TURNING DOWN THE NOISE IN BLOGOSPHERE

Nisha Ranga
OBJECTIVE

• INTRODUCTION
• COVERING THE BLOGOSPHERE
  • Definition
  • Near-optimal solution
• PERSONALIZED COVERAGE
  • No-regret algorithm
• EVALUATE ALGORITHM
INTRODUCTION

TOP STORIES OF BLOGOSPHERE

'Lipstick on a pig': Attack on Palin or common line?
sept. 2008

Lady Gaga Meat Dress for Halloween? Butcher's Weigh In
Oct 2010
COVERING THE BLOGOSPHERE
CHARACTERIZE THE POST BY FEATURES

Low level feature

USA, Obama, Network, ipad
(noun, entities)

High level feature

Rise of China, Japan after earth quake
(Topic)
• **BLOGOSPHERE**: A blogosphere is a triplet \( \langle U, \text{Posts}, \text{cover} \cdot (\cdot) \rangle \)

\[ U = \{u_1, u_2, \ldots\} \text{ is a finite set of features } & \text{Posts is a finite set of posts} \]

Cover function = Amount by which ‘Document d’ covers ‘features f’

Set of posts – Cover function = Amount by which ‘Set A’ covers ‘feature f’

\[ F_A(f) = \sum \text{cover}_A(f) \]
**DRAWBACK**

- **Feature significance in corpus**: All features in a corpus are treated equally, and thus we cannot emphasize the importance of certain features *eg. covering “Cathedral High School” should not be as valuable as covering “Obama.”*

- **Feature significance in post**: This objective function does not characterize how relevant a post is to a particular feature *eg. a post about Obama’s speech covers Obama just as much as a post that barely mentions him*

- Incremental coverage: This coverage notion is too strong, since after seeing one post that covers a certain feature, we will never gain anything from another post that covers the same feature. *eg. If first post covered the Obama’s speech really well then the second post doesn’t has anything new to offer*
Some features are more important than others

Solution: Define Probabilistic Coverage

\[ \text{cover}_A (f) = P(\text{feature } f \mid \text{post } d) \]

With topic as feature; we say that post \(d\) is about topic \(f\)

i.e The first post is about Obama
Want to cover important features

Solution:
Associate a weight ‘w’ with each feature ‘f’
Where f is the frequency of feature in corpus cover an important feature by multiple posts

\[ F_A (f) = \sum w \text{cover}_A (f) \]
• Avoid same coverage by all the posts

Solution:
We define probability that at least one post in the set A covers the feature f
It define feature importance in individual posts and in whole corpus

\[ F_A (f) = \sum w_f \text{cover}_A (f) \]
Each user has different interests, and selected post may cover stories not of his interest

Utilize users feedback in order to learn a personalized notion of coverage of each user

$F(A)$ assigns fixed weight to each feature. It may vary among different users

User’s coverage function is of the form

$$F_A (f) = \sum \pi_i w_f cover_A (f)$$

Where $\pi_i$ is personalized preference
Feedback a user provides for a post is independent of other post presented in the same set

Easy to identify the ‘liking’ of the user

If a user did not like the post either because of its contents or because the previous post have already covered the story

- Incremental coverage of a post
Evaluation

- Evaluation is done on data collected over 2 week period
- Algorithm was run to select important and prevalent stories
- Measuring Topicality
- Measuring Redundancy
Yahoo! Performs well on both redundancy and topicality measurement; but it uses rich features like CTR, Search trends, user voting etc.

While TDN + LDA uses only text based features.
Google has good topicality
Algorithm TDN+LDA is as good as Google