Vocabulary-Based Document Classification

Brian Denton

December 1, 2009

Primary Objectives

- Classify documents into one of two subject categories based on the words used in the document
- Find estimate of prediction error
- Find words with highest predictive ability

Supervised Learning

- Because it is known *a priori* there are two subject categories into which all documents must be classified this is an example of supervised learning.

Description of Data

- The data come from 4,000 documents selected from the Reuters news network, each discussing one of two topics. The topics are known to us only as “pos” and “neg”.
- The data are presented in three text files: pos.txt, neg.txt, and voc.txt
- voc.txt is a list of all words contained in the 4,000 documents

⋮
absentia
absolut
absolv
absorb
absorpt
abstain
abstent
abstract
absurd
⋮

- pos.txt and neg.txt each have 2,000 lines (one line per document). Each line is a sequence of integer pairs. The first integer in the pair is the index of a word from voc.txt the second integer is the number of times that word appears.

  - For example, if the first line in pos.txt is:

    551 4 1322 1 2240 1 3285 4 5624 1 6266 1

    then the word at index 551 in voc.txt appears 4 times, the word at index 1322 appears 1 time, and so on. All words from voc.txt not listed appear zero times.

  - The data matrix is sparse. A linked list implementation vastly improves efficiency.

Logistic Regression Model

$$p_\theta\left(y^{(i)} \mid \mathbf{x}^{(i)}\right) = \frac{1}{1 + \exp\left[-y^{(i)} \sum_{j=1}^{m} \theta_j f\left(x_j^{(i)}\right)\right]}$$

$$y^{(i)} = \begin{cases} 1 & \text{if document } i \text{ is from pos.txt} \\ -1 & \text{if document } i \text{ is from neg.txt} \end{cases}, \quad i = 1, \ldots, n$$

$\mathbf{x}^{(i)}$ is a $1 \times m$ row vector with one column for each word in voc.txt

$f\left(x_j^{(i)}\right)$ is the relative frequency of word j in document i, $j = 1, \ldots, m$

$\boldsymbol{\theta}$ is an $m \times 1$ column vector of regression parameters where a particular vector element θ_j is the regression parameter for word j

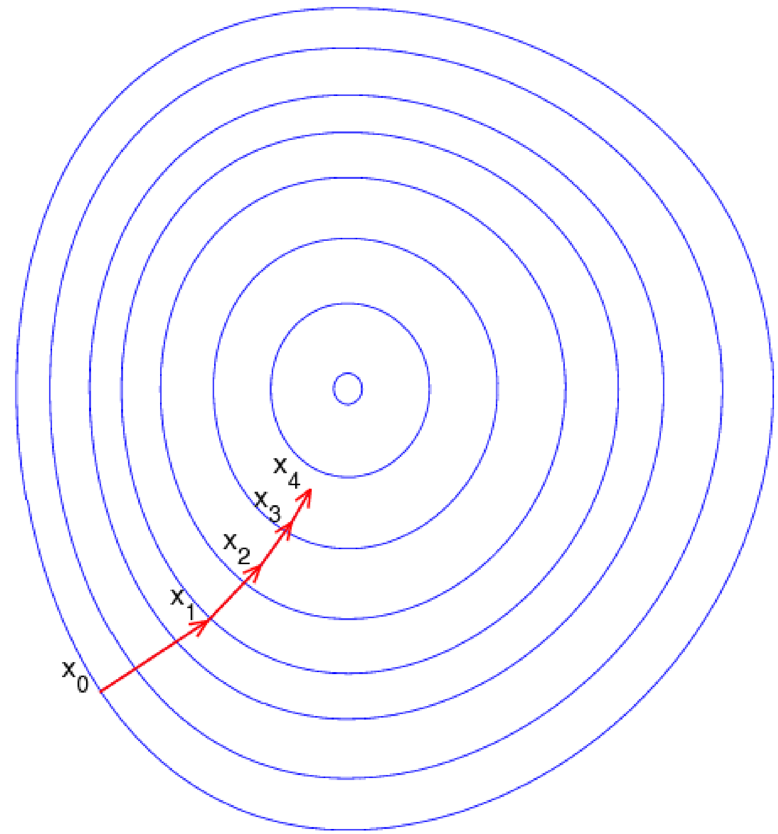
There are 4,000 documents in the dataset so $n = 4,000$

There are 34,803 words in voc.txt so $m = 34,803$

Gradient Ascent

- Gradient ascent is an iterative search method used here to find the parameter vector $\boldsymbol{\theta}$ that maximizes the log likelihood function $\ell(\boldsymbol{\theta})$.

