Notes 03-6: Regression

Linear regression techniques attempt to model data using a straight line.

Given a set of data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where y_i is some response corresponding to x_i, linear regression is a method for determining the function that best fits the observed data points.

The first step in fitting a straight line to the data points is to construct a scatter plot.

If the points appear to approximate a straight line, linear regression may be an appropriate analysis technique.

DIAGRAM = Classification.C.1.b1

If they don't, some other technique is required.

DIAGRAM = Classification.C.1.b2

The method of least squares assumes the best-fit curve is one that has the minimal sum of the deviations squared from a given set of data points.

The general regression equation can be written as

$$\hat{y} = \alpha + \beta x$$

where α and β are called the regression coefficients.

The regression coefficients can be estimated from the following two equations:

$$\beta = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\alpha = \bar{y} - \beta\bar{x}$$

Where $\bar{x}$ is the mean of the x values in the sample, $\bar{y}$ is the mean of the y values, β represents the slope of the line through the points, and α represents the y-intercept.

Example – Linear regression

Consider the table shown below, where *Salary* is shown for various values of *Years of Service*. The objective is to use the data in this table to predict *Salary* based upon *Years of Service*. *Salary* is called the *explanatory* variable and *Years of Service* is called the *response* variable.

Salary	Years of Service
30	3
57	8
64	9
72	13
36	3
43	6
59	11
90	21
20	1
83	16

A scatter plot corresponding to the values in the table is shown below.

DIAGRAM = Classification.C.1.d

Based upon the values in the table, $\bar{x} = 9.1$, $\bar{y} = 55.4$, $\beta = 3.54$, and $\alpha = 23.19$. Therefore $\hat{y} = 23.19 + 3.54x$.

Salary can now predicted for any value of *Years of Service*. However, keep in mind that it is just a prediction. For example, the actual versus predicted *Salary* for *Years of Service* from the original table is shown below.

Salary	Years of Service	$\hat{y} = \alpha + \beta x$
30	3	33.81
57	8	51.51
64	9	55.05
72	13	69.21
36	3	33.81
43	6	44.43
59	11	62.13
90	21	97.53
20	1	26.73

83	16	79.83

When interpreting the regression coefficients:

The estimated slope β = 3.54 implies that each additional year of service results in an increase in salary of $3,450.

The regression line should not be used to predict the response $\hat{y}$ when x lies outside the range of the initial values.

Example

DIAGRAM = Classification.C.1.e

Coefficient of Determination

The *coefficient of determination* represents the proportion of the total variability that is explained by the model.

The coefficient of determination is represented by

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where the numerator is the measure of the total variability of the fitted values and the denominator is the measure of the total variability of the original values.

A value close to 1 implies that most of the variability is explained by the model.

A value close to 0 implies that the model is not appropriate.

Naïve Bayes

The *Naïve Bayes* classifier is a well-known and highly effective classifier based upon *Bayes' Rule*, a technique used to estimate the likelihood of class membership of an unseen instance given the set of labeled instances.

The *prior* (or *unconditional*) probability, $P(a)$, associated with a proposition a (i.e., an assertion that a is true) is the degree of belief accorded to it in the absence of any other information.

Example – Prior probability

$P(rain = \text{true}) = 0.25$ or $P(rain) = 0.25$

The *posterior* (or *conditional*) probability, $P(a \mid b)$, associated with a proposition a is the degree of belief accorded to it given that all we know is b.

Example – Posterior probability

$P(rain \mid thunder) = 0.8$

A prior probability, such as $P(rain)$, can be thought of as a special case of the posterior probability $P(rain \mid \;)$, where the probability is conditioned on no evidence.

Posterior probabilities can be defined in terms of prior probabilities. Specifically,

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

for $P(b) > 0$, which can also be written as

$$P(a \wedge b) = P(a|b)P(b)$$

In a nutshell: For a and b to be true, we need b to be true, and we need a to be true given b.

Since conjunction is commutative $P(a \wedge b) = P(b \wedge a)$, so

$$P(a \wedge b) = P(a|b)P(b)$$

can be written equivalently as

$$P(b \wedge a) = P(b|a)P(a)$$

Then, since $P(a \wedge b) = P(b \wedge a)$, we have Bayes' Rule

$$P(b|a)P(a) = P(a|b)P(b)$$

which can be written as

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$

A Naïve Bayes classifier applies to classification tasks where each instance *x* is described by a conjunction of attribute values (i.e., a tuple $\langle a_1, a_2, \ldots, a_n \rangle$) and where the target class can take on any value from some finite set *C* (i.e., the set of possible class values).

A set of labeled instances is provided from which the prior and posterior probabilities can be derived.

Predicting with Naïve Bayes

When a new instance is presented, the classifier is asked to predict the class label.

The Bayesian approach considers a set of candidate hypotheses (i.e., the various possible class labels) and determines the hypothesis (i.e., the class label) that is most probable given the labeled instances (known as the *maximum posteriori hypothesis* (*MAP*)).

Given a new instance with attribute values $\langle a_1, a_2, \ldots, a_n \rangle$, the most probable class label is given by

$$C_{MAP} = \arg\max_{C_j \in C} P(C_j | a_1, a_2, \ldots, a_n)$$

Using Bayes' Rule, the above expression can be written as

$$C_{MAP} = \arg\max_{C_j \in C} \frac{P\left(a_1, a_2, \ldots, a_n \middle| C_j\right) P(C_j)}{P(a_1, a_2, \ldots, a_n)}$$

or

$$C_{MAP} = \arg\max_{C_j \in C} P(a_1, a_2, \ldots, a_n | C_j) P(C_j)$$

That is, the denominator $P(a_1, a_2, \ldots, a_n)$ can be dropped because it is a constant term independent of C_j.

