Naive Bayes Classifier

Dr. Ashish Tendulkar

IIT Madras

Machine Learning Techniques

Introduction

- Generative counterpart of logistic regression.
- Uses Bayes theorem for calculating probability of a sample belonging to a class.
- Makes strong (naive) conditional independence assumption between the features given a label.
- Simple yet very powerful classifier that is used extensively in applications like document classification and spam filtering.

Part 1: Training Setup

Binary classification

Index of example

$$D = \{(\mathbf{X}, \mathbf{y})\} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n}$$

Feature matrix (n, m)

Label vector $(n,)$

$y^{(i)} \in \{0, 1\}$

Multiclass classification

Index of example

$$D = \{(\mathbf{X}, \mathbf{Y})\} = \left\{\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)\right\}_{i=1}^{n}$$

Feature matrix (n, m)

Label matrix (n, k)

$\mathbf{y}^{(i)} \in \{0, 1\}^k$

Spot the difference!

Part 2: Model

Naive Bayes' assumption

Naive Bayes classifier makes a strong conditional independence assumption:

Features are conditionally independent given the label.

It enables us to express joint probability of features given label as product of probabilities of individual features given label:

$$p(x_1, x_2, \ldots, x_m|y) = p(x_1|y)\, p(x_2|y) \ldots p(x_m|y) = \prod_{j=1}^{m} p(x_j|y)$$

NB classifier predicts probability, $p(y|\mathbf{x})$, of class label, y, given a feature vector $\mathbf{x}$, using Bayes' theorem

Class conditional density

Class prior

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x},y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

Posterior probability

Evidence

With Naive Bayes assumption, posterior probability $p(y|\mathbf{x})$ becomes

$$p(y = y_c|\mathbf{x}) = \frac{p(\mathbf{x}|y_c)\, p(y_c)}{p(\mathbf{x})}$$

Rewriting $p(\mathbf{x}|y) = p(x_1, x_2, \ldots, x_m|y)$ and $p(\mathbf{x}) = p(x_1, x_2, \ldots, x_m)$

$$= \frac{p(x_1, x_2, \ldots, x_m|y_c)\, p(y_c)}{p(x_1, x_2, \ldots, x_m)}$$

Expressing denominator as a sum over all k labels.

$$= \frac{p(x_1, x_2, \ldots, x_m|y_c)\, p(y_c)}{\sum_{r=1}^{k} p(x_1, x_2, \ldots, x_m, y_r)}$$

$$= \frac{p(x_1, x_2, \ldots, x_m | y_c)\ p(y_c)}{\sum_{r=1}^{k} p(x_1, x_2, \ldots, x_m, y_r)}$$

Rewriting the denominator after applying the chain rule

$$\sum_{r=1}^{k} p(x_1, x_2, \ldots, x_m, y_r) = \sum_{r=1}^{k} p(x_1, x_2, \ldots, x_m | y_r) p(y_r)$$

$$= \frac{p(x_1, x_2, \ldots, x_m | y_c)\ p(y_c)}{\sum_{r=1}^{k} p(x_1, x_2, \ldots, x_m | y_r) p(y_r)}$$

Rewriting numerator and denominator following conditional independence assumption $p(x_1, x_2, \ldots, x_m | y) = p(x_1 | y)\ p(x_2 | y) \ldots p(x_m | y)$

$$p(y = y_c | \mathbf{x}) = \frac{p(y_c) p(x_1 | y_c) p(x_2 | y_c) \ldots p(x_m | y_c)}{\sum_{r=1}^{k} p(y_r) p(x_1 | y_r) p(x_2 | y_r) \ldots p(x_m | y_r)}$$

$$p(y = y_c|\mathbf{x}) = \frac{p(y_c)p(x_1|y_c)p(x_2|y_c)\dots p(x_m|y_c)}{\sum_{r=1}^{k} p(y_r)p(x_1|y_r)p(x_2|y_r)\dots p(x_m|y_r)}$$