- In the figure, the blue curves represent the level curves of $\ell(\boldsymbol{\theta})$ and the points $\mathbf{x}_t$ represent each iteration of $\boldsymbol{\theta}^{(t)}$



Log-Likelihood and the Gradient Vector

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\left(p_\theta\left(y^{(i)} \mid \mathbf{x}^{(i)}\right)\right)$$

$$= -\sum_{i=1}^{n} \log\left(1 + \exp\left[-y^{(i)} \sum_{j=1}^{m} \theta_j f\left(x_j^{(i)}\right)\right]\right)$$

The gradient vector of the log-likelihood is $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta})$ with k^{th} element:

$$\nabla_{\theta_k} \ell(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_k} \ell(\boldsymbol{\theta})$$

$$= \sum_{i=1}^{n} y^{(i)} f\left(x_k^{(i)}\right) \frac{A_i}{1 + A_i}, \quad \text{where } A_i = \exp\left[-y^{(i)} \sum_{j=1}^{m} \theta_j f\left(x_j^{(i)}\right)\right]$$

$$\equiv \mathbf{d}_k$$

Solve for $\boldsymbol{\theta}$ using Gradient Ascent

1. Initialize parameter vector to $\boldsymbol{\theta}^{(0)} = \mathbf{0}$
   Specify log-likelihood convergence stopping condition ε
   Specify initial stepsize α_0 and stepsize stopping condition α^*
2. Calculate log-likelihood $\ell\left(\boldsymbol{\theta}^{(t)}\right)$
3. Calculate gradient vector $\mathbf{d}^{(t)} = \nabla\ell\left(\boldsymbol{\theta}^{(t)}\right)$
4. Do line search to update $\boldsymbol{\theta}$
   a. Set $\alpha = \alpha_0$
   b. Calculate $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \alpha\mathbf{d}^{(t)}$
   c. Calculate $\ell\left(\boldsymbol{\theta}^{(t+1)}\right)$
   d. IF $\left\{\ell\left(\theta_i^{(t+1)}\right) - \ell\left(\theta_i^{(t)}\right)\right\} \geq 0,\ \forall i$ OR $\alpha < \alpha^*$ THEN terminate line search and go to step 5.
   e. ELSE set $\alpha = \alpha / 2$ and go to 4(b)
5. IF $\left|\ell\left(\boldsymbol{\theta}^{(t+1)}\right) - \ell\left(\boldsymbol{\theta}^{(t)}\right)\right| < \varepsilon$ THEN terminate maximum likelihood algorithm and return $\boldsymbol{\theta}^{(t+1)}$ as MLE for $\boldsymbol{\theta}$.
   a. ELSE go to Step 3 and take $\ell\left(\boldsymbol{\theta}^{(t+1)}\right)$ as new $\ell\left(\boldsymbol{\theta}^{(t)}\right)$

