Minimum Description Length Principle and Naïve Bayes Classifier

Dr. Kuppusamy .P

Associate Professor / SCOPE

Minimum Description Length (MDL) **Principle**

- Representing the concept in shortest explanation for the observed data.
- Apply Bayesian theorem to solve the arguments of inductive bias using MDL principle.
- The Minimum Description Length principle is motivated by h_{MAP}, $h_{MAP} = \underset{h \in H}{argmax}$ P(D|h) $*$ P(h)

- Applying the **logarithm**, $h_{MAP} = \underset{h \in H}{argmax}\ log_2$P(D|h) $+ log_2$P(h)
- Transforming into minimize i.e., to get minimizing the length $h_{MAP} = \underset{h \in H}{argmin} - log_2$P(D|h) $- \ log_2$P(h)

- This equation interpreted as that short hypotheses are preferred for encoding the hypotheses and data.

E.g.,

- Consider the problem of designing a code to transmit messages drawn at random, where the probability of encountering message i is p_i.
- Assume, the code that minimizes the expected number of bits must transmit to encode a message.
- To minimize the expected code length, we should assign shorter codes to messages that are more probable.
- The optimal code (i.e., the code that minimizes the expected message length) assigns **log** $\boldsymbol{p_i}$ bits to encode message i.

Minimum Description Length Principle

- Consider the number of bits required to encode message *i* using code C is denoted as $L_c(i)$ i.e., description length of message *i* with respect to *C*.
- The Equation

$$h_{MAP} = \underset{h \in H}{argmin} - log_2 \mathrm{P}(\mathrm{D}|\mathrm{h}) - log_2 \mathrm{P}(\mathrm{h})$$

h_{MAP} provide results based on shanon coding theory as:

- $-log_2\mathrm{P}(\mathrm{h})$ is the description length of h under the optimal encoding for the hypothesis space H, i.e., $L_{C_H}(h) = -log_2\mathrm{P}(\mathrm{h})$, C_H is the optimal code for hypothesis space H.
- $-log_2\mathrm{P}(\mathrm{D}|\mathrm{h})$ is the description length of the training data D given hypothesis h, under its optimal encoding i.e., $L_{C_{D|h}}(h) = -log_2\mathrm{P}(\mathrm{D}|\mathrm{h})$, $C_{D|h}$ is the optimal code for describing data D assuming that both the sender and receiver know the hypothesis h.
- So, rewrite Equation

$$h_{MAP} = \underset{h}{argmin} L_{C_H}(h) + L_{C_{D|h}}(D|h).$$

- It minimizes the sum given by the description length of the hypothesis plus the description length of the data given the hypothesis.
- C_H and $C_{D|h}$ are the optimal encodings for H and for D given h respectively.

Minimum Description Length Principle

- The Minimum Description Length (MDL) principle recommends choosing the hypothesis that minimizes the sum of these two description lengths.
- Assuming we choose the codes C_1 and C_2 to represent the hypothesis and the data given the hypothesis.

$$h_{MDL} = \underset{h \in H}{argmin} L_{C_1}(h) + L_{C_2}(D|h)$$

$$h_{MAP} = \underset{h}{argmin} L_{C_H}(h) + L_{C_{D|h}}(D|h).$$

- When $C_1 = C_H$ and $C_2 = C_{D|h}$, both equations $h_{MDL} = h_{MAP}$.

Intuition:

- MDL principle recommends the shortest method for re-encoding the training data, where we count both the size of the hypothesis and any additional cost of encoding the $(D|h)$.

Example.

- Apply the MDL principle to decision trees problem with training data.

What to be chosen for the representations C_1 and C_2 of hypotheses and data?

- For C_1, choose some obvious encoding of decision trees, in which the description length grows with the number of nodes in the tree and with the number of edges.

Minimum Description Length Principle

How shall we choose the encoding C_2 of the (D|h)?

