10-701
Machine Learning

Naïve Bayes classifiers

Types of classifiers

- We can divide the large variety of classification approaches into three major types

1. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

2. Generative:
   - build a generative statistical model
   - e.g., Bayesian networks

3. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., decision tree

Naïve Bayes Classifier

- Naïve Bayes classifiers assume that given the class label (Y) the attributes are **conditionally independent** of each other:

$$X = \begin{bmatrix} x^1 \\ \vdots \\ x^n \end{bmatrix}$$

$$p(X \mid y) = \prod_j p_j(x^j \mid y)$$

Product of probability terms

Specific model for attribute *j*

- Using this idea the full classification rule becomes:

$$\begin{aligned}
\hat{y} &= \arg\max_v \, p(y = v \mid X) \\
&= \arg\max_v \, \frac{p(X \mid y = v)\, p(y = v)}{p(X)} \\
&= \arg\max_v \, \prod_j p_j(x^j \mid y = v) p(y = v)
\end{aligned}$$

v are the classes we have

Conditional likelihood: Full version

$$L(X_i \mid y_i = 1, \Theta) = \prod_j p(x_i^j \mid y_i = 1, \theta_1^j)$$

Vector of binary attributes for sample *i*

The set of all parameters in the NB model

The specific parameters for attribute *j* in class 1

Note the following:

1. We assume conditional independence between attributes **given** the class label
2. We learn a **different** set of parameters for the two classes (class 1 and class 2).

Learning parameters

$$L(X_i \mid y_i = 1, \Theta) = \prod_j p(x_i^j \mid y_i = 1, \theta_1^j)$$

- Let X_1 … X_{k1} be the set of input samples with label ‘y=1’
- Assume all attributes are **binary**
- To determine the MLE parameters for $p(x^j = 1 \mid y = 1)$ we simply count how many times the j’th entry of those samples in class 1 is 0 (termed n0) and how many times its 1 (n1). Then we set:

$$p(x^j = 1 \mid y = 1) = \frac{n1}{n0 + n1}$$

Final classification

- Once we computed all parameters for attributes in both classes we can easily decide on the label of a **new** sample X.

Can be easily be extended to multi-class classification

$$\hat{y} = \arg\max_v p(y = v \mid X)$$

$$= \arg\max_v \frac{p(X \mid y = v) p(y = v)}{p(X)}$$

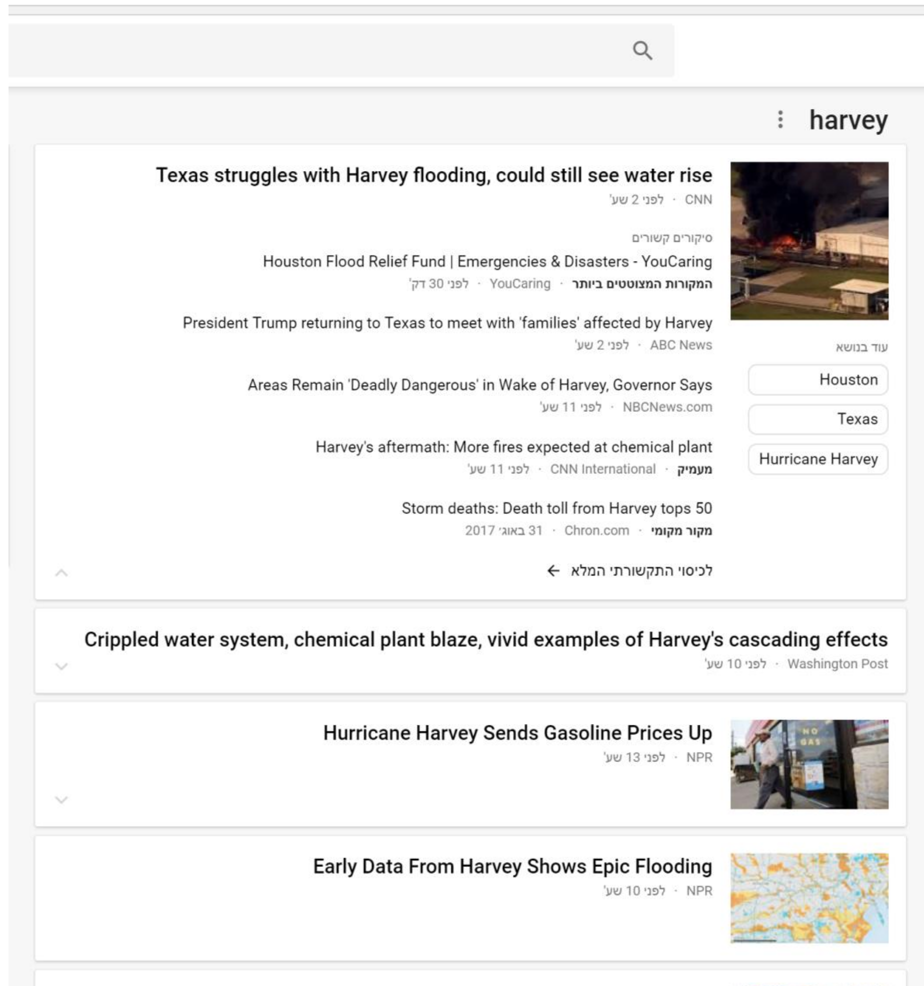
$$= \arg\max_v \prod_j p_j(x^j \mid y = v) p(y = v)$$

