Security Analytics

Topic 5: Probabilistic Classification Models: Naïve Bayes

Purdue University
Prof. Ninghui Li
Based on slides by Prof. Jenifer Neville and Chris Clifton

Readings

- Principle of Data Mining
  - Chapter 10: Predictive Modeling for Classification
    - 10.8 The Naïve Bayes Model
- From Speech and Language Processing. Daniel Jurafsky & James H. Martin
  - Chapter 6: Naive Bayes and Sentiment Classification
  - https://web.stanford.edu/~jurafsky/slp3/6.pdf

The Classification Problem

- Given input **x**, the goal is to predict *y*, which is a categorical variable
  - *y* is called the class label
  - **x** is the feature vector
- Example:
  - **x:** monthly income and bank saving amount;
    y: risky or not risky
  - **x**: bag-of-words representation of an email;
    y: spam or not spam

Precision and Recall

- Given a dataset, we train a classifier that gets 99% accuracy
- **Did we do a good job?**
- Build a classifier for brain tumor:
  - 99.9% of brain scans do not show signs of tumor
  - ***Did we do a good job?***
- By simply saying "NO" to all examples we reduce the error by a factor of 10!
  - ***Clearly Accuracy is not the best way to evaluate the learning system when the data is heavily skewed!***
- **Intuition**: we need a measure that captures the class we care about! (rare)

Precision and Recall

- The learner can make two kinds of mistakes:
  - False Positive
  - False Negative

	True Label 1	True Label 0
Predicted 1	True Positive	False Positive
Predicted 0	False Negative	True Negative

- **Precision**: $$\frac{\text{True Pos}}{\text{Predicted Pos}} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Pos}}$$
- *"when we predicted the rare class, how often are we right?"*
- **Recall** $$\frac{\text{True Pos}}{\text{Actual Pos}} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Neg}}$$
- "Out of all the instances of the rare class, how many did we catch?"

Precision and Recall

- **Precision and Recall give us two reference points to compare learning performance**

	Precision	Recall
Algorithm 1	0.5	0.4
Algorithm 2	0.7	0.1
Algorithm 3	0.02	1

- *Which algorithm is better?*

We need a single score

- Option 1: **Average** $$\frac{P+R}{2}$$
- Option 2: **F-Score** $$2\frac{PR}{P+R}$$

Properties of f-score:

- Ranges between 0-1
- Prefers precision and recall with similar values

NAÏVE BAYES CLASSIFIER

Example

- Example: Play Tennis

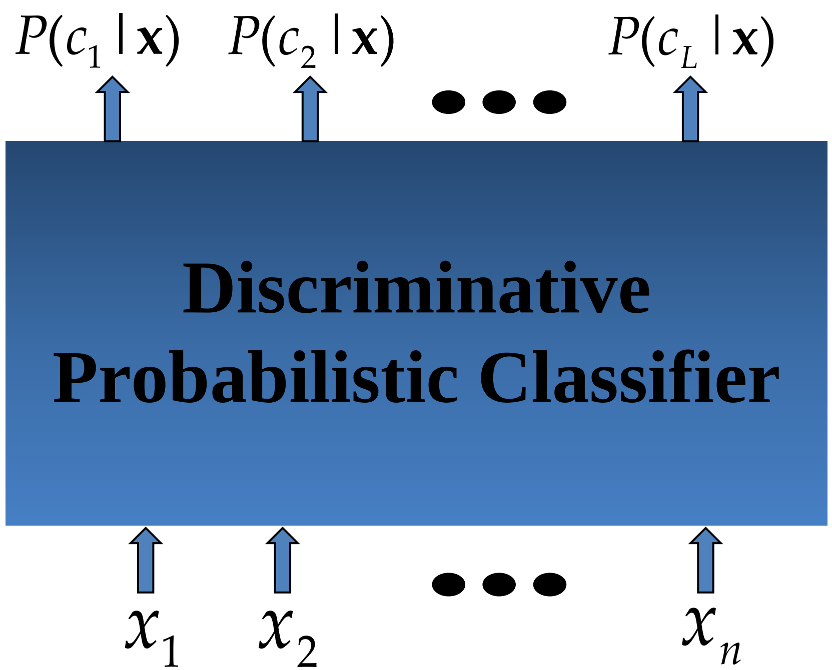
PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Probabilistic Classification

- Establishing a probabilistic model for classification
  - **Discriminative model**

$$\boxed{P(c \mid \mathbf{x}) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)}$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