- Assume sequence of instances X = (x_l . . . x_m) is already known to both the transmitter and receiver. So that we need only transmit the classifications (f (x_l) . . . f (x_m)).
- When the training classifications (f (x_l) . . . f (x_m)) are identical to the predictions of the hypothesis, then there is no need to transmit any information about these examples (the receiver can compute these values once it has received the hypothesis).
- Therefore, The description length of the classifications given the hypothesis is zero.
- If few examples are misclassified by h, then for each misclassification we need to transmit a message that identifies which example is misclassified (using at most log_2m bits) as well as its correct classification (using at most log_2k bits, k - number of possible classifications).
- The hypothesis h_{MDL} under the encodings C_1 and C_2 is minimizing the sum of these description lengths.
- Thus the MDL principle provides a way of trading off hypothesis complexity for the number of errors committed by the hypothesis.
- It might select a shorter hypothesis that makes a few errors over a longer hypothesis that perfectly classifies the training data.
- Based on this perspective, Apply the MDL principle to choose the best size for a decision tree to deal the overfitting.

BAYES OPTIMAL CLASSIFIER

What is the most probable (hypothesis) classification of the new instance given the training data?

- Apply the MAP hypothesis to the new instance for classification.
- Consider a H containing three hypotheses, h_1, h_2 and h_3. The posterior probabilities of these hypotheses given the training data are 0.4, 0.3, and 0.3 respectively.
- Thus, h_1 is the MAP hypothesis.
- Suppose a new instance x is encountered, which is classified positive by h_1, but negative by h_2 and h_3.
- Taking all hypotheses into account, the probability that x is positive is 0.4 (the probability associated with h_1), and the probability that it is negative is 0.6. The most probable classification (negative) in this case is different from the classification generated by the MAP hypothesis.
- In general, the most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities.
- If the possible classification of the new example can take on any value v_j from some set V, then the probability P(v_j|D) that the correct classification for the new instance is v_j,

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i) * \mathrm{P}(h_i|\mathrm{D})$$

- The optimal classification of the new instance is the value v_j, for which P (v_j|D) is maximum.

BAYES OPTIMAL CLASSIFIER

$$Bayes\ optimal\ classification = \underset{v_j \in V}{argmax} \sum_{h_i \in H} P(v_j|h_i) * \mathrm{P}(h_i|\mathrm{D})$$

Example, the set of possible classifications of the new instance is V = (+, -), and

- $\mathrm{P}(h_1|\mathrm{D})$=0.4 ; $\mathrm{P}(-|h_1)$=0 $\mathrm{P}(+|h_1)$=1
- $\mathrm{P}(h_2|\mathrm{D})$=0.3 ; $\mathrm{P}(-|h_2)$=1 $\mathrm{P}(+|h_2)$=0
- $\mathrm{P}(h_3|\mathrm{D})$=0.3 ; $\mathrm{P}(-|h_3)$=1 $\mathrm{P}(+|h_3)$=0

- Therefore, $\sum_{h_i \in H} \mathrm{P}(+|h_i) * \mathrm{P}(h_i|\mathrm{D}) = 0.4$; $\sum_{h_i \in H} \mathrm{P}(-|h_i) * \mathrm{P}(h_i|\mathrm{D}) = 0.6$

$$Bayes\ optimal\ classification = \underset{v_j \in (+,-)}{argmax} \sum_{h_i \in H} P(v_j|h_i) * \mathrm{P}(h_i|\mathrm{D}) = -$$ i.e., Negative

- Any model that classifies new instances according to above Equation is called a Bayes optimal classifier.
- This method maximizes the probability that the new instance is classified correctly, given the available data, hypothesis space, and prior probabilities over the hypotheses.

