Naive Bayes

Victor Lavrenko and Nigel Goddard
School of Informatics

Overview

- Naive Bayes classifier
  - components and their function
  - independence assumption
  - dealing with missing data
- Continuous example
- Discrete example
- Pros and cons

Bayesian classification

- Goal: learning function \( f(x) \rightarrow y \)
  - \( y \): one of \( k \) classes (e.g., spam/ham, digit 0-9)
  - \( x = x_1, x_2, \ldots \): values of attributes (numeric or categorical)
  - Probabilistic classification:
    - most probable class given observation: \( \hat{y} = \arg \max_y P(y|x) \)
  
  - Bayesian probability of a class:
    \[
    P(y|x) = \frac{P(x|y)P(y)}{\sum_y P(x|y')P(y')}
    \]

Bayesian classification: components

\[
P(y|x) = \frac{P(x|y)P(y)}{\sum_y P(x|y')P(y')}
\]

- \( P(y) \): prior probability of each class
  - encodes how much classes are common, which are rare
  - apriori much more likely to have common cold than Avian flu
- \( P(x|y) \): class-conditional model
  - describes how likely to see observation \( x \) for class \( y \)
  - assuming it’s Avian flu, do the symptoms look plausible?
- \( P(x) \): normalize probabilities across observations
  - does not affect which class is most likely (arg max)

Bayesian classification: normalization

- Normalizer: \( P(x) = \sum_y P(x|y)P(y) \)
- an “outlier” has a low probability under every class
  \[
P(X=x_0 \mid Y=3) < P(X=x_0 \mid Y=2)
\]

Naive Bayes: a generative model

- A complete probability distribution for each class
  - defines likelihood for any point \( x \)
  - \( P(\text{class} \mid \text{observation}) \)
  - can “generate” synthetic observations
  - will share many properties of the original data
- Not all probabilistic classifiers do this
  - possible to estimate \( P(y|x) \) directly
  - e.g., logistic regression:
  \[
P(y|x) = \frac{1}{1 + \exp \left( -\sum \lambda_i g(y, x) \right)}
\]

Independence assumption

- Compute \( P(x_1, x_2, x_3|y) \) for every observation \( x_1, x_2, x_3 \)
  - class-conditional “counts”, based on training data
  - problem: may not have seen every \( x_1, x_2, x_3 \) for every \( y \)
    - digits: \( 2^{10} \) possible black/white patterns (20x20)
    - spam: every possible combination of words: \( 2^{10^{10}} \)
  - often have observations for individual \( x \) for every \( y \)
- Idea: assume \( x_1, x_2, x_3 \) conditionally independent given \( y \)
  \[
P(x_1, x_2, x_3|y) = \prod_{i=1}^3 P(x_i|y)
\]

Conditional independence

- Probabilities going to the beach and getting a heat stroke are not independent:
  \( P(B, S) > P(B)P(S) \)
- May be independent if we know the weather is hot
  \( P(B, S|H) = P(B|H)P(S|H) \)
- Hot weather "explains" all the dependence between beach and heatstroke
- In classification:
  - class-value explains all the dependence between attributes

Overview

- Naive Bayes classifier
  - Continuous example
  - general concepts
  - working example
  - example of failure
- Discrete example
  - general concepts
  - problems with Naive Bayes
- Pros and cons
Continuous example
- Distinguish children from adults based on size:
  - class: $\{a, c\}$, attributes: height [cm], weight [kg]
  - training examples: $(h_1, w_1, y_1), \ldots, (h_{12}, w_{12}, y_{12})$, 4 adults, 12 children
  - Class probabilities: $P(a) = \frac{4}{12} = 0.33 / P(c) = 0.67$
- Model for adults:
  - height: Gaussian with mean, variance $\mu_a, \sigma_a$
  - weight: Gaussian $\mu_a, \sigma_a$
- Assume height and weight independent
- Model for children: same, using $(\mu_a, \sigma_a)$, $(\mu_a, \sigma_a)$

Discrete example: spam
- Separate spam from valid email, attributes = words
  - D1: "send us your password" spam
  - D2: "send us your review" ham
  - D3: "review your password" ham
  - D4: "review us" spam
  - D5: "send your password" spam
  - D6: "send your account" spam

New email: "review us now!"

