Why probabilities?

Why are we bringing here a Bayesian framework?

Recall Classification framework:

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n) / x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{Y}$.
Task: Learn a classification function: $f : \mathbb{R}^d \longrightarrow \mathbb{Y}$

Learn a mapping from x to y.
We would like to find this mapping $f(x) = y$ through $p(y|x)$!

Discriminative Algorithms

- **Discriminative Algorithms**:
  - Idea: model $p(y|x)$, conditional distribution of y given x.
  - In Discriminative Algorithms: find a decision boundary that separates positive from negative example.
  - To predict a new example, check on which side of the decision boundary it falls.
  - Model $p(y|x)$ directly.

Generative Algorithms

- **Generative Algorithms** adopt a different approach:
  - Idea: Build a model for what positive examples look like. Build a different model for what negative example look like.
  - To predict a new example, match it with each of the models and see which match is best.
  - Model $p(x|y)$ and $p(y)$!
  - Use Bayes rule to obtain $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$.

Generative Algorithms

- **Generative Algorithms** adopt a different approach:
  - Idea: Build a model for what positive examples look like. Build a different model for what negative example look like.
  - To predict a new example, match it with each of the models and see which match is best.
  - Model $p(x|y)$ and $p(y)$!
  - Use Bayes rule to obtain $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$.
  - To make a prediction:

$$\text{argmax}_y p(y|x) = \text{argmax}_y \frac{p(x|y)p(y)}{p(x)}$$

$$\text{argmax}_y p(y|x) \approx \text{argmax}_y p(x|y)p(y)$$

Naive Bayes Classifier

- Probabilistic model.
- Highly practical method.
- Application domains to natural language text documents.
- Naive because of the strong independence assumption it makes (not realistic).
- Simple model.
- Strong method can be comparable to decision trees and neural networks in some cases.

Setting

- A training data (x_i, y_i), x_i is a feature vector and y_i is a discrete label.
- d features, and n examples.
- Example: consider document classification, each example is a documents, each feature represents the presence or absence of a particular word in the document.
- We have a training set.
- A new example with feature values $x_{new} = (a_1, a_2, \cdots, a_d)$.
- We want to predict the label y_{new} of the new example.

Setting

$$y_{new} = \mathsf{argmax}_{y \in \mathbb{Y}} \quad p(y|a_1, a_2, \cdots, a_d)$$

Setting

$$y_{new} = \mathrm{argmax}_{y \in \mathbb{Y}} \quad p(y|a_1, a_2, \cdots, a_d)$$

Use Bayes rule to obtain:

$$y_{new} = \mathrm{argmax}_{y \in \mathbb{Y}} \quad \frac{p(a_1, a_2, \cdots, a_d|y) * p(y)}{p(a_1, a_2, \cdots, a_d)}$$

Setting

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y|a_1, a_2, \cdots, a_d)$$

Use Bayes rule to obtain:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; \frac{p(a_1, a_2, \cdots, a_d|y) * p(y)}{p(a_1, a_2, \cdots, a_d)}$$

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(a_1, a_2, \cdots, a_d|y) * p(y)$$

Setting

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y|a_1, a_2, \cdots, a_d)$$

Use Bayes rule to obtain:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; \frac{p(a_1, a_2, \cdots, a_d|y) * p(y)}{p(a_1, a_2, \cdots, a_d)}$$

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(a_1, a_2, \cdots, a_d|y) * p(y)$$

Can we estimate these two terms from the training data?

Setting

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y|a_1, a_2, \cdots, a_d)$$

Use Bayes rule to obtain:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; \frac{p(a_1, a_2, \cdots, a_d|y) * p(y)}{p(a_1, a_2, \cdots, a_d)}$$

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(a_1, a_2, \cdots, a_d|y) * p(y)$$

Can we estimate these two terms from the training data?

1. $p(y)$ can be easy to estimate: count the frequency with which each label y occurs in the training data.

Setting

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y|a_1, a_2, \cdots, a_d)$$

Use Bayes rule to obtain:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; \frac{p(a_1, a_2, \cdots, a_d|y) * p(y)}{p(a_1, a_2, \cdots, a_d)}$$

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(a_1, a_2, \cdots, a_d|y) * p(y)$$

Can we estimate these two terms from the training data?

