SENDERS, RECEIVERS, AND AUTHORS IN DOCUMENT CLASSIFICATION

Anna Drummond
Chris Jermaine
Rice University

Document Classification: a Classical Problem

- But what do you do when you have people associated?
  - Author(s)
  - Sender
  - Receiver(s)
  - Those carbon copied on email

Document Classification: a Classical Problem

- But what do you do when you have people associated?
  - — Author(s)
  - — Sender
  - — Receiver(s)
  - — Those carbon copied on email
- In our problem domain, such people are key information
  - — Electronic discovery in courtroom litigation
  - — 70% of e-discovery is searching through emails
  - — Must find those relevant to some aspect of the case
  - — Too expensive to do first pass by hand means multi-label classification
  - — Clearly, sender/receiver information is important!

What’s the Obvious Way to Handle People?

- Just use the traditional bag-of-words...
  - — and append people on at the end
  - — then use a standard classifier
- Example: we have [Joe, Jen, John, Sue] in our database
  - — And bag-of-words encoding of a particular email is [0, 2, 4, 1, 0]
  - — Joe sent an email to Jen and Sue

What’s the Obvious Way to Handle People?

- Just use the traditional bag-of-words...
  - — and append people on at the end
  - — and use a standard classifier
- Example: we have [Joe, Jen, John, Sue] in our database
  - — And bag-of-words encoding of a particular email is [0, 2, 4, 1, 0]
  - — Joe sent the email to Jen and Sue
  - — So we encode the email as [0, 2, 4, 1, 0] with [1, 0, 0, 0] and [0, 1, 0, 1] appended
  - — Or, [0, 2, 4, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1]

Unfortunately, Not Particularly Useful

- 4,659 emails from a construction litigation
- Nine, non-exclusive possible labels
- Learned a model using a SVM... Here is the AUC:

	No people	With People
Label 1	.9147	.9092
Label 2	.9501	.9514
Label 3	.8824	.8850
Label 4	.7749	.7754
Label 5	.7971	.8015
Label 6	.7335	.7363
Label 7	.9211	.9193
Label 8	.7396	.7404
Label 9	.7241	.7314

avg: 0.8264 with
0.8278 w/o

What's the Problem?

- SVM actually does well on emails with few people
- But very badly on emails with many people
  - — SVM does not understand "receivers" or "senders" is really a single, set-valued att
- Weight of "receivers" vis-a-vis words-in-doc should not vary (much) with size
  - — Ex: I often send emails to Joe, Jen, John, and Sue about data mining...
  - — Is the recipient set {Joe, Jen, John, Sue} more indicative of DM than {Joe, Jen}?
  - — Probably not!

What's the Problem?

- SVM actually does well on emails with few people
- But very badly on emails with many people
  - — SVM does not understand "receivers" or "senders" is really a single, set-valued att
- Weight of "receivers" vis-a-vis words-in-doc should not vary (much) with size
  - — Ex: I often send emails to Joe, Jen, John, and Sue about data mining...
  - — Is the recipient set {Joe, Jen, John, Sue} more indicative of DM than {Joe, Jen}?
  - — Probably not!
- Can't we just normalize?
  - — [0, 2, 4, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1] becomes [0, 2, 4, 1, 0, 1, 0, 0, 0, 0, 0.5, 0, 0.5]
  - — Yes, but this normalization does not understand the relative importance of people

Our Solution

- Map each person to a point in a low-dimensional latent space
- For a given cat. (sender, receiver, etc.) each person is weighted
  - — Very important to a category relative to others? You have a high weight

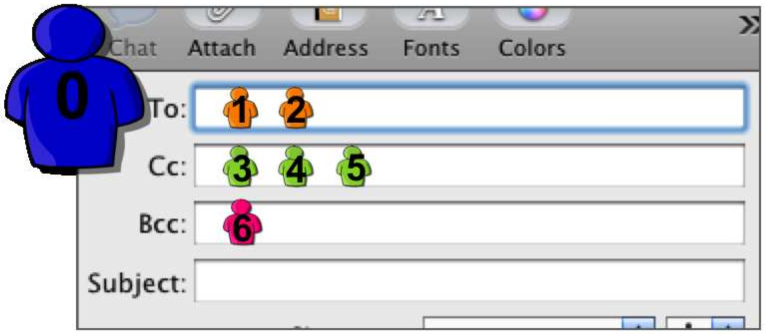
Our Solution

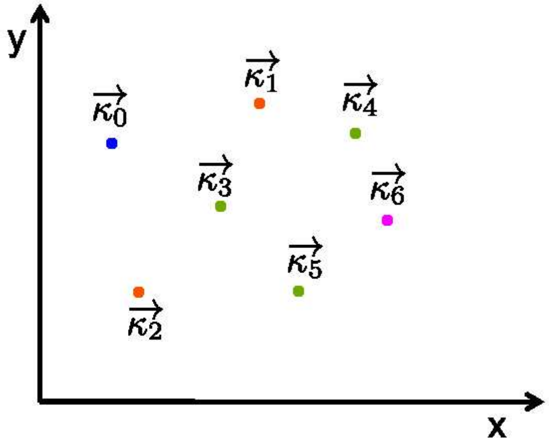
- Map each person to a point in a low-dimensional latent space
- For a given cat. (sender, receiver, etc.) each person is weighted
  - — Very important to a category relative to others? You have a high weight
- That category is then represented as a low-dim, weighted sum:

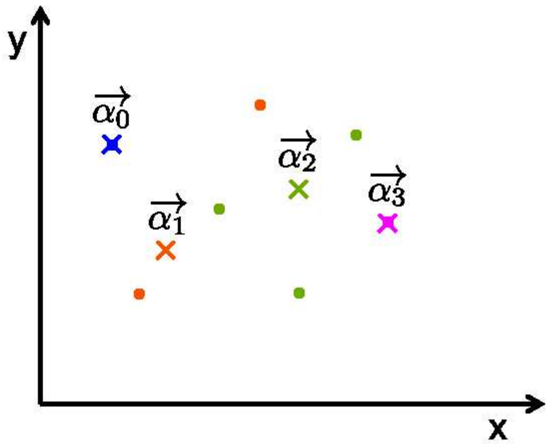
$$\overrightarrow{\alpha_{D_c}} = \sum_{p \in D_c} \frac{\overrightarrow{\kappa_p} \times \overrightarrow{w_{p,c}}}{\overrightarrow{w_{p,c}}}$$

  - — Here, D_c is the set of people associated with category c in document D
  - — w is the weight vector, and kappa is the latent position
- Then, append $\overrightarrow{\alpha_{D_c}}$ to the bag-of-words vector

Pictorially







In Our Paper…

- We suggest multiple ways in which this method can be used
- And evaluate the embedding-based-method extensively
- Ex. On the construction litigation problem, we have:

	No people	With People	Embedding
Label 1	.9147	.9092	.9159
Label 2	.9501	.9514	.9585
Label 3	.8824	.8850	.8842
Label 4	.7749	.7754	.7957
Label 5	.7971	.8015	.8408
Label 6	.7335	.7363	.8063
Label 7	.9211	.9193	.9419
Label 8	.7396	.7404	.8615
Label 9	.7241	.7314	.8155

Avg:
0.8264 vs.
0.8278 vs.
0.8689

Questions?