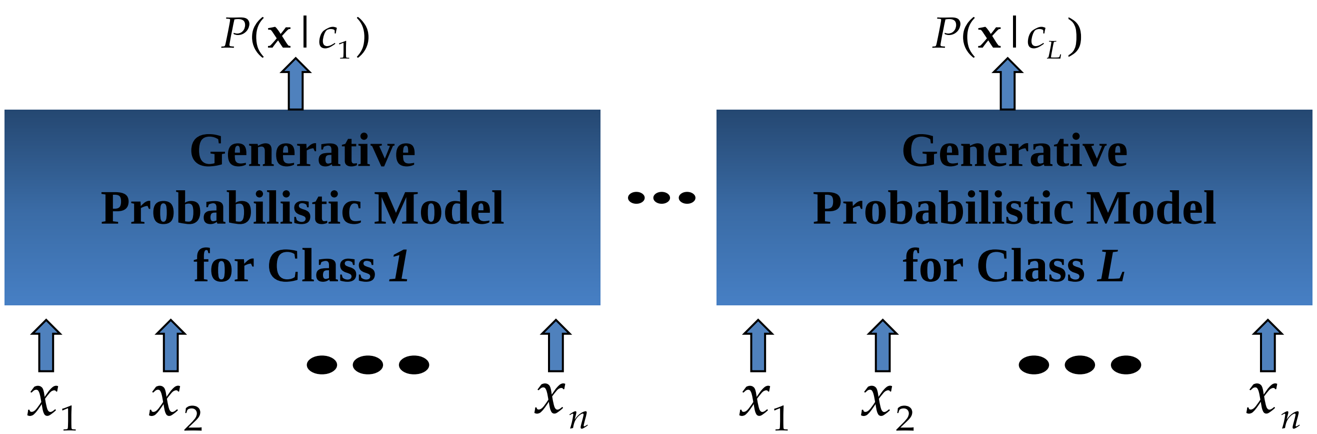
- To train a discriminative classifier regardless its probabilistic or non-probabilistic nature, all training examples of different classes must be jointly used to build up a single discriminative classifier.
- Output L probabilities for L class labels in a probabilistic classifier while a single label is achieved by a non-probabilistic classifier .
- Example: Logistic Regression, SVM, etc.

Probabilistic Classification

Establishing a probabilistic model for classification (cont.)

– **Generative model (must be probabilistic)**

$$P(\mathbf{x}|c) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

- L probabilistic models have to be trained independently
- Each is trained on only the examples of the same label
- Output L probabilities for a given input with L models
- "Generative" means that such a model produces data subject to the distribution via sampling.

Bayes rule for probabilistic classifier

- The learner considers a set of candidate labels, and attempts to find the most probable one $y \in Y$, given the observed data.

- Such maximally probable assignment is called maximum a posteriori assignment (**MAP**); Bayes theorem is used to compute it:

$$y_{MAP} = \mathrm{argmax}_{y \in Y} P(y|x) = \mathrm{argmax}_{y \in Y} P(x|y)\, P(y)/P(x)$$

$$= \mathrm{argmax}_{y \in Y} P(x|y)\, P(y)$$

Since $P(x)$ is the same for all $y \in Y$

Bayes Classifier

Maximum **A** **P**osterior (**MAP**) classification rule

For an input $\boldsymbol{x}$, find the largest one from L probabilities output by a discriminative probabilistic classifier

$$P(c_1 \mid \mathbf{x}), \ldots, P(c_L \mid \mathbf{x}).$$

Assign x to label c^* if $P(c^* \mid \mathbf{x})$ is the largest.

- Generative classification with the MAP rule
  - Apply Bayesian rule to convert them into posterior probabilities

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i) P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} \mid c_i) P(c_i)$$

$$\text{for } i = 1, 2, \cdots, L$$

Common factor for all L probabilities

  - Then apply the MAP rule to assign a label

Naïve Bayes

- Bayes classification

$$P(c\,|\,\mathbf{x}) \propto P(\mathbf{x}\,|\,c)P(c) = P(x_1,\cdots,x_n\,|\,c)P(c) \ \text{ for } c = c_1,\ldots,c_L.$$

Difficulty: learning the joint probability $P(x_1,\cdots,x_n\,|\,c)$ is infeasible!

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

$$P(x_1,x_2,\cdots,x_n\,|\,c) = \underline{P(x_1\,|\,x_2,\cdots,x_n,c)}P(x_2,\cdots,x_n\,|\,c)$$

Applying the independence assumption

$$= \underline{P(x_1\,|\,c)}P(x_2,\cdots,x_n\,|\,c)$$

$$= P(x_1\,|\,c)P(x_2\,|\,c)\cdots P(x_n\,|\,c)$$

  - Apply the MAP classification rule: assign $\mathbf{x}' = (a_1,a_2,\cdots,a_n)$ to c^*

$$\underbrace{[P(a_1\,|\,c^*)\cdots P(a_n\,|\,c^*)]}_{\text{estimate of } P(a_1,\cdots,a_n\,|\,c^*)}P(c^*) > \underbrace{[P(a_1\,|\,c)\cdots P(a_n\,|\,c)]}_{\text{esitmate of } P(a_1,\cdots,a_n\,|\,c)}P(c), \quad c \neq c^*, c = c_1,\cdots,c_L$$

