

Text Categorization: Discriminative Classifiers

Part 1

ChengXiang “Cheng” Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Overview

- What is text categorization?
- Why text categorization?
- How to do text categorization?
 - Generative probabilistic models
 - **Discriminative approaches**
- How to evaluate categorization results?

Anatomy of Naïve Bayes Classifier

Two categories: θ_1 and θ_2

$$\text{score}(d) = \log \frac{p(\theta_1 | d)}{p(\theta_2 | d)} = \log \frac{p(\theta_1) \prod_{w \in V} p(w | \theta_1)^{c(w,d)}}{p(\theta_2) \prod_{w \in V} p(w | \theta_2)^{c(w,d)}}$$

$$= \log \frac{p(\theta_1)}{p(\theta_2)} + \sum_{w \in V} c(w,d) \log \frac{p(w | \theta_1)}{p(w | \theta_2)}$$

Category bias (β_0) doesn't depend on d !

Sum over all words (features $\{x_i\}$)

Weight on each word (feature) β_i

Feature value: $x_i = c(w,d)$



Generalize

$$d = (x_1, x_2, \dots, x_M), \quad x_i \in \mathcal{R}$$

$$\text{score}(d) = \beta_0 + \sum_{i=1}^M x_i \beta_i \quad \beta_i \in \mathcal{R}$$

= Logistic Regression!

Discriminative Classifier 1: Logistic Regression

Binary Response Variable: $Y \in \{0,1\}$

Predictors: $X = (x_1, x_2, \dots, x_M)$, $x_i \in \mathbb{R}$

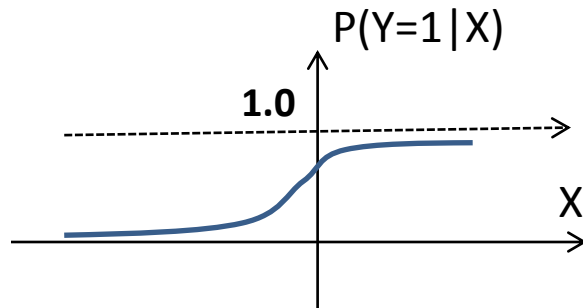
$$Y = \begin{cases} 1 & \text{category}(d) = \theta_1 \\ 0 & \text{category}(d) = \theta_2 \end{cases}$$

Modeling $p(Y|X)$ directly

Allow many other features than words!

$$\log \frac{p(\theta_1 | d)}{p(\theta_2 | d)} = \log \frac{p(Y = 1 | X)}{p(Y = 0 | X)} = \log \frac{p(Y = 1 | X)}{1 - p(Y = 1 | X)} = \beta_0 + \sum_{i=1}^M x_i \beta_i \quad \beta_i \in \mathbb{R}$$

$$p(Y = 1 | X) = \frac{e^{\beta_0 + \sum_{i=1}^M x_i \beta_i}}{e^{\beta_0 + \sum_{i=1}^M x_i \beta_i} + 1}$$



Estimation of Parameters

- Training Data: $T = \{(X_i, Y_i)\}, i=1, 2, \dots, |T|$
- Parameters: $\vec{\beta} = (\beta_0, \beta_1, \dots, \beta_M)$
- Conditional likelihood: $p(T | \vec{\beta}) = \prod_{i=1}^{|T|} p(Y = Y_i | X = X_i, \vec{\beta})$

$Y_i = 1$

$$p(Y = 1 | X) = \frac{e^{\beta_0 + \sum_{i=1}^M x_i \beta_i}}{e^{\beta_0 + \sum_{i=1}^M x_i \beta_i} + 1}$$

