Naive Bayes - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs

1.1 Introduction

Naive Bayes (NB) is a generative family of classifiers derived from Bayes' theorem under the *naive* assumption that features are conditionally independent given the class. Despite this strong assumption, NB performs well in high-dimensional settings and admits simple, closed-form parameter estimates.

1.2 Data and Notation

Let

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \qquad \mathbf{x}_i \in \mathbb{R}^d, \;\; y_i \in \{1, 2, \ldots, K\}.$$

We write $\pi_k \triangleq \Pr(Y = k)$ for class priors, $p(\mathbf{x} \mid Y = k)$ for class-conditional densities, and x_{ij} for the j-th component of $\mathbf{x}_i$.

1.3 Naive Bayes Classification Model

The goal is to compute the posterior probability of a class c given a new observation $\mathbf{x} \in \mathbb{R}^d$. By Bayes' theorem:

$$P(Y = c \mid \mathbf{x}) = \frac{P(Y = c)\, P(\mathbf{x} \mid Y = c)}{P(\mathbf{x})}.$$

Since $P(\mathbf{x})$ is the same for all classes, for classification we can use the proportionality:

$$P(Y = c \mid \mathbf{x}) \propto P(Y = c)\, P(\mathbf{x} \mid Y = c).$$

Naive (Conditional Independence) Assumption

The *naive* assumption is that the features are conditionally independent given the class label Y. That is, for $\mathbf{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^d$,

$$P(\mathbf{x} \mid Y = c) = \prod_{j=1}^{d} P(x_j \mid Y = c).$$

Here, x_j is the jth feature of $\mathbf{x}$ (a scalar), and $P(x_j \mid Y = c)$ is the likelihood of feature j given class c. Thus, the posterior becomes:

$$P(Y = c \mid \mathbf{x}) \propto P(Y = c) \prod_{j=1}^{d} P(x_j \mid Y = c).$$

For classification, the decision rule is to choose the class $\hat{c}$ that maximizes this posterior:

$$\hat{c} = \arg \max_{c \in \{1,\ldots,K\}} P(Y = c) \prod_{j=1}^{d} P(x_j \mid Y = c).$$

Logarithmic Transformation

Because the product of many probabilities can be very small and to turn the product into a sum (which is often easier to work with), we take the logarithm:

$$\log P(Y = c \mid \mathbf{x}) \propto \log P(Y = c) + \sum_{j=1}^{d} \log P(x_j \mid Y = c).$$

Thus, the decision rule can be equivalently written as:

$$\hat{c} = \arg \max_{c \in \{1,\ldots,K\}} \left\{ \log P(Y = c) + \sum_{j=1}^{d} \log P(x_j \mid Y = c) \right\}.$$

1.4 Compact Formulation

The Bayes posterior is

$$\Pr(Y = k \mid \mathbf{x}) = \frac{\pi_k \, p(\mathbf{x} \mid Y = k)}{\sum_{t=1}^{K} \pi_t \, p(\mathbf{x} \mid Y = t)}. \tag{1}$$

NB assumes conditional independence across features given Y:

$$p(\mathbf{x} \mid Y = k) = \prod_{j=1}^{d} p_j(x_j \mid Y = k). \tag{2}$$

For classification, constants in the denominator cancel, yielding the *log-score*

$$\begin{aligned} \text{score}_k(\mathbf{x}) \;&=\; \log \pi_k + \sum_{j=1}^{d} \log p_j(x_j \mid Y = k), \\ \hat{y}(\mathbf{x}) &= \arg \max_{k \in \{1,\ldots,K\}} \text{score}_k(\mathbf{x}). \end{aligned} \tag{3}$$

Computing in log-space avoids numerical underflow from long products.

1.5 Parameter Estimation via (Regularized) Maximum Likelihood

Given i.i.d. data and Eqn. (2), the complete-data log-likelihood is

$$\mathcal{L}(\{\pi_k\}, \{p_j(\cdot \mid k)\}) = \sum_{i=1}^{n} \log \pi_{y_i} + \sum_{i=1}^{n} \sum_{j=1}^{d} \log p_j(x_{ij} \mid y_i). \tag{4}$$

The MLE prior is $\widehat{\pi}_k = \frac{n_k}{n}$ with $n_k = \sum_i \mathbf{1}\{y_i = k\}$. Estimates for $p_j(\cdot \mid k)$ depend on the chosen likelihood model per feature.

1.6 Common Likelihood Models

Bernoulli Naive Bayes (binary features). Assume $x_j \in \{0, 1\}$ and $X_j \mid Y = k \sim \text{Bernoulli}(\theta_{jk})$:

$$p_j(x_j \mid k) = \theta_{jk}^{x_j}(1 - \theta_{jk})^{1-x_j}.$$

Let $m_{jk} = \sum_{i:y_i=k} x_{ij}$ and $n_k = |\{i : y_i = k\}|$. Then

$$\widehat{\theta}_{jk}^{\text{MLE}} = \frac{m_{jk}}{n_k}, \qquad \widehat{\theta}_{jk}^{\text{MAP}} = \frac{m_{jk} + a - 1}{n_k + a + b - 2} \quad \text{for } \theta_{jk} \sim \text{Beta}(a, b).$$

The class score is

$$\text{score}_k(\mathbf{x}) = \log \widehat{\pi}_k + \sum_{j=1}^{d} \Big[x_j \log \widehat{\theta}_{jk} + (1 - x_j) \log(1 - \widehat{\theta}_{jk}) \Big]. \tag{5}$$