Rewriting numerator and denominator compactly

$$= \frac{p(y_c)\prod_{j=1}^{m} p(x_j|y_c)}{\sum_{r=1}^{k} p(y_r)\prod_{j=1}^{m} p(x_j|y_r)}$$

Parameters of naive Bayes classifier

$$p(y = y_c|\mathbf{x}) = \frac{p(y_c)\prod_{j=1}^{m} p(x_j|y_c)}{\sum_{r=1}^{k} p(y_r)\prod_{j=1}^{m} p(x_j|y_r)}$$

k prior probabilities

$\{p(y_1), p(y_2), \ldots, p(y_k)\}$

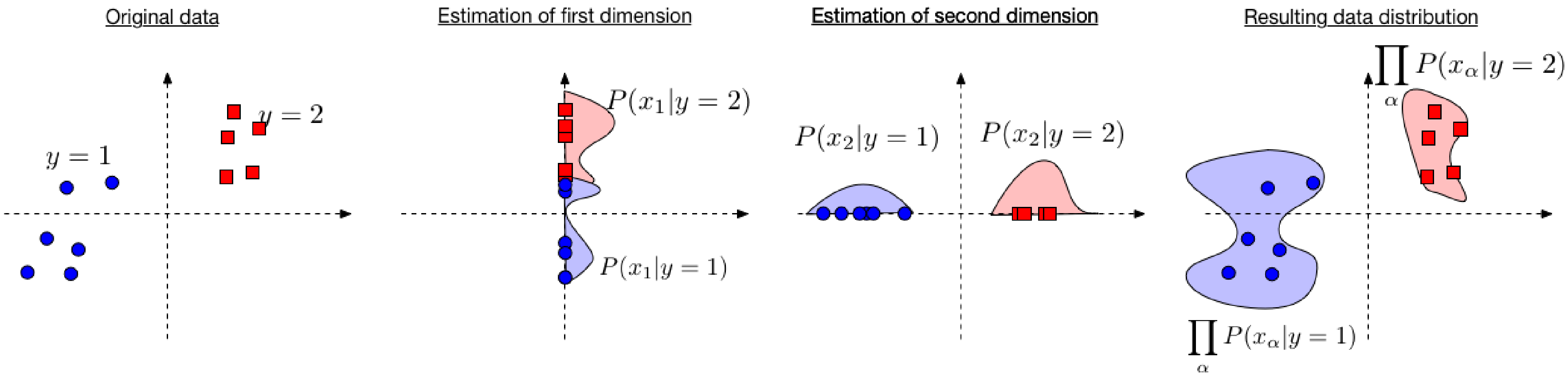
$k \times m$ class conditional densities

$\{p(x_1|y_r), p(x_2|y_r), \ldots, p(x_m|y_r)\}$

m conditional densities per class and there are k such classes.

The number of parameters for each conditional density vary and depends on its mathematical form.

NB schematic



Credits: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html

Modeling conditional densities: $p(x_i|y)$

- Depends on the nature of the feature x_i:
  - Binary feature - e.g. *word is present or not*
    - *categorical feature is generalization of binary.*
  - Multinomial feature - e.g. *word count* c_i as features with additional constraint that $\sum_{i=1}^{m} c_i = l$, the length of the sequence they represent.
  - Continuous feature - Features are real numbers. e.g. *area of an apartment in sq. feet.*

Modeling $p(x_j|y_c)$

Probability distribution used for modeling $p(x_j|y_c)$ depends on the nature of the feature x_j:

- Categorical feature: $p(x_j|y_c) \sim \mathrm{Cat}(e, \mu_{j1c}, \mu_{j2c}, \ldots, \mu_{jec})$
- Binary feature: $p(x_j|y_c) \sim \mathrm{Bernoulli}(\mu_{jc})$

- Continuous feature: $p(x_j|y_c) \sim \mathcal{N}(\mu_{jc}, \sigma_{jc})$
- Multinomial feature: $p(\mathbf{x}|y_c) \sim \mathrm{Multinomial}(l, \mu_{1c}, \mu_{2c}, \ldots, \mu_{mc})$

Note: we need to estimate parameters of relevant distributions, one for each feature, for each class label.

