CPSC 340: Machine Learning and Data Mining

Probabilistic Classification

Bonus slides

Feature Representation for Spam

- Are there better features than bag of words?
  - We add bigrams (sets of two words):
    - “CPSC 340”, “wait list”, “special deal”.
  - Or trigrams (sets of three words):
    - “Limited time offer”, “course registration deadline”, “you’re a winner”.
  - We might include the sender domain:
    - <sender domain == “mail.com”>.
  - We might include regular expressions:
    - <your first and last name>.

"Proportional to" for Probabilities

- When we say "p(y) ∝ exp(-y²)" for a function 'p', we mean:

  $p(y) = \beta \exp(-y^2)$ for some constant 'β'.

- However, if 'p' is a probability then it must sum to 1.
  - If $y \in \{1,2,3,4\}$ then $p(1)+p(2)+p(3)+p(4)=1$

- Using this fact, we can find β:

$$\beta\exp(-1^2)+\beta\exp(-2^2)+\beta\exp(-3^2)+\beta\exp(-4^2)=1$$

$$\Leftrightarrow \beta[\exp(-1^2)+\exp(-2^2)+\exp(-3^2)+\exp(-4^2)=1$$

$$\Leftrightarrow \beta = \frac{1}{\exp(-1^2)+\exp(-2^2)+\exp(-3^2)+\exp(-4^2)}$$

Probability of Paying Back a Loan and Ethics

- Article discussing predicting “whether someone will pay back a loan”:
  - https://www.thecut.com/2017/05/what-the-words-you-use-in-a-loan-application-reveal.html
- Words that increase probability of paying back the most:
  - *debt-free, lower interest rate, after-tax, minimum payment, graduate.*
- Words that decrease probability of paying back the most:
  - *God, promise, will pay, thank you, hospital.*
- Article also discusses an important issue: are all these features ethical?
  - Should you deny a loan because of religion or a family member in the hospital?
  - ICBC is limited in the features it is allowed to use for prediction.

Avoiding Underflow

- During the prediction, the probability can underflow:

$$p(y_i = c \mid x_i) \propto \prod_{j=1}^{d} \underbrace{[p(x_{ij} \mid y_i = c)]}\, \underbrace{p(y_i = c)}$$

All these are < 1 so the product gets very small.

- Standard fix is to (equivalently) maximize the logarithm of the probability:

Rember that $\log(ab) = \log(a) + \log(b)$ so $\log(\prod a_i) = \sum_i \log(a_i)$

Since log is monotonic the 'c' maximizing $p(y_i = c \mid x_i)$ also maximizes $\log p(y_i = c \mid x_i)$,

so maximize $\log\left(\prod_{j=1}^{d} [p(x_{ij} \mid y_i = c)]\, p(y_i = c)\right) = \sum_{j=1}^{d} \log(p(x_{ij} \mid y_i = c)) + \log(p(y_i = c))$

Less-Naïve Bayes

- Given features {x1,x2,x3,…,xd}, naïve Bayes approximates p(y|x) as:

$$p(y \mid x_1, x_2, \dots, x_d) \propto p(y)\,p(x_1, x_2, \dots, x_d \mid y)$$

product rule applied repeatedly

$$= p(y)\,p(x_1 \mid y)\,p(x_2 \mid x_1, y)\,p(x_3 \mid x_2, x_1, y) \cdots p(x_d \mid x_1, x_2, \dots, x_{d-1}, y)$$

$$\approx p(y)\,p(x_1 \mid y)\,p(x_2 \mid y)\,p(x_3 \mid y) \cdots p(x_d \mid y) \quad \text{(naive Bayes assumption)}$$

- The assumption is very strong, and there are "less naïve" versions:
  - Assume independence of all variables except up to 'k' largest 'j' where j < i.
    - E.g., naïve Bayes has k=0 and with k=2 we would have:

$$\approx p(y)\,p(x_1 \mid y)\,p(x_2 \mid x_1, y)\,p(x_3 \mid x_2, x_1, y)\,p(x_4 \mid x_3, x_2, y) \cdots p(x_d \mid x_{d-2}, x_{d-1}, y)$$

    - Fewer independence assumptions so more flexible, but hard to estimate for large 'k'.
  - Another practical variation is "tree-augmented" naïve Bayes.

Computing p(x_i) under naïve Bayes

- Generative models don't need p(x_i) to make decisions.
- However, it's easy to calculate under the naïve Bayes assumption:

$$p(x_i) = \sum_{c=1}^{K} p(x_i, y=c) \qquad \text{(marginalization rule)}$$

$$= \sum_{c=1}^{K} p(x_i \mid y=c)\, p(y=c) \qquad \text{(product rule)}$$

$$= \sum_{c=1}^{K} \underbrace{\left[\prod_{j=1}^{d} p(x_{ij} \mid y=c)\right] p(y=c)}_{\text{These are the quantities we compute during training}} \qquad \text{(naïve Bayes assumption)}$$

Gaussian Discriminant Analysis

- Classifiers based on Bayes rule are called generative classifier:
  - They often work well when you have tons of features.
  - But they need to know $p(x_i \mid y_i)$, probability of features given the class.
    - How to "generate" features, based on the class label.

- To fit generative models, usually make BIG assumptions:
  - Naïve Bayes (NB) for discrete x_i:
    - Assume that each variables in x_i is independent of the others in x_i given y_i.

  - Gaussian discriminant analysis (GDA) for continuous x_i.
    - Assume that $p(x_i \mid y_i)$ follows a multivariate normal distribution.
    - If all classes have same covariance, it's called "linear discriminant analysis".