Multinomial Naive Bayes (bag-of-words counts). Let $\mathbf{x} = (x_1, \ldots, x_V)$ be counts over a vocabulary of size V with length $N = \sum_{v=1}^{V} x_v$. Conditional on $Y = k$, words are i.i.d. from $\boldsymbol{\phi}_{\cdot|k}$:

$$p(\mathbf{x} \mid k) = \frac{N!}{\prod_{v=1}^{V} x_v!} \prod_{v=1}^{V} \phi_{v|k}^{x_v}, \qquad \sum_{v=1}^{V} \phi_{v|k} = 1.$$

Let n_{vk} be the total count of word v across all training documents with $y = k$ and $N_k = \sum_v n_{vk}$. Then

$$\widehat{\phi}_{v|k}^{\text{MLE}} = \frac{n_{vk}}{N_k}, \qquad \widehat{\phi}_{v|k}^{\text{MAP}} = \frac{n_{vk} + \alpha}{N_k + \alpha V} \quad \text{for } \boldsymbol{\phi}_{\cdot|k} \sim \text{Dirichlet}(\alpha, \ldots, \alpha).$$

Since the multinomial coefficient does not depend on k,

$$\text{score}_k(\mathbf{x}) = \log \widehat{\pi}_k + \sum_{v=1}^{V} x_v \log \widehat{\phi}_{v|k}. \tag{6}$$

Gaussian Naive Bayes (continuous features). Assume $X_j \mid Y = k \sim \mathcal{N}(\mu_{jk}, \sigma_{jk}^2)$ and independence across j:

$$p_j(x_j \mid k) = \frac{1}{\sqrt{2\pi\sigma_{jk}^2}} \exp\Big(- \frac{(x_j - \mu_{jk})^2}{2\sigma_{jk}^2} \Big).$$

With $\mathcal{I}_k = \{i : y_i = k\}$ and $n_k = |\mathcal{I}_k|$,

$$\widehat{\mu}_{jk} = \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} x_{ij}, \qquad \widehat{\sigma}_{jk}^2 = \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} (x_{ij} - \widehat{\mu}_{jk})^2.$$

The class score becomes

$$\text{score}_k(\mathbf{x}) = \log \widehat{\pi}_k - \frac{1}{2}\sum_{j=1}^{d}\Big[\log\big(2\pi\widehat{\sigma}_{jk}^2\big) + \frac{(x_j - \widehat{\mu}_{jk})^2}{\widehat{\sigma}_{jk}^2}\Big]. \tag{7}$$

Decision boundaries. Between k and ℓ,

$$\begin{aligned}\Delta_{k\ell}(\mathbf{x}) &= \text{score}_k(\mathbf{x}) - \text{score}_\ell(\mathbf{x}) \\ &= \log\frac{\widehat{\pi}_k}{\widehat{\pi}_\ell} - \frac{1}{2}\sum_{j=1}^{d}\left[\log\frac{\widehat{\sigma}_{jk}^2}{\widehat{\sigma}_{j\ell}^2} + \frac{(x_j - \widehat{\mu}_{jk})^2}{\widehat{\sigma}_{jk}^2} - \frac{(x_j - \widehat{\mu}_{j\ell})^2}{\widehat{\sigma}_{j\ell}^2}\right].\end{aligned}$$

If $\widehat{\sigma}_{jk}^2 = \widehat{\sigma}_{j\ell}^2$ for all j, $\Delta_{k\ell}$ is linear in $\mathbf{x}$; otherwise it is axis-aligned quadratic.

1.7 Algorithm (Naive Bayes Classifier)

1. **Estimate priors:** $\widehat{\pi}_k = n_k/n$ (or user-specified).
2. **Estimate class-conditionals:**
   - Bernoulli: $\widehat{\theta}_{jk}$ (optionally MAP with Beta smoothing).
   - Multinomial: $\widehat{\phi}_{v|k}$ (optionally MAP with add-α).
   - Gaussian: $\widehat{\mu}_{jk}, \widehat{\sigma}_{jk}^2$ (optionally with conjugate priors).
3. **Predict:** For a query $\mathbf{x}$, compute $\text{score}_k(\mathbf{x})$ from Eqn. (5), Eqn. (6), or Eqn. (7) and output $\hat{y}(\mathbf{x}) = \arg\max_k \text{score}_k(\mathbf{x})$.

1.8 Variables and Dimensions

- $\mathbf{x}_i \in \mathbb{R}^d$: feature vector; $x_{ij} \in \mathbb{R}$ its j-th component.
- $y_i \in \{1, \ldots, K\}$: class label; n samples; d features; K classes.
- $\pi_k \in [0, 1]$: class prior; $\sum_k \pi_k = 1$.
- Bernoulli: $\theta_{jk} \in (0, 1)$; counts m_{jk}, n_k.
- Multinomial: vocabulary size V; parameters $\phi_{v|k} \in (0, 1)$ with $\sum_v \phi_{v|k} = 1$; counts n_{vk}, totals N_k.
- Gaussian: per-feature means $\mu_{jk} \in \mathbb{R}$ and variances $\sigma_{jk}^2 > 0$; optionally collect $\boldsymbol{\mu}_k = (\mu_{1k}, \ldots, \mu_{dk})^\top \in \mathbb{R}^d$ and diagonal covariance $\text{diag}(\sigma_{1k}^2, \ldots, \sigma_{dk}^2)$.

1.9 Summary

From first principles: apply Bayes rule and impose conditional independence Eqn. (2), yielding the additive log-score Eqn. (3). Fit priors and per-feature class-conditionals by (regularized) maximum likelihood with simple closed forms for Bernoulli, Multinomial, and Gaussian models. Prediction amounts to maximizing a sum of log-priors and log-likelihoods; under Gaussian NB, equal per-feature variances produce linear decision boundaries, while unequal variances yield axis-aligned quadratics.