Let $\mathbf{w}$ be the set of all parameters: priors as well as class conditional densities

Bernoulli Distribution

When x_j is a binary feature, we use Bernoulli distribution to model the class conditional density: $p(x_j|y_c)$

Parameterized by $\mu_j c$, $p(x_j|y_c)$ is calculated as follows:

- $p(x_j = 1|y_c) = \mu_{jc}$
- $p(x_j = 0|y_c) = 1 - \mu_{jc}$

Combine these two equations into a compact form as follows:

$$p(x_j|y_c;\mu_{jc}) = \mu_{jc}^{x_j}(1-\mu_{jc})^{(1-x_j)}$$

Verify that the compact form and earlier form are equivalent.

When $x_j = 1$, $\mu_{jc}^{1}(1-\mu_{jc})^{(1-1)} = \mu_{jc}^{1}(1-\mu_{jc})^{0} = \mu_{jc}$

and $x_j = 0$, $\mu_{jc}^{0}(1-\mu_{jc})^{(1-0)} = \mu_{jc}^{0}(1-\mu_{jc})^{1-0} = 1 - \mu_{jc}$

For $s \leq m$ binary features and k classes, we will have $k \times s$ parameters for s Bernoulli distributions.

Categorical Distribution

When x_j is a categorical feature i.e. it takes one of the $e > 2$ discrete values [e.g. $\{\text{red, green, blue}\}$ or roll of a dice], we use categorical distribution to model the class conditional density $p(x_j|y_c)$.

Let $v = \{v_1, v_2, \ldots, v_e\}$ be the set of e discrete values.

For discrete set v, $p(x_j|y_c)$ is parameterized by the $|v|$, that is # events in v and probability of each event $\mu_{j1c}, \mu_{j2c}, \ldots, \mu_{jec}$ such that $\sum_{q=1}^{e} \mu_{jqc} = 1$

For $x_j = v_q$ such that $v_q \in v$:

$$p(x_j = v_q | y_c; e, \mu_{j1c}, \mu_{j2c}, \ldots, \mu_{jec}) = \mu_{jqc}$$

Let $\boldsymbol{\mu_{jc}} = [\mu_{j1c}, \mu_{j2c}, \ldots, \mu_{jec}]$ be the parameter vector for $p(x_j|y_c)$

$p(x_j|y_c)$ can be written in a compact form as follows:

$$p(x_j|y_c; e, \boldsymbol{\mu_{jc}}) = \mu_{j1c}^{\mathbb{1}(x_j=v_1)} \mu_{j2c}^{\mathbb{1}(x_j=v_2)} \ldots \mu_{jec}^{\mathbb{1}(x_j=v_e)}$$

where $\mathbb{1}(x_j = v_q) = 1 \text{ if } x_j = v_q \text{ else } 0$

Verify that the compact form is equivalent to the following:

$$p(x_j = v_1|y_c; e, \boldsymbol{\mu_{jc}}) = \mu_{j1c}$$

$$p(x_j = v_2|y_c; e, \boldsymbol{\mu_{jc}}) = \mu_{j2c}$$

$$\vdots$$

$$p(x_j = v_e|y_c; e, \boldsymbol{\mu_{jc}}) = \mu_{jec}$$

Total parameters = $k \times \sum_{j=1}^{m} |v_j|$

Multinomial Distribution

When $\mathbf{x}$ is count vector i.e. each component x_j is a count of appearance in the object it represents and $\sum x_j = l$, which is the length of the object, we use multinomial distribution to model $p(\mathbf{x}|y_c)$.

Used for modelling documents that are represented by the word counts.