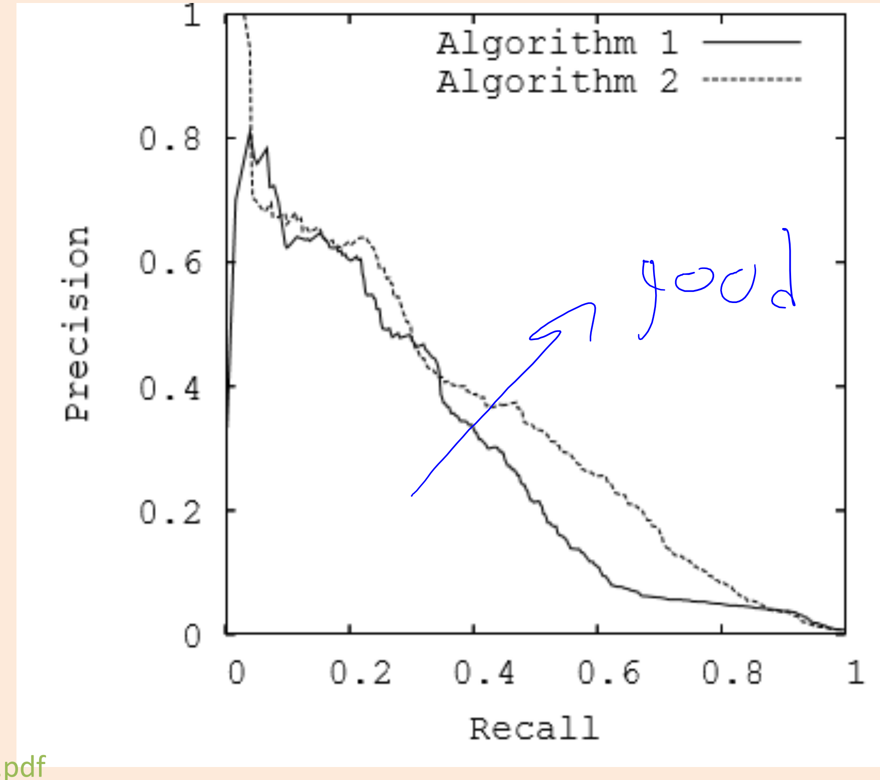
Other Performance Measures

- Classification error might be wrong measure:
  - Use weighted classification error if have different costs.
  - Might want to use things like Jaccard measure: TP/(TP + FP + FN).

- Often, we report precision and recall (want both to be high):
  - Precision: "if I classify as spam, what is the probability it actually is spam?"
    - Precision = TP/(TP + FP).
    - High precision means the filtered messages are likely to really be spam.
  - Recall: "if a message is spam, what is probability it is classified as spam?"
    - Recall = TP/(TP + FN)
    - High recall means that most spam messages are filtered.

Precision-Recall Curve

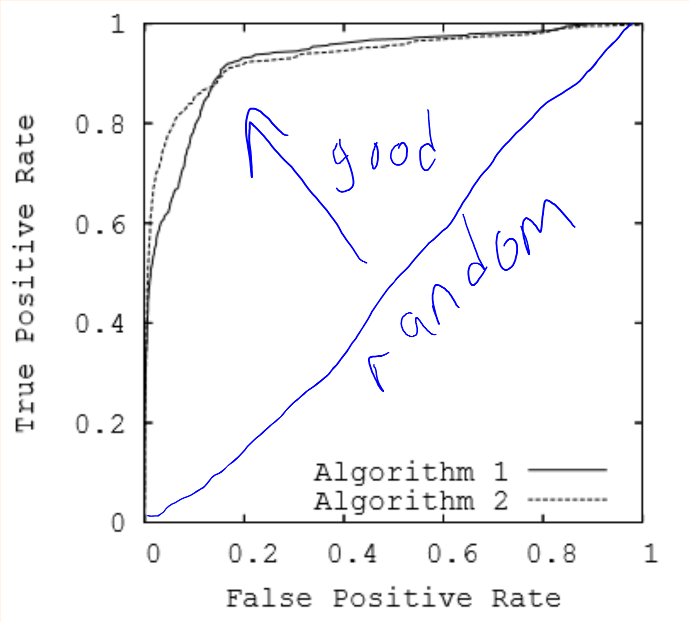
- Consider the rule $p(y_i = \text{'spam'} \mid x_i) > t$, for threshold ‘t’.
- Precision-recall (PR) curve plots precision vs. recall as ‘t’ varies.



http://pages.cs.wisc.edu/~jdavis/davisgoadrichcamera2.pdf

ROC Curve

- Receiver operating characteristic (ROC) curve:
  – Plot true positive rate (recall) vs. false positive rate (FP/FP+TN).
    (negative examples classified as positive)



  – Diagonal is random, perfect classifier would be in upper left.
  – Sometimes papers report area under curve (AUC).
    - Reflects performance for different possible thresholds on the probability.

http://pages.cs.wisc.edu/~jdavis/davisgoadrichcamera2.pdf

More on Unbalanced Classes

- With unbalanced classes, there are many alternatives to accuracy as a measure of performance:
  - Two common ones are the Jaccard coefficient and the F-score.

- Some machine learning models don't work well with unbalanced data. Some common heuristics to improve performance are:
  - Under-sample the majority class (only take 5% of the spam messages).
    - https://www.jair.org/media/953/live-953-2037-jair.pdf
  - Re-weight the examples in the accuracy measure (multiply training error of getting non-spam messages wrong by 10).
  - Some notes on this issue are here.