Since a Naïve Bayes classifier assumes the effect of an attribute value on a given class is independent of the values of the other attributes (called the *class conditional independence assumption*), given C_{MAP}, the probability of observing the conjunction $\langle a_1, a_2, \ldots, a_n \rangle$ is just the product of the probabilities of the individual attributes. That is,

$$P(a_1, a_2, \ldots, a_n | C_j) = \prod_{i=1}^{n} P(a_i | C_j)$$

Substituting $\prod_{i=1}^{n} P(a_i|C_j)$ for $P(a_1, a_2, \ldots, a_n|C_j)$ in the equation for C_{MAP} yields

$$C_{NB} = \arg\max_{C_j \in C} P(C_j) \prod_{i=1}^{n} P(a_i | C_j)$$

where C_{NB} denotes the assigned class label output by the Naïve Bayes classifier.

Naïve Bayes Classifier for Categorical Attributes

```
Algorithm: Naïve Bayes Learner
Input: D = a set of labeled instances of the form ⟨a1, a2, ..., an⟩,
       where each ai corresponds to a value from the domain of
       attributes A1, A2, ..., An, respectively, and an is the
       assigned class label
Output: classProbability = an array of prior probabilities
        attributeProbability = an array of posterior
probabilities
        v = an array of the number of unique values in the
domain of each attribute

Method:
totalCount = 0
m = the number of unique classes in the domain of An
n = the number of attributes in the instances of D
for j = 1 to m
    classCount [j] = 0
    for i = 1 to n - 1
        v [i] = the number of unique values in the domain of Ai
        for k = 1 to v [i]
        attributeCount [j, i, k] = 0
for each instance of D
    totalCount ++
    j = an integer corresponding to the class of the current instance
    classCount [j] ++
    for i = 1 to n - 1
        k = an integer corresponding to the value of the current attribute
```

```
            attributeCount [j, i, k] ++
    for j = 1 to m
        classProbability [j] = classCount [j] / totalCount
        for i = 1 to n
            for k = 1 to v [i]
                attributeProbability [j, i, k] = attributeCount [j, i, k] / classCount [j]
```

```
Algorithm: NaiveBayesClassifier
Input: classProbability = an array of prior probabilities
       attributeProbability = an array of posterior probabilities
       m = the number of unique classes in the domain of A_n
       n = the number of attributes in the instances of D
       v = an array of the number of unique values in the domain of each attribute
       ⟨a_1, a_2, …, a_{n-1}⟩ = an unlabeled instance
Output: C_NB = the class label

Method:
    C_NB = 0
    for j = 1 to m
        C_Temp = classProbability [j]
        for i = 1 to n - 1
            for k = 1 to v [i]
                if a_i == the attribute value corresponding to v [i]
                    C_Temp = C_Temp * attributeProbability [j, i, k]
                    break
        if C_Temp > C_NB
            C_NB = C_Temp
```

Example – Predicting a class label using a Naïve Bayes classifier

Tuple	Age	Income	Student	Credit Rating	Buys Computer
t_1	<=30	*high*	*no*	*fair*	*no*
t_2	<=30	*high*	*no*	*excellent*	*no*
t_3	31..40	*high*	*no*	*fair*	*yes*
t_4	>40	*medium*	*no*	*fair*	*yes*
t_5	>40	*low*	*yes*	*fair*	*yes*
t_6	>40	*low*	*yes*	*excellent*	*no*
t_7	31..40	*low*	*yes*	*excellent*	*yes*
t_8	<=30	*m*	*no*	*fair*	*no*

t_9	<=30	*low*	*yes*	*fair*	*yes*
t_{10}	>40	*medium*	*yes*	*fair*	*yes*
t_{11}	<=30	*medium*	*yes*	*excellent*	*yes*
t_{12}	31..40	*medium*	*no*	*excellent*	*yes*
t_{13}	31..40	*high*	*yes*	*fair*	*yes*
t_{14}	>40	*medium*	*no*	*excellent*	*no*

The class label attribute is Buys Computer and it has two unique values: *yes* and *no*. The unlabeled instance to be classified is

⟨Age = "<=30", Income = *medium*, Student = *yes*, Credit Rating = *fair*⟩.

Let a_1 = "<=30", a_2 = *medium*, a_3 = *yes*, and a_4 = *fair*. So, the problem is to determine $P(C_j|a_1, a_2, a_3, a_4)$ for all j. Now,

$P(C_1) = P(\text{Buys Computer} = yes) = 9/14$

and

$P(C_2) = P(\text{Buys Computer} = no) = 5/14$.

To determine C_{NB}, we only need to concern ourselves with the conditional probabilities associated with the attribute values on the unlabeled instance. So,

$$\begin{aligned} P(C_1)\prod_{i=1}^{n} P(a_i|C_1) &= P(C_1)\,P(a_1|C_1)\,P(a_2|C_1)\,P(a_3|C_1)\,P(a_4|C_1) \\ &= (9/14)(2/9)(4/9)(6/9)(6/9) \\ &= (0.643)(0.222)(0.444)(0.667)(0.667) \\ &= 0.028 \end{aligned}$$

and

$$\begin{aligned} P(C_2)\prod_{i=1}^{n} P(a_i|C_2) &= P(C_2)\,P(a_1|C_2)\,P(a_2|C_2)\,P(a_3|C_2)\,P(a_4|C_2) \\ &= (5/14)(3/5)(2/5)(1/5)(2/5) \\ &= (0.357)(0.6)(0.4)(0.2)(0.4) \\ &= 0.007 \end{aligned}$$

We need to maximize $P(C_j)\prod_{i=1}^{n} P(a_i|C_j)$. Therefore, $C_{NB} = C_1$ = (Buys Computer = *yes*).