$Y_i = 0$

$$p(Y = 0 | X) = \frac{1}{e^{\beta_0 + \sum_{i=1}^M x_i \beta_i} + 1}$$

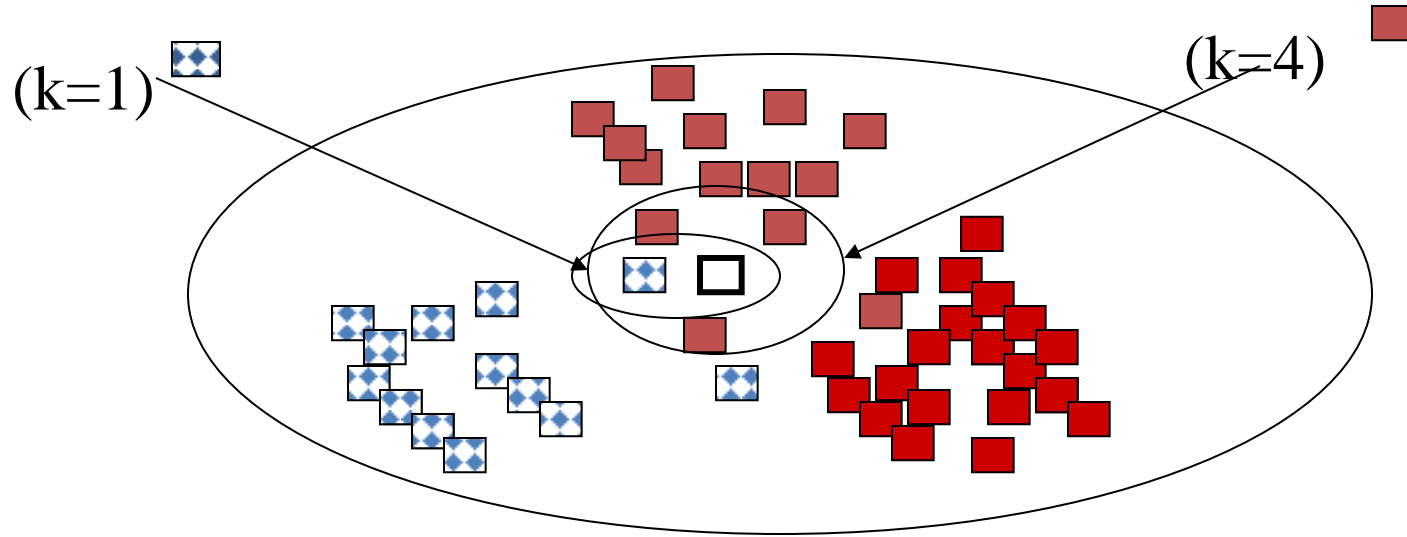
- Maximum Likelihood estimate $\vec{\beta}^* = \arg \max_{\vec{\beta}} p(T | \vec{\beta})$

Can be computed in many ways (e.g., Newton's method)

Discriminative Classifier 2: K-Nearest Neighbors (K-NN)

- Find k examples in the training set that are most similar to the text object to be classified (“neighbor” documents)
- Assign the category that is most common in these neighbor text objects (neighbors vote for the category)
- Can be improved by considering the distance of a neighbor (a closer neighbor has more influence)
- Can be regarded as a way to directly estimate the conditional probability of label given data instance, i.e., $p(Y|X)$
- Need a similarity function to measure similarity of two text objects

Illustration of K-NN Classifier

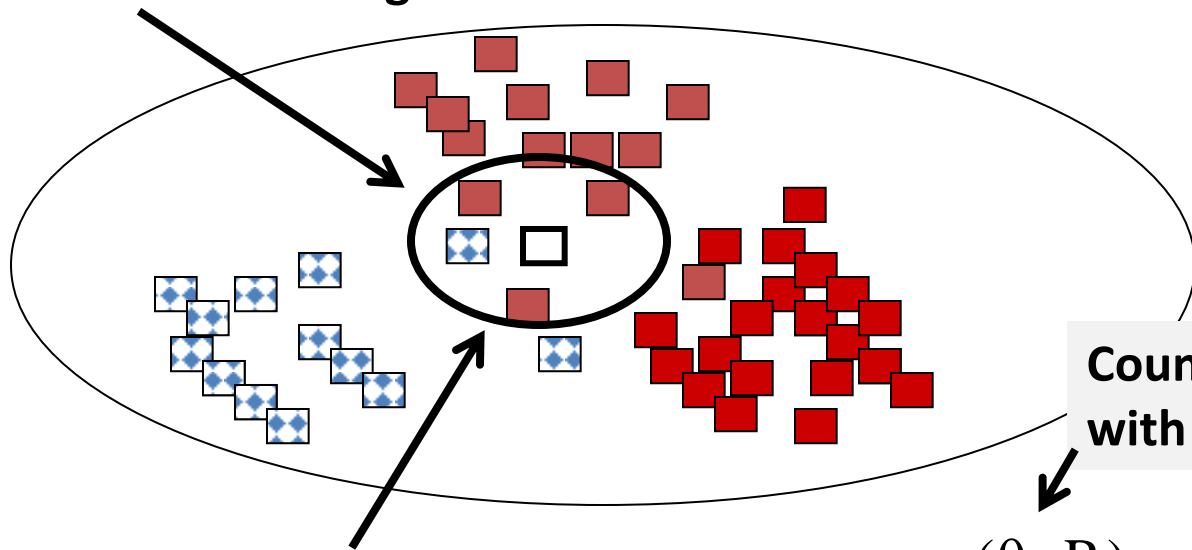


K-NN as an Estimate of $p(Y|X)$

Assume $p(\theta_i | d)$ is locally smooth, i.e.,
the same for all the d 's in this region R



$$p(\theta_i | d) = p(\theta_i | R)$$



Estimate $p(\theta_i | R)$ based on
the known categories in the region

$$p(\theta_i | R) = \frac{c(\theta_i, R)}{|R|}$$

Total # of
docs in R