Perform this computation for both class 1 and class 2 and select the class that leads to a higher probability as your decision

Prior on the prevalence of samples from each class

Example: Text classification

- Text classification is all around us



Feature transformation

- How do we encode the set of features (words) in the document?
- What type of information do we wish to represent? What can we ignore?
- Most common encoding: '**Bag of Words**'
- Treat document as a collection of words and encode each document as a vector based on some dictionary
- The vector can either be binary (present / absent information for each word) or discrete (number of appearances)

- Google is a good example
- Other applications include job search adds, spam filtering and many more.

Feature transformation: Bag of Words

- In this example we will use a binary vector
- For document X_i we will use a vector of *m** indicator features $\{\phi^j(X_i)\}$ for whether a word appears in the document
  - $\phi^j(X_i) = 1$, if word *j* appears in document X_i;
    $\phi^j(X_i) = 0$ if it does not appear in the document
- $\Phi(X_i) = [\phi^1(X_i) \ldots \phi^m(X_i)]^T$ is the resulting feature vector for the entire dictionary for document X_i
- For notational simplicity we will replace each document X_i with a fixed length vector $\Phi_i = [\phi^1 \ldots \phi^m]^T$, where $\phi^j = \phi^j(X_i)$.

*The size of the vector for English is usually ~10000 words

Naïve Bayes classifiers for continuous values

- So far we assumed a binomial or discrete distribution for the data given the model ($p(X_i|y)$)
- However, in many cases the data contains continuous features:
  - Height, weight
  - Levels of genes in cells
  - Brain activity
- For these types of data we often use a Gaussian model
- In this model we assume that the observed input vector X is generated from the following distribution

$$X \sim N(\mu, \Sigma)$$

Gaussian Bayes Classifier Assumption

- The i'th record in the database is created using the following algorithm

1. Generate the output (the "class") by drawing $y_i \sim Multinomial(p_1, p_2, \ldots p_{Ny})$
2. Generate the inputs from a Gaussian PDF that depends on the value of y_i :

$$\mathbf{x}_j \sim N(\boldsymbol{\mu}_i, \Sigma_i).$$

Gaussian Bayes Classification

$$P(y=v \mid X) = \frac{p(X \mid y=v)P(y=v)}{p(X)}$$

- To determine the class when using the Gaussian assumption we need to compute p(X|y):

$$P(X \mid y) = \frac{1}{(2\pi)^{n/2} \mid \Sigma \mid^{1/2}} \exp\left[-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)\right]$$

Once again, we need lots of data to compute the values of the mean μ and the covariance matrix Σ

Gaussian Bayes Classification

- Here we can also use the Naïve Bayes assumption: Attributes are independent given the class label
- In the Gaussian model this means that the covariance matrix becomes a **diagonal matrix** with zeros everywhere except for the diagonal
- Thus, we only need to learn the values for the variance term for each attribute in each class: $x^j \sim N(\mu_v^j, \sigma_v^j)$

$$P(X|y=v) = \prod_j P(x^j|y=v) = \prod_j \frac{1}{(2\pi)^{1/2}\sigma_v^j} exp\left[-\frac{\left(x^j - \mu_v^j\right)^2}{2\left(\sigma_v^j\right)^2}\right]$$

Separate means and variance for each class

MLE for Gaussian Naïve Bayes Classifier

- For each class we need to estimate one global value (prior) and two values for each feature (mean and variance)
- The prior is computed in the same way we did before (counting) which is the MLE estimate
- Let the numbers of input samples in class 1 be k1. The MLE for mean and variance is computed by setting:

$$\mu_1^j = \sum_{i\ s.t.\ y_i=1} \frac{x_i^j}{k1} \qquad {\sigma_1^j}^2 = \sum_{i\ s.t.\ y_i=1} \frac{(x_i^j - \mu_1^j)^2}{k1}$$

Possible problems with Naïve Bayes classifiers: Assumptions

- In most cases, the assumption of conditional independence given the class label is violated

  - much more likely to find the word 'Donald' if we saw the word 'Trump' regardless of the class
- This is, unfortunately, a major shortcoming which makes these classifiers inferior in many real world applications (though not always)
- There are models that can improve upon this assumption without using the full conditional model (one such model are Bayesian networks which we will discuss later in this class).

Important points

- Problems with estimating full joints
- Advantages of Naïve Bayes assumptions
- Applications to discrete and continuous cases
- Problems with Naïve Bayes classifiers