1. $p(y)$ can be easy to estimate: count the frequency with which each label y.
2. $p(a_1, a_2, \cdots, a_d|y)$ is not easy to estimate unless we have a very very large sample. (We need to see every example many times to get reliable estimates)

Naive Bayes Classifier

Makes a simplifying assumption that the feature values are conditionally independent given the label.
Given the label of the example, the probability of observing the conjunction $a_1, a_2, \cdots, a_d$ is the product of the probabilities for the individual features:

$$p(a_1, a_2, \cdots, a_d | y) = \prod_j p(a_j | y)$$

Naive Bayes Classifier

Makes a simplifying assumption that the feature values are conditionally independent given the label.

Given the label of the example, the probability of observing the conjunction $a_1, a_2, \cdots, a_d$ is the product of the probabilities for the individual features:

$$p(a_1, a_2, \cdots, a_d | y) = \prod_j p(a_j | y)$$

Naive Bayes Classifier:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y) \prod_j p(a_j | y)$$

Naive Bayes Classifier

Makes a simplifying assumption that the feature values are conditionally independent given the label.

Given the label of the example, the probability of observing the conjunction $a_1, a_2, \cdots, a_d$ is the product of the probabilities for the individual features:

$$p(a_1, a_2, \cdots, a_d | y) = \prod_j p(a_j | y)$$

Naive Bayes Classifier:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y) \prod_j p(a_j | y)$$

Can we estimate these two terms from the training data?

Naive Bayes Classifier

Makes a simplifying assumption that the feature values are conditionally independent given the label.

Given the label of the example, the probability of observing the conjunction $a_1, a_2, \cdots, a_d$ is the product of the probabilities for the individual features:

$$p(a_1, a_2, \cdots, a_d | y) = \prod_j p(a_j | y)$$

Naive Bayes Classifier:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} \;\; p(y) \prod_j p(a_j | y)$$

Can we estimate these two terms from the training data?

Yes!

Algorithm

Learning: Based on the frequency counts in the dataset:

1. Estimate all $p(y)$, $\forall y \in \mathbb{Y}$.
2. Estimate all $p(a_j|y)$ $\forall y \in \mathbb{Y}$, $\forall a_i$.

Classification: For a new example, use:

$$y_{new} = \text{argmax}_{y \in \mathbb{Y}} p(y) \prod_j p(a_j|y)$$

Note: No model per se or hyperplane, just count the frequencies of various data combinations within the training examples.

Example

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

Can we predict the class of the new example?

Example

$$y_{new} = \text{argmax}_{y \in \{yes,no\}} p(y) * p(Masters|y) * p(UX\ \ Design|y) * p(Java|y) * p(TRUE|y)$$

$$p(yes) = 8/14 = 0.572$$

$$p(no) = 6/14 = 0.428$$

Conditional probabilities:

$$\begin{aligned} p(masters|yes) &= 4/8 & p(masters|no) &= 1/6 \\ p(UX\ \ Design|yes) &= 2/8 & p(UX\ \ Design|no) &= 2/6 \\ p(Java|yes) &= 6/8 & p(Java|no) &= 1/6 \\ p(TRUE|yes) &= 4/8 & p(TRUE|no) &= 3/6 \end{aligned}$$

$$p(yes) * p(Masters|yes) * p(UX\ \ Design|yes) * p(Java|yes) * p(TRUE|yes) = 0.026$$

$$p(no) * p(Masters|no) * p(UX\ \ Design|no) * p(Java|no) * p(TRUE|no) = 0.002$$

$$y_{new} = yes$$

Text Classification

- Given a document (corpus), define an attribute for each word position in the document.
- The value of the attribute is the English word in that position.
- To reduce the number of probabilities that needs to be estimated, besides NB independence assumption, we assume that: The probability of a given word w_k occurrence is independent of the word position within the text. That is:

$$p(x_1 = w_k|c_j), p(x_2 = w_k|c_j), \cdots$$

estimated by:

$$p(w_k|c_j)$$