was employed. For each experiment, the initial step size was set to 0.1, and the tolerance, as defined in (Algorithm 2) by ϵ, was set to 10^{-6}. Finally, the hyper-parameters $\rho_1 \in [0.01, 0.03, 0.06, 0.09, 0.12]$ and $\rho_2 \in [0.00001, 0.0001, 0.001, 0.005, 0.01, 0.05]$, are reported based on the best performance during the 5-fold cross validation procedure. We report the best accuracy for our method based on these experiments in table 1. As expected, smaller datasets obtain larger penalties to mitigate over-fitting (see Table 2).

Dataset	breast-w	colic	heart-statlog	iris	kr-vs-kp	mushroom	segmentation	zoo
Instances	699	368	270	150	3169	8124	2310	101
Inst./Class	350	184	135	50	1598	4062	330	14
ℓ_1 Penalty	0.03	0.03	0.01	0.12	0.01	0.06	0.03	0.12
ℓ_2 Penalty	0.001	0.001	0.01	0.05	0.00001	0.005	0.01	0.05

Table 2. Characteristics of the datasets and optimal penalty parameters selected.

Remark 2. Table 1 focuses on the accuracy metric, as is typically done in the weighted Naive Bayes literature (see, e.g., [6,9]). The above datasets do not present any major class imbalance, making accuracy a reasonable metric to compare performance. We note, however, that it would be useful for the community to report additional metrics in the future.

6 Conclusion

In this paper, an accelerated proximal gradient method was applied to a regularized class-specific attribute weighted Naive Bayes framework. A brief introduction to iterative shrinkage-thresholding algorithms, including ISTA and FISTA was given. The convexity of the negative log-likelihood loss function was proved and rigorous theoretical convergence guarantees for `BARISTA` were presented. The supplemental material contains the proofs of Theorem 1 and Lemmas 1 & 2, as well as plots analogous to Figure 1 for all datasets considered. Performance results have also been shown, comparing the accuracy of `BARISTA` to other weighted Naive Bayes methods. The results indicate that `BARISTA` is very competitive, especially on small sample data where regularization can be a valuable tool. One should keep in mind, however, that no method will outperform all the others on every dataset. Methods based on Naive Bayes such as BARISTA are expected to perform well when the features are approximately conditionally independent. In the presence of strong dependence, one should consider techniques outside of Naive Bayes. Moving forward, it would be interesting to explore second order techniques that handle ℓ_1 regularization to further speed up convergence, as well as other approaches to alleviate the conditional independence assumption that hinders Naive Bayes classifiers in practice.

References

1. A. Beck. *First-order methods in optimization.* SIAM, 2017.
2. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
3. D. Dua and C. Graff. UCI machine learning repository, 2017.
4. U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *International Joint Conference on Artificial Intelligence*, 1993.
5. L. Jiang, G. Kong, and C. Li. Wrapper framework for test-cost-sensitive feature selection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(3):1747–1756, 2021.
6. L. Jiang, L. Zhang, L. Yu, and D. Wang. Class-specific attribute weighted naive bayes. *Pattern Recognition*, 88:321–330, 2019.
7. B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
8. S. Taheri, J. Yearwood, M. Mammadov, and S. Seifollahi. Attribute weighted naive bayes classifier using a local optimization. *Neural Computing and Applications*, 24, 04 2014.
9. S. Wang, J. Ren, and R. Bai. A regularized attribute weighting framework for naive bayes. *IEEE Access*, PP:1–1, 12 2020.
10. A. R. Webb. *Statistical pattern recognition.* John Wiley & Sons, 2003.
11. J. Wu and Z. Cai. Attribute weighting via differential evolution algorithm for attribute weighted naive bayes (wnb). *Journal of Computer Information Systems*, 7, 05 2011.
12. L. Yu, S. Gan, Y. Chen, and M. He. Correlation-based weight adjusted naive bayes. *IEEE Access*, 8:51377–51387, 2020.
13. N. Zaidi, J. Cerquides, M. Carman, and G. Webb. Alleviating naive bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, 14:1947–1988, 06 2013.
14. H. Zhang and L. Jiang. Fine tuning attribute weighted naive bayes. *Neurocomputing*, 488:402–411, 2022.
15. Z. Zheng, Y. Cai, Y. Yang, and Y. Li. 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, 2018.
16. H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.