It is parameterized by the length of object l and probability of features $\{x_1, \ldots, x_m\}$: $\mu_{1c}, \ldots, \mu_{mc}$.

The probability of $p(\mathbf{x}|y_c)$ such that $\sum_{j=1}^{m} x_j = l$ is given by:

$$p(\mathbf{x}|y_c; l, \mu_{1c}, \mu_{2c}, \ldots, \mu_{mc}) = \frac{n!}{x_1! \ldots x_m!} \prod_{j=1}^{m} \mu_{jc}^{x_j}$$

Total parameters = $k \times m$

Gaussian Distribution

When x_j is a continuous feature i.e. it takes a real value, we use gaussian (or normal) distribution to model the class conditional density $p(x_j|y_c)$.

It is parameterised by the mean μ_{jc} and variance σ_{jc}^2.

value of j-th feature

mean of x_j for class y_c

$$p(x_j|y_c;\mu_{jc},\sigma_{jc}^2) = \frac{1}{\sqrt{2\pi}\sigma_{jc}} e^{-\frac{1}{2}\left(\frac{x_j-\mu_{jc}}{\sigma_{jc}}\right)^2}$$

standard deviation of x_j for class y_c

This is 1-D gaussian distribution. It models class conditional density for a single feature.

Multivariate Gaussian Distribution

Alternately, we can use multivariate gaussian distribution to represent $p(\mathbf{x}|y)$ with parameters mean vector $\mu_{m\times 1}$ and covariance matrix $\Sigma_{m\times m}$.

In NB setting, since the features are conditionally independent of one another, all entries of Σ except diagonal are zero.

- The diagonal entries represent variance of that feature i.e. $\Sigma_{jj} = \sigma_j^2$

$$p(\mathbf{x}|y;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^m|\Sigma|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

Total parameters = $k \times 2m$

Once we learn parameters of different conditional densities, we use them to infer class label for new example.

Inference

We assign a class label y_c to a new example $\mathbf{x}$ that maximizes the posterior probability.

Let $\mathbf{w}$ be the set of all paramaters.

$$y = \text{argmax}_{y_c} p(y_c | \mathbf{x}; \mathbf{w})$$

Using the definition of posterior probability

$$= \text{argmax}_{y_c} \frac{p(\mathbf{x} | y_c; \mathbf{w}) \; p(y_c; \mathbf{w})}{p(\mathbf{x}; \mathbf{w})}$$

Since $p(\mathbf{x}; \mathbf{w})$ is independent of y_c, we ignore denominator from this computation

$$= \text{argmax}_{y_c} p(\mathbf{x} | y_c; \mathbf{w}) p(y_c; \mathbf{w})$$

$$= \text{argmax}_{y_c} p(\mathbf{x}|y_c;\mathbf{w})p(y_c;\mathbf{w})$$

Expanding $p(\mathbf{x}|y_c;\mathbf{w})$ with naive Bayes assumption, we get

$$= \text{argmax}_{y_c} \left(\prod_{j=1}^{m} p(x_j|y_c;\mathbf{w}) \right) p(y_c;\mathbf{w})$$

This equation involves multiplication of small numbers, there is a risk of underflow in this calculation:

$$y = \text{argmax}_{y_c} \left(\sum_{j=1}^{m} \log p(x_j|y_c;\mathbf{w}) \right) + \log p(y_c;\mathbf{w})$$

For a new example, $\mathbf{x}$, we assign a class label y_c that yields max value among all $y = \{y_1, \ldots, y_k\}$.

The following equation is useful for getting the class label. It does not return the probability of an example belonging to class y_c.

$$y = \operatorname{argmax}_{y_c} \left(\sum_{j=1}^{m} \log p(x_j | y_c; \mathbf{w}) \right) + \log p(y_c; \mathbf{w})$$

In case, we want the probability, we should use the following equation

$$p(y_c | \mathbf{x}; \mathbf{w}) = \frac{p(\mathbf{x} | y_c; \mathbf{w}) \, p(y_c; \mathbf{w})}{p(\mathbf{x}; \mathbf{w})}$$

This calculation should also be performed in $\log$ space.