Problems with Naive Bayes
- Zero-frequency problem
  - any mail containing "account" is spam: $P(\text{ham}|\text{account}) = 0.02$
  - solution: never allow zero probabilities
- Laplace smoothing: add a small positive number to all counts:
  - $P(w|c) = \frac{\text{num}(w, c) + 1}{\text{num}(c) + 2}$
- May use global statistics in place of true $P(w|c)$
- Very common problem (Zipf’s law: 50% most words occur once)
- Assumes word independence
  - every word contributes independently to $P(\text{spam}|\text{email})$
  - fooling NB: add lots of "hammy" words to spam email

Overview
- Naive Bayes classifier
  - Continuous example
  - Discrete example
  - Pros and Cons
    - dealing with missing data
    - computational cost and incremental updates

Missing data
- Suppose don’t have value for some attribute $X_j$
  - applicant’s credit history unknown
  - some medical test not performed on patient
  - how to compute $P(X_1, \ldots, X_n | y)$
- Easy with Naive Bayes
  - ignore attribute in instance where its value is missing
  - compute likelihood based on observed attributes
  - no need to "fill in" or explicitly model missing values
  - based on conditional independence between attributes

Decision boundary
- Different means, same variance: straight line / plane
- Same mean, different variance: circle / ellipse
- General case: parabolic curve

Problems with Naive Bayes
- Zero-frequency problem
  - any mail containing "account" is spam: $P(\text{account}|\text{ham}) = 0.01$
  - solution: never allow zero probabilities
- Laplace smoothing: add a small positive number to all counts:
  - $P(w|c) = \frac{\text{num}(w, c) + 1}{\text{num}(c) + 2}$
- May use global statistics in place of true $P(w|c)$
- Very common problem (Zipf’s law: 50% most words occur once)
- Assumes word independence
  - every word contributes independently to $P(\text{spam}|\text{email})$
  - fooling NB: add lots of "hammy" words to spam email

Missing data (2)
- Ex: three coin tosses: Event: $\{X_1\cap H, X_2\cap \overline{T}, X_3 \cap T\}$
  - event = head, unknown (either head or tail), tail
  - $P(\overline{h}, T) = P(\overline{h}, \overline{T}) + P(\overline{h}, T)$
  - General case: $X_j$ has missing value
    - $P(x_j | \ldots, x_j, y) = P(x_j | y) \cdot \frac{P(x_j | y)}{P(x_j | y)}$
    - $\sum_i P(x_i, \ldots, x_j, y) = P(x_i) \cdots P(x_j | y)$
Summary

- Naive Bayes classifier
  - explicit handles class priors
  - “normalizes” across observations; outliers comparable
  - assumption: all dependence is “explained” by class label
- Continuous example
  - unable to handle correlated data
- Discrete example
  - cannot be handled by Naive Bayes
  - must deal with zero-frequency problem
- Pros:
  - handles missing data
  - good computational complexity
  - incremental updates

Computational complexity

- One of the fastest learning methods
- O(nd+cd) training time complexity
  - c ... number of classes
  - n ... number of instances
  - d ... number of dimensions (attributes)
  - both training and prediction
  - no hidden constants (number of iterations, etc.)
  - testing: O(nc)
- O(dc) space complexity
  - only decision trees are more compact

Incremental updates

- Allows incremental updates: O(d) insertion / deletion
- Binomial: store raw counts instead of probabilities
  - new example of class c:
    - \( \hat{n}_{ci} = x_i \) for each \( x_i \) in example, \( n_i = \hat{n}_i = 1 \)
  - when need to classify:
    - \( P(x_i | c) = (\hat{n}_{ci} + v) / (n_i + 2v) \)
    - \( P(c) = n_i / n \)
- Gaussian: store partial sums instead of mean/variance
  - \( \hat{s}_{ci} = x_i \) \( \hat{s}_{ci}^2 = x_i^2 \)
  - when need to classify:
    - mean = \( \hat{s}_{ci} / n \)
    - variance = \( \hat{s}_{ci}^2 / n - \text{mean}^2 \)

General structure for Naive Bayes

- Task
  - c-class classification \((c \geq 2)\)
- Model structure
  - \( c \times d \) independent distributions
  - continuous: Gaussian, discrete: Bernoulli
- Score function
  - class-conditional likelihood
- Optimization / search method
  - analytic solution
- Book: section 4.2