GIBBS ALGORITHM

- Bayes optimal classifier provides best optimal. But it can be quite costly to apply due to it computes the posterior probability for every hypothesis in H and then combines the predictions of each hypothesis to classify each new instance.
- Gibbs algorithm provides less optimal as follows
    1. Choose a hypothesis h from H at random, according to the posterior probability distribution over H.
    2. Use h to predict the classification of the next instance x.
- Given a new instance to classify, the Gibbs algorithm simply applies a hypothesis drawn at random according to the current posterior probability distribution.
- Surprisingly, under certain conditions the expected misclassification error for the Gibbs algorithm is at most twice the expected error of the Bayes optimal classifier.
- More precisely, the expected value is taken over target concepts drawn at random according to the prior probability distribution assumed by the learner.
- Under this condition, the expected value of the error of the Gibbs algorithm is at worst twice the expected value of the error of the Bayes optimal classifier.

Naive Bayes Classifier (Bayesian learning method)

- Naïve denotes the occurrence of a certain feature is independent of the occurrence of other features.
- Primarily used in text classification that includes a high-dimensional training dataset.

Naive Bayes assumption:

- independent - each feature is independent on other feature
- equal – Each feature contributes equally to the outcome

- The Naive Bayes (probabilistic) classifier applies to each instance x is described by a conjunction of attribute values and where the target function *f(x)* can take on any value from some finite set of target values Y.
- A set of training examples of the target function and a new instance is described by the tuple of attribute values $(a_1, a_2 \ldots a_n)$ and *f(x)*. The learner predicts the classification for this new instance.
- The Bayesian approach classifies the new instance by assigning the most probable target value y, given the attribute values $X = (a_1, a_2 \ldots a_n)$ that describe the instance.

$$P(\mathrm{y}|\mathrm{X}) = \frac{\mathrm{P}(\mathrm{X}|\mathrm{y}) * \mathrm{P}(\mathrm{y})}{\mathrm{P}(\mathrm{X})}$$

Rewrite using chain rule by substituting for X

$$P(\mathrm{y}|a_1, a_2 \ldots a_n) = \frac{\mathrm{P}(a_1|\mathrm{y})\ \mathrm{P}(a_2|\mathrm{y})\mathrm{P}(a_3|\mathrm{y}) \ldots .\mathrm{P}(a_n|\mathrm{y}) * \mathrm{P}(\mathrm{y})}{\mathrm{P}(a_1)\mathrm{P}(a_2)\ \ldots .\ \mathrm{P}(a_n)}$$

Naive Bayes Classifier (Bayesian learning method)

- $P(\mathrm{y}|a_1, a_2 \ldots a_n) = \dfrac{\mathrm{P}(a_1|\mathrm{y})\ \mathrm{P}(a_2|\mathrm{y})\mathrm{P}(a_3|\mathrm{y}) \ldots\mathrm{P}(a_n|\mathrm{y}) * \mathrm{P}(\mathrm{y})}{\mathrm{P}(a_1)\mathrm{P}(a_2)\ \ldots\ \mathrm{P}(a_n)}$

- i.e., $P(\text{Target Function}|\text{Features}) = \dfrac{P(Features|Target\ Function) * P(Target\ function)}{P(Feature1) * P(Feature2) \ldots\ldots P(Featuren)}$

- The denominator remains constant for a given input. So, ignore this term.

$P(\mathrm{y}|a_1, a_2 \ldots a_n) = \mathrm{P}(a_1|\mathrm{y})\ \mathrm{P}(a_2|\mathrm{y})\mathrm{P}(a_3|\mathrm{y}) \ldots .\ \mathrm{P}(a_n|\mathrm{y}) * \mathrm{P}(\mathrm{y})$

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

- Compute the probability of given set of inputs for all possible values of the class *y*, and choose the y with maximal probability value.

$$y = \underset{y_i \in Y}{argmax}\ \mathrm{P}(\mathrm{y}) * \prod_{i=1}^{n} \mathrm{P}(a_i|\mathrm{y})\ ;$$

- P(y) – class probability and $\mathrm{P}(a_i|\mathrm{y})$ – conditional probability

Example: Naive Bayes Classifier for Discrete Data

- Consider a set of 14 training examples of the target concept PlayTennis.
- The attributes Outlook, Temp, Humidity, and Wind.
- Apply naive Bayes classifier and to classify the new test instance:

(Outlook = sunny, Temperature = hot, Humidity = Normal, Wind=Weak)

- Predict the target value (yes or no) of the target concept PlayTennis for this new instance.

$$y = \underset{y_i \in Y}{argmax} \, \mathrm{P(y)} * \prod_{i=1}^{n} \mathrm{P}(a_i|\mathrm{y})$$

Outlook	**Temp**	**Humidity**	**Windy**	**Play**
Rainy	Hot	High	Weak	No
Rainy	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Sunny	Mild	High	Weak	Yes
Sunny	Cool	Normal	Weak	Yes
Sunny	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Rainy	Mild	High	Weak	No
Rainy	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Weak	Yes
Rainy	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Sunny	Mild	High	Strong	No

Naive Bayes Classifier for Discrete Data

New test instance : Predict the play?

(Outlook = sunny, Temp = hot, Humidity = Normal, Wind=Weak)

$$P(yes|today) = \frac{P(yes) * P(outlook = sunny|yes)\ P(Temp = hot\ |yes)\ P(Humidity = Normal|yes)\ P(wind=weak|yes)}{P(today)}$$

$$P(yes|today) = \frac{9}{14} * \frac{3}{9} * \frac{2}{9} * \frac{6}{9} * \frac{6}{9} = 0.02105$$

$$P(No|today) = \frac{P(No) * P(outlook = sunny|No)\ P(Temp = hot\ |No)\ P(Humidity = Normal|No)\ P(wind=weak|No)}{P(today)}$$

$$P(No|today) = \frac{5}{14} * \frac{2}{5} * \frac{2}{5} * \frac{1}{5} * \frac{2}{5} = 0.00457$$

Normalize the data

$$P(yes|today) = \frac{0.02105}{0.02105 + 0.00457} = 0.8216$$

$$P(No|today) = \frac{0.00457}{0.02105 + 0.00457} = 0.1783$$

$$P(yes|today) > P(No|today)$$

So, For new data, we can play tennis,. i.e., P(yes)

Outlook	Yes	No	P(Yes)	P(No)
Sunny	3	2	3/9	2/5
Overcast	4	0	4/9	0/5
Rainy	2	3	2/9	3/5
Total	9	5	100%	100%

Temp	Yes	No	P(Yes)	P(No)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity	Yes	No	P(Yes)	P(No)
Normal	6	1	6/9	1/5
High	3	4	3/9	4/5
Total	9	5	100%	100%

Wind	Yes	No	P(Yes)	P(No)
Strong	3	3	3/9	3/5
Weak	6	2	6/9	2/5
Total	9	5	100%	100%

Example2: Naive Bayes Classifier for Discrete Data

- Apply naive Bayes classifier and to classify the new instance:

  (Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong)
- Predict the target value (yes or no)?
- First, the probabilities of the different target values over the 14 training examples

  P(P1ayTennis = yes) = 9/14 = 0.64 ;   P(P1ayTennis = no) = 5/14 = 0.36

$$P(yes|today) = \frac{P(yes) * P(outlook = sunny|yes)\ P(Temp = cool\ |yes)\ P(Humidity = high|yes)\ P(wind{=}strong|yes)}{P(today)}$$

- $P(yes|today) = \frac{9}{14} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} = ?$

$$P(No|today) = \frac{P(No) * P(outlook = sunny|No)\ P(Temp = cool|No)\ P(Humidity = high|No)\ P(wind{=}strong|No)}{P(today)}$$

- $P(No|today) = \frac{5}{14} * \frac{2}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} = ?$
- **Normalize the data**
- $P(yes|today) = ?$
- $P(No|today) = ?$

Outlook				
	Yes	No	P(Yes)	P(No)
Sunny	3	2	3/9	2/5
Overcast	4	0	4/9	0/5
Rainy	2	3	2/9	3/5
Total	9	5	100%	100%