Part 3: Loss function

Likelihood describes joint probability of observed data D given the parameter $\mathbf{w}$ for the chosen statistical model.

$$L(\mathbf{w}) = p(D; \mathbf{w}) = p(\mathbf{X}, \mathbf{y}; \mathbf{w})$$

Since training examples are i.i.d., we can express this as a product of probability of individual samples:

$$L(\mathbf{w}) = \prod_{i=1}^{n} p(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w})$$

For mathematical and computational convenience, we calculate log likelihood by taking log on both the sides:

$$\log L(\mathbf{w}) = \log \left(\prod_{i=1}^{n} p(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w}) \right)$$

The product becomes sum in the log space.

$$l(\mathbf{w}) = \sum_{i=1}^{n} \log \left(p(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w}) \right)$$

Log likelihood is defined as

$$l(\mathbf{w}) = \sum_{i=1}^{n} \log \left(p(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w}) \right)$$

Our job is to find the parameter vector $\mathbf{w}$ such that the $l(\mathbf{w})$ is maximized.

Equivalently we can minimize the negative log likelihood (NLL) to maintain uniformity with other algorithms:

$$\begin{aligned} J(\mathbf{w}) &= -l(\mathbf{w}) \\ &= -\sum_{i=1}^{n} \log \left(p(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w}) \right) \end{aligned}$$

$$l(\mathbf{w}) = \sum_{i=1}^{n} \log\left(p(\mathbf{x}^{(i)}, y^{(i)}; \mathbf{w})\right)$$

Simplifying with naive Bayes assumptions of conditional independence of features given label:

$$= \sum_{i=1}^{n} \log\left(\left(\prod_{i=1}^{m} p(\mathbf{x}_j^{(i)} | y^{(i)}; \mathbf{w})\right) \; p(y^{(i)}; \mathbf{w})\right)$$

Applying log on product makes it summation in log.

$$= \sum_{i=1}^{n} \left(\sum_{i=1}^{m} \log p(x_j^{(i)} | y^{(i)}; \mathbf{w})\right) + \log p(y^{(i)}; \mathbf{w})$$

Rearranging

$$= \sum_{i=1}^{n} \log p(y^{(i)}; \mathbf{w}) + \sum_{i=1}^{n} \sum_{j=1}^{m} \log p(x_j^{(i)} | y^{(i)}; \mathbf{w})$$

$$l(\mathbf{w}) = \sum_{i=1}^{n} \log p(y^{(i)}; \mathbf{w}) + \sum_{i=1}^{n} \sum_{i=1}^{m} \log p(x_j^{(i)} | y^{(i)}; \mathbf{w})$$

The calculation of $p(x_j^{(i)} | y^{(i)})$ depends on the probability distribution of the features.

Part 4: Optimization for parameter estimation

The parameter estimation by maximizing the log likelihood function is carried out with the following three steps:

1. Calculate partial derivation of log likelihood function w.r.t. each parameter.
2. Set the partial derivative to 0, which is the condition at maxima.
3. Solve the resulting equation to obtain the parameter value.

Since $p(x_j|y)$ depends on the choice of probability distribution, we will discuss parameter estimation for different distributions separately.

Estimating prior probability: $p(y)$

The total number of parameters to be estimated is equal to the number of class labels k - one prior per label.

$$p(y = y_c) = \frac{\sum_{i=1}^{n} 1(y^{(i)} = y_c)}{n}$$

Note that $1(y^{(i)} = y_c) = 1$ when $y^{(i)} = y_c$ else 0.

The prior probability for class y_c is equal to the ratio of the number of examples with label y_c to the total number of examples in the training set n.