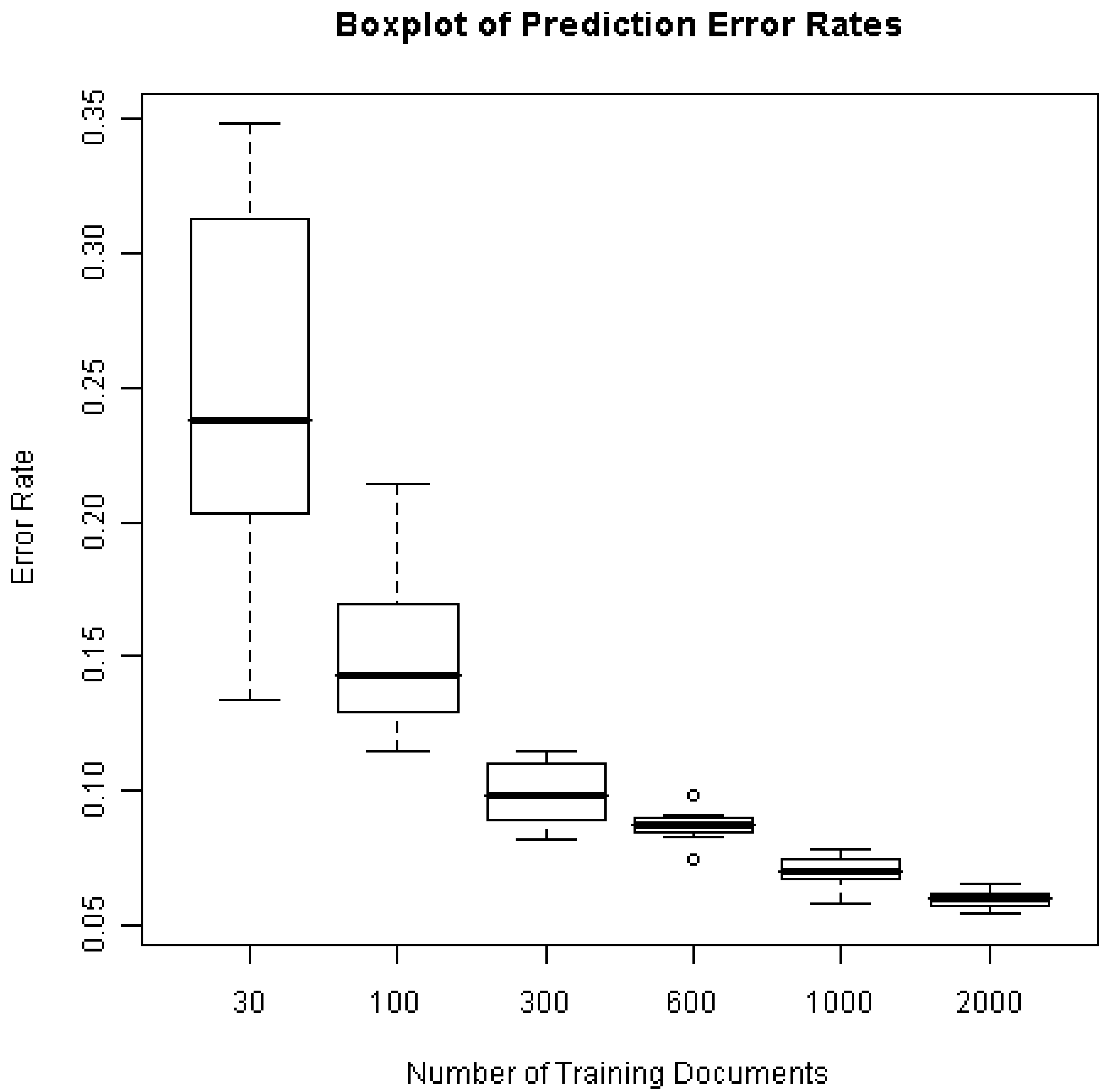
Generate Model Predictions

- Use the obtained parameter vector $\boldsymbol{\theta}$ to estimate the probabilities $p_\theta\left(y^{(i)}=1 \mid \mathbf{x}^{(i)}\right)$
- Assign documents to predicted category
- Assess accuracy of predictions

◦ Results

Elapsed Time (seconds)							
Number of Training Documents	Number of Test Documents	Number of Simulations	Mean	Std Dev	Minimum	Median	Maximum
30	2000	10	2.72	0.62	1.85	2.66	4.13
100	2000	10	12.61	1.36	10.39	12.65	14.37
300	2000	10	56.27	4.96	45.73	57.29	61.80
600	2000	10	145.68	8.92	134.31	142.43	160.24
1000	2000	10	310.52	32.72	273.76	302.77	383.23
2000	2000	10	774.97	55.02	685.13	777.52	852.40

Error Rate							
Number of Training Documents	Number of Test Documents	Number of Simulations	Mean	Std Dev	Minimum	Median	Maximum
30	2000	10	0.2483	0.0685	0.1340	0.2378	0.3480
100	2000	10	0.1518	0.0303	0.1150	0.1430	0.2145
300	2000	10	0.0995	0.0111	0.0820	0.0985	0.1145
600	2000	10	0.0872	0.0062	0.0745	0.0878	0.0985
1000	2000	10	0.0703	0.0060	0.0585	0.0705	0.0785
2000	2000	10	0.0594	0.0036	0.0545	0.0598	0.0655


Boxplot of Prediction Error Rates
Error Rate
0.05
0.10
0.15
0.20
0.25
0.30
0.35
30
100
300
600
1000
2000
Number of Training Documents

10 Smallest θ Values and the Corresponding Word from voc.txt

θ	Word
-91.935016	rate
-85.344031	export
-85.249718	year
-68.279730	import
-61.775788	quarter
-61.422642	figur
-59.743160	open
-59.394257	rostelekom
-58.319778	cost
-58.318453	market

10 Largest θ Values and the Corresponding Word from voc.txt

θ	Word
78.148321	debt
80.106942	deal
83.724733	hold
85.244287	acquir
85.850047	takeov
119.527991	acquisit
129.839461	share
136.824028	stake
142.434176	privatis
168.210442	merger