Naïve Bayes

- Algorithm: Discrete-Valued Features
  - Learning Phase: Given a training set S of F features and L classes,

    For each target value of c_i $(c_i = c_1, \cdots, c_L)$

    $\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S;

    For every feature value x_{jk} of each feature x_j $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

    $\hat{P}(x_j = x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in S;

    Output: $F * L$ conditional probabilistic (generative) models
  - Test Phase: Given an unknown instance $\mathbf{x}' = (a'_1, \cdots, a'_n)$

    "Look up tables" to assign the label c^* to $\mathbf{X}'$ if

    $$[\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 \mid c_i) \cdots \hat{P}(a'_n \mid c_i)]\hat{P}(c_i), \quad c_i \neq c^*, c_i = c_1, \cdots, c_L$$

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example

- Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14 \quad P(\text{Play}=No) = 5/14$

Example

- Test Phase
  - Given a new instance, predict its label

$\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

  - Look up tables achieved in the learning phrase

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

  - Decision making with the MAP rule

$P(Yes|\mathbf{x}') \approx [P(Sunny|\mathrm{Yes})P(Cool|Yes)P(High|\mathrm{Yes})P(Strong|Yes)]P(\mathrm{Play}{=}Yes) = 0.0053$

$P(No|\mathbf{x}') \approx [P(Sunny|\mathrm{No})\ P(Cool|\mathrm{No})P(High|No)P(Strong|No)]P(\mathrm{Play}{=}No) = 0.0206$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

Zero conditional probability

- If no example contains the feature value
  - In this circumstance, we face a zero conditional probability problem during test

$$\hat{P}(x_1 | c_i) \cdots \hat{P}(a_{jk} | c_i) \cdots \hat{P}(x_n | c_i) = 0 \quad \text{for } x_j = a_{jk},\ \hat{P}(a_{jk} | c_i) = 0$$

  - For a remedy, class conditional probabilities re-estimated with

$$\hat{P}(a_{jk} | c_i) = \frac{n_c + mp}{n + m}$$

(m-estimate)

n_c : number of training examples for which $x_j = a_{jk}$ and $c = c_i$

n : number of training examples for which $c = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of x_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)

Zero conditional probability

Example: P(outlook=overcast|no)=0 in the play-tennis dataset

Adding m "virtual" examples (m: up to 1% of #training example)

- In this dataset, # of training examples for the "no" class is 5.
- We can only add m**=1** "virtual" example in our m-esitmate remedy.

– The "outlook" feature can takes only 3 values. So $p=1/3$.

– Re-estimate P(outlook|no) with the m-estimate

$$P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}$$

$$P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9} \qquad P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}$$

Numerical Stability

- Recall: NB classifier: $$\propto \prod_{i=1}^{m} P(X_i|Y)P(Y)$$
  - **Multiplying probabilities can get us into problems!**
  - Imagine computing the probability of 2000 independent coin flips
  - Most programming environments: $(.5)^{2000}=0$

Numerical Stability

- Our problem: **Underflow Prevention**
- Recall: *log(xy) = log(x) + log(y)*
- better to sum logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

Naïve Bayes: Dealing with Continuous-valued Features

When facing a continuous-valued feature

Conditional probability often modeled with the normal distribution

$$\hat{P}(x_j \mid c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

μ_{ji} : mean (avearage) of feature values x_j of examples for which $c = c_i$

σ_{ji} : standard deviation of feature values x_j of examples for which $c = c_i$

- Learning Phase: $\text{for } \mathbf{X} = (X_1, \cdots, X_n),\ C = c_1, \cdots, c_L$

  Output: $n \times L$ normal distributions and $P(C = c_i)\ i = 1, \cdots, L$
- Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$
  - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
  - Apply the MAP rule to assign a label (the same as the discrete case)

Naïve Bayes

- Example: Continuous-valued Features
  - Temperature is naturally of continuous value.

Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

No: 27.3, 30.1, 17.4, 29.5, 15.1

  - Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \quad \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \quad \sigma_{No} = 7.09$$

  - **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x\,|\,Yes) = \frac{1}{2.35\sqrt{2\pi}}\exp\left(-\frac{(x-21.64)^2}{2\times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}}\exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x\,|\,No) = \frac{1}{7.09\sqrt{2\pi}}\exp\left(-\frac{(x-23.88)^2}{2\times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}}\exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

Summary

- Naïve Bayes: the conditional independence assumption
  - Training and test are very efficient
  - Two different data types lead to two different learning algorithms
  - Working well sometimes for data violating the assumption!
- A popular generative model
  - Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption
  - Many successful applications, e.g., spam mail filtering
  - A good candidate of a base learner in ensemble learning
  - Apart from classification, naïve Bayes can do more…