Estimating class conditional densities

Bernoulli distribution

$$l(\mathbf{w}) = \sum_{i=1}^{n} \log p(y^{(i)}; \mathbf{w}) + \sum_{i=1}^{n} \sum_{j=1}^{m} \log p(x_j^{(i)} | y^{(i)}; \mathbf{w})$$

Recall $p(x_j | y_c; w_{jc}) = w_{jc}^{x_j}(1 - w_{jc})^{(1-x_j)}$, substituting this in $l(\mathbf{w})$

$$= \sum_{i=1}^{n} \log w_{y^{(i)}} + \sum_{i=1}^{n} \sum_{j=1}^{m} \log \left(w_{jy^{(i)}}^{x_j^{(i)}} (1 - w_{jy^{(i)}})^{1-x_j^{(i)}} \right)$$

Distributing log into the bracket - multiplication turns into addition

$$= \sum_{i=1}^{n} \log w_{y^{(i)}} + \sum_{i=1}^{n} \sum_{j=1}^{m} x_j^{(i)} \log w_{jy^{(i)}} + (1 - x_j^{(i)}) \log (1 - w_{jy^{(i)}})$$

Parameters for label y_r: $\mathbf{w} = w_{1r}, w_{2r}, \ldots, w_{mr}$

(Step 1) calculate $\frac{\partial l(\mathbf{w})}{\partial\, w_{jy_r}}$ and set it to 0.

$$\frac{\partial l(\mathbf{w})}{\partial w_{jy_r}} = \frac{\partial}{\partial w_{jy_r}} \left(\sum_{i=1}^{n} \log w_{y^{(i)}} + \sum_{i=1}^{n} \sum_{j=1}^{m} x_j^{(i)} \log w_{jy^{(i)}} + (1 - x_j^{(i)}) \log (1 - w_{jy^{(i)}}) \right)$$

Applying derivative to individual terms in the loss equation.

$$= \sum_{i=1}^{n} \frac{\partial}{\partial w_{jy_r}} \log w_{y^{(i)}} + \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\partial}{\partial w_{jy_r}} \left(x_j^{(i)} \log w_{jy^{(i)}} + (1 - x_j^{(i)}) \log (1 - w_{jy^{(i)}}) \right)$$

The derivatives of the first term and all terms where $y^{(i)} \neq y_r$ are 0. Retaining terms where $y^{(i)} = y_r$.

$$= \sum_{i=1}^{n} 1(y^{(i)} = y_r) \frac{\partial}{\partial w_{jy_r}} \left(x_j^{(i)} \log w_{jy_r} + (1 - x_j^{(i)}) \log (1 - w_{jy_r}) \right)$$

$$= \sum_{i=1}^{n} 1(y^{(i)} = y_r) \left(\frac{x_j^{(i)}}{w_{jy_r}} - \frac{1 - x_j^{(i)}}{1 - w_{jy_r}} \right)$$

(Step 2) Setting $\frac{\partial l(\mathbf{w})}{\partial\, w_{jy_r}}$ to 0.

$$\frac{\partial l(\mathbf{w})}{\partial w_{jy_r}} = \sum_{i=1}^{n} 1(y^{(i)} = y_r) \left(\frac{x_j^{(i)}}{w_{jy_r}} - \frac{(1 - x_j^{(i)})}{(1 - w_{jy_r})} \right) = 0$$

(Step 3) Solving it further with algebraic manipulation:

$$\sum_{i=1}^{n} 1(y^{(i)} = y_r) \left(\frac{x_j^{(i)}}{w_{jy_r}} - \frac{(1 - x_j^{(i)})}{(1 - w_{jy_r})} \right) = 0$$

$$\sum_{i=1}^{n} 1(y^{(i)} = y_r) \left(x_j^{(i)}(1 - w_{jy_r}) - (1 - x_j^{(i)}) w_{jy_r} \right) = 0$$

$$\sum_{i=1}^{n} 1(y^{(i)} = y_r) \left(x_j^{(i)} - w_{jy_r} \right) = 0$$

$$\sum_{i=1}^{n} 1(y^{(i)} = y_r) x_j^{(i)} = \sum_{i=1}^{n} 1(y^{(i)} = y_r) w_{jy_r}$$

