Content-Based Recommendation of RSS Feed Items

Michael Groble
Task Description

• Corpus
  – Approximately 17,000 items from my Google Reader history
  – Each labeled +1 if annotated as “starred”, -1 otherwise

• Attempt to learn classifier to predict next week’s starred items based on previous 16 weeks history
Performance Measure

• From corpus, only ~4% of items are marked starred

• Classification accuracy not the best metric
  – all -1 classification policy gives 96% accuracy

• Use $F_{10}$ instead
  – Precision = correct hypothesized stars / all hypothesized stars
  – Recall = correct hypothesized stars / all true stars
  – $F_{10} = 11 * \text{Precision} * \text{Recall} / (10 * \text{Precision} + \text{Recall})$
Investigations

• Item Selection
  – Some items published to multiple feeds and get different labels (typically only label the first-read as starred)
  – “Unlabel” some non-starred items which are too similar to starred ones
    • Compare using all training data vs. filtering out unlabeled items vs. semi-supervised learning with unlabeled items

• Feature Selection
  – Compare unigram only vs. unigram + bigram features
  – Compare information gain vs. mutual information ranking
  – Consider latent topic probabilities

• Kernel Selection
  – Linear vs. Bhattacharyya kernels

• Algorithm Selection
  – Multinomial Naïve Bayes
  – Support Vector Machine
Example Latent Topics

• Most prevalent feed topic clusters (over all labels, April-August)

- google
  - search
- yahoo
  - results
- ebay
- microsoft
- engine
- internet
- news
- engines
- technorati
- advertising
- data
- online
- ask
- iphone
  - apple
  - t
  - jobs
  - store
  - steve
  - line
  - stores
  - ipod
  - phone
  - new
  - sold
  - iphones
  - launch
  - today
- video
  - content
  - tv
  - youtube
  - videos
  - live
  - player
  - media
  - television
  - channel
  - joost
  - watch
  - quality
  - movie
- microsoft
  - open
  - source
  - software
  - ibm
  - community
  - linux
  - deal
  - patent
  - red
  - patents
  - license
  - open-source
  - hat
  - novell
- facebook
  - platform
  - users
  - myspace
  - applications
  - developers
  - application
  - friends
  - slide
  - user
  - social
  - growth
  - apps
  - rockyou
  - app
- company
  - million
  - quarter
  - cents
  - says
  - today
  - stock
  - year
  - billion
  - shares
  - share
  - price
  - revenue
  - market
  - sales
- mobile
  - phone
  - service
  - phones
  - wireless
  - internet
  - services
  - calls
  - customers
  - new
  - verizon
  - network
  - announced
  - devices
  - voice
Results

• Best overall performance achieved through SVM with Bhattacharyya kernel on bigram features ($F_{10} = 45.52\%$)
  – Filtered out 75% of items while retaining 70% of starred items
  – Both Unigram and Bigram Naïve Bayes with filtered similar items and top 5,000 features ranked by information gain produced very similar results to this best result

• Semi-supervised approaches did not improve $F_{10}$

• Latent topic features did improved accuracy and precision, but degraded recall and $F_{10}$