Temp				
	Yes	No	P(Yes)	P(No)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity				
	Yes	No	P(Yes)	P(No)
Normal	6	1	6/9	1/5
High	3	4	3/9	4/5
Total	9	5	100%	100%

Wind				
	Yes	No	P(Yes)	P(No)
Strong	3	3	3/9	3/5
Weak	6	2	6/9	2/5
Total	9	5	100%	100%

Gaussian Naive Bayes Classifier for Continuous Data

- Continuous values associated with each feature are assumed to be distributed based on Gaussian distribution.
- Likelihood of the features is assumed to be Gaussian.
- So, conditional probability is given by: $P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}$
- Example:
- Classify whether a given person's datapoint is a male or a female.
- The features include height, weight, and foot size.

Gender	Height (ft)	Weight (lbs)	Foot size (inch)
Male	6	180	12
Male	5.92	190	11
Male	5.58	170	12
Male	5.92	165	10
Female	5	100	6
Female	5.5	150	8
Female	5.42	130	7
Female	5.75	150	9

- P(Male) = 4/8 = 0.5 ; P(Female) = 4/8 = 0.5

Male:

- Mean(height) = 5.855
- Variance(Height) = $\frac{\sum(x_i-\bar{x})^2}{n-1} = \frac{(6-5.855)^2+(5.92-5.855)^2+(5.58-5.855)^2+(5.92-5.855)^2}{4-1}$ =0.035055
- Calculate for all features

Gender	Mean (height)	Variance (Height)	Mean (weight)	Variance (Weight)	Mean (Footsize)	Variance (Footsize)
Male	5.855	0.035	176.25	122.92	11.25	0.91667
Female	5.4175	0.097225	132.5	558.33	7.5	1.6667

Gaussian Naive Bayes Classifier for Continuous Data

- Classify the new datapoint

Gender	Height (ft)	Weight (lbs)	Foot size (inch)
?	6	130	8

Gender	Mean (height)	Variance (Height)	Mean (weight)	Variance (Weight)	Mean (Footsize)	Variance (Footsize)
Male	5.855	0.035	176.25	122.92	11.25	0.91667
Female	5.4175	0.097225	132.5	558.33	7.5	1.6667

- $\text{P(male)} = \dfrac{\text{P(male)} * \text{P(H|male) P(W |male) P(F|male)}}{\text{Marginal probability or Evidence}}$
- $\text{P(female)} = \dfrac{\text{P(female)} * \text{P(H|female) P(W |female) P(F|female)}}{\text{Marginal probability or Evidence}}$

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}$$

- The evidence (normalizing constant) is the sum of the posteriors equals one.
- evidence=P(male)*P(ht|male)*P(wt|male)*P(footsize|male)+P(female)*P(ht|female)*P(wt|female)*P(foot size|female)
- The evidence may be ignored since it is a positive constant. (Normal distributions are always positive)

- $P(H|M) = \frac{1}{\sqrt{2*3.14*0.035033}} * e^{-\frac{(6-5.855)^2}{2*0.035033}}$ =1.5789 P(H|F) = 2.2356e^{-1}
- P(W|M) = 5.9881e^{-6} P(W|F)=1.6789e^{-2}
- P(Foot|M) = 1.3112e^{-3} P(Foot|F) = 2.8669e^{-1}

- $\text{P(male)} = \dfrac{\text{P(male)} * \text{P(H|male) P(W |male) P(F|male)}}{\text{Marginal probability or Evidence}}$=0.5*1.5789*5.9881e^{-6}*1.3112e^{-3}=6.1984e-9
- $\text{P(female)} = \dfrac{\text{P(female)} * \text{P(H|female) P(W |female) P(F|female)}}{\text{Marginal probability or Evidence}}$=0.5*2.2346e^{-1}*1.6789e^{-2}*2.8669e^{-1}=**5.377e^{-4}**
- **P(female)** > P(male)

References

1. Tom M. Mitchell, Machine Learning, McGraw Hill , 2017.

2. EthemAlpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2017.

3. Wikipedia