This yields:

$$w_{jy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)} = y_r) x_j^{(i)}}{\sum_{i=1}^{n} 1(y^{(i)} = y_r)}$$

examples with label y_r and $x_j = 1$

$$w_{jy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)}=y_r)x_j^{(i)}}{\sum_{i=1}^{n} 1(y^{(i)}=y_r)}$$

examples with label y_r

What if $\sum_{i=1}^{n} 1(y^{(i)} = y_r)x_j^{(i)} = 0$?

Leads to w_{jy_r} = 0, which would mean $p(x_j|y_r)$ = 0

Leads to $p(y = y_r|\mathbf{x}) = 0$ since $p(x_j|y_r)$ = 0

Fixing problem with zero count

Laplace smoothing: We can correct it by adding +1 to numerator and +2 to denominator (1 for each value of feature: $x_j \in \{0, 1\}$).

$$w_{jy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)} = y_r) x_j^{(i)} + 1}{\sum_{i=1}^{n} 1(y^{(i)} = y_r) + 2}$$

In general, we can add $+c$ to numerator and $+2c$ to denominator. c is a hyperparameter that helps control overfitting.

$$w_{jy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)} = y_r) x_j^{(i)} + c}{\sum_{i=1}^{n} 1(y^{(i)} = y_r) + 2c}$$

However too high value of c leads to underfitting.

Categorical distribution

$$w_{jvy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)}=y_r)\, 1(x_j^{(i)}=v)}{\sum_{i=1}^{n} 1(y^{(i)}=y_r)}$$

In plain english, this is ratio of number of examples with label y_r and $x_j = v$ to the total number of training examples with label y_r.

Parameters:

$$\mathbf{w} = \{w_{111}, \ldots, w_{1e1}, \ldots, w_{m11}, \ldots, w_{me1}, \ldots, w_{mek}\}$$

Incorporating smoothing, we obtain

$$w_{jvy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)}=y_r)\, 1(x_j^{(i)}=v)+c}{\sum_{i=1}^{n} 1(y^{(i)}=y_r)+ce}$$

Smoothing factor c is a hyperparameter and $c = 1$ leads to Laplace smoothing.

Multinomial distribution

$$w_{jy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)}=y_r)\, x_j^{(i)}}{\sum_{i=1}^{n} 1(y^{(i)}=y_r) \sum_{j=1}^{m} x_j^{(i)}}$$

In plain english, this is ratio of number of training examples where x_j appears with label y_r to the sum of feature values in training examples with label y_r.

Incorporating smoothing, we obtain

$$w_{jy_r} = \frac{\sum_{i=1}^{n} 1(y^{(i)}=y_r)\, x_j^{(i)} + c}{\sum_{i=1}^{n} 1(y^{(i)}=y_r) \sum_{j=1}^{m} x_j^{(i)} + cm}$$

Smoothing factor c is a hyperparameter and $c = 1$ leads to Laplace smoothing.

Gaussian/Normal distribution

Let n_r be the number of examples of class y_r

$$n_r = \sum_{i=1}^{n} 1(y^{(i)} = y_r)$$

There are two parameters per feature $\{\mu_j, \sigma_j^2\}$ per label.

$$\mu_{jr} = \frac{1}{n_r} \sum_{i=1}^{n} 1(y^{(i)} = y_r) x_j^{(i)}$$

$$\sigma_{jr}^2 = \frac{1}{n_r} \sum_{i=1}^{n} 1(y^{(i)} = y_r)(x_j^{(i)} - \mu_{jr})^2$$

Part 5: Evaluation

Evaluation

Classification evaluation measures with cross validation and test set:

- Confusion matrix
- Precision/recall/F1 score
- AUC ROC/PR curve