A log-linear model with latent features for dyadic prediction

Aditya Krishna Menon and Charles Elkan

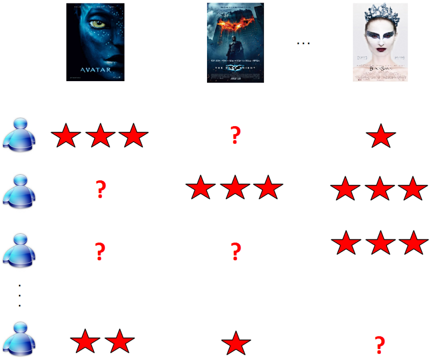
University of California, San Diego

December 17, 2010

Outline

The movie rating prediction problem

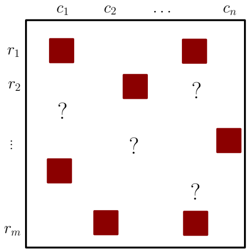
- Given users' ratings of movies they have seen, predict ratings on the movies they have not seen



- Popular solution strategy is collaborative filtering: leverage everyone's ratings to determine individual users' tastes

Generalizing the problem: dyadic prediction

- In dyadic prediction, our training set is $\{((r_i, c_i), y_i)\}_{i=1}^{n}$, where each pair (r_i, c_i) is called a dyad, and each y_i is a label
- **Goal**: Predict the label y' for a new dyad (r', c')
  - Matrix completion with r_i's as rows and c_i's as columns



- The choice of r_i, c_i and y_i yields different problems
  - In movie rating prediction, r_i = user ID, c_i = movie ID, and y_i is the user's rating of the movie

Different instantiations of dyadic prediction

- Dyadic prediction captures problems in a range of fields:
  - Collaborative filtering: will a user like a movie?
  - Link prediction: do two people know each other?
  - Item response theory: how will a person respond to a multiple choice question?
  - Political science: how will a senator vote on a bill?
  - ...
- Broadly, two major ways to instantiate different problems:
  - r_i, c_i could be unique identifiers, feature vectors, or both
  - y_i could be ordinal (e.g. 1–5 stars), or nominal (e.g. $\{$ friend, colleague, family $\}$)

Proposed desiderata of a dyadic prediction model

- Bolstered by the Netflix challenge, there has been significant effort on improving the accuracy of dyadic prediction models
- However, other factors have not received as much attention:
  - Predicting well-calibrated probabilities over the labels, e.g. $\Pr[\text{Rating} = 5 \text{ stars}|\text{user, movie}]$
    - Essential when we want to make decisions based on users' predicted preferences
  - Ability to handle nominal labels in addition to ordinal ones
    - e.g. user-user interactions of $\{$ friend, colleague, family $\}$, user-item interactions of $\{$ viewed, purchased, returned $\}$, ...
  - Allowing both unique identifiers and feature vectors
    - Helpful for accuracy and cold-start dyads respectively
    - Want them to complement each other's strengths

This work

- We are interested in designing a simple yet flexible dyadic prediction model meeting these desiderata
- To this end, we propose a log-linear model with latent features (LFL)
  - Mathematically simple to understand and train
  - Able to exploit the flexibility of the log-linear framework
- Experimental results show that our model meets the new desiderata without sacrificing accuracy

Outline

The log-linear framework

- Given inputs $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$, a log-linear model assumes the probability

$$p(y|x; w) = \frac{\exp(\sum_i w_i f_i(x, y))}{\sum_{y'} \exp(\sum_i w_i f_i(x, y'))}$$

  where w is a vector of weights, and each $f_i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a feature function
  - Freedom to pick f_i's means this is a very flexible class of model
  - Captures logistic regression, CRFs, ...
- A useful basis for a dyadic prediction model:
  - Directly models probabilities of labels given examples
  - Natural mechanism for combining identifiers and side-information descriptions of the inputs x
  - Labels y can be nominal

A simple log-linear model for dyadic prediction

- For a dyad x with members $(r(x), c(x))$ that are unique identifiers, we can construct sets of indicator feature functions:

$$f^1_{ry'}(x, y) = \mathbf{1}[r(x) = r, y = y']$$
$$f^2_{cy'}(x, y) = \mathbf{1}[c(x) = c, y = y']$$
$$f^3_{y'}(x, y) = \mathbf{1}[y = y']$$

  - For simplicity, we'll call each $r(x)$ a user, each $c(x)$ a movie, and each y a rating
- Using these feature functions yields the probability model

$$p(y|x; w) = \frac{\exp(\alpha^y_{r(x)} + \beta^y_{c(x)} + \gamma^y)}{\sum_{y'} \exp(\alpha^{y'}_{r(x)} + \beta^{y'}_{c(x)} + \gamma^{y'})}$$

where $w = \{\alpha^y_r\} \cup \{\beta^y_c\} \cup \{\gamma^y\}$ for simplicity

  - $\alpha^y_{r(x)}$ = affinity of user $r(x)$ for rating y, and so on

Incorporating side-information into the model

- If the dyad x has a vector $s(x)$ of side-information, we can simply augment our probability model to use this information:

$$p(y|x;w) = \frac{\exp(\alpha^{y}_{r(x)} + \beta^{y}_{c(x)} + \gamma^{y} + (\delta^{y})^{T} s(x))}{\sum_{y'} \exp(\alpha^{y'}_{r(x)} + \beta^{y'}_{c(x)} + \gamma^{y'} + (\delta^{y'})^{T} s(x))}$$

  - Additional weights $\{\delta^{y}\}$ used to exploit the extra information
  - Corresponds to adding more feature functions based on $s(x)$

Are we done?

- This log-linear model is conceptually and practically simple
  - Parameters can be learnt by optimizing conditional log-likelihood using stochastic gradient descent
- But some questions remain:
  - Is it rich enough to be a useful method?
  - Is it suitable for ordinal labels?
- In fact, the model is not sufficiently expressive: there is no interaction between users' and movies' weights
  - The ranking of all movies $c_1, \ldots, c_n$ according to the probability $p(y|x; w)$ is independent of the user!

Outline

Dyadic prediction: definition and goals

A simple log-linear model for dyadic prediction

Adding latent features to the log-linear model

Experimental results

Conclusion

Capturing interaction effects: the LFL model

- To explicitly model interactions between users and movies, we modify the probability distribution:

$$p(y|x;w) = \frac{\exp(\alpha^{y}_{r(x)} + \beta^{y}_{c(x)} + \gamma^{y})}{\sum_{y'} \exp(\alpha^{y'}_{r(x)} + \beta^{y'}_{c(x)} + \gamma^{y'})}$$

Capturing interaction effects: the LFL model

- To explicitly model interactions between users and movies, we modify the probability distribution:

$$p(y|x;w) = \frac{\exp(\alpha^{y}_{r(x)}\beta^{y}_{c(x)} + \gamma^{y})}{\sum_{y'} \exp(\alpha^{y'}_{r(x)}\beta^{y'}_{c(x)} + \gamma^{y'})}$$

Capturing interaction effects: the LFL model

- To explicitly model interactions between users and movies, we modify the probability distribution:

$$p(y|x;w) = \frac{\exp(\sum_{k=1}^{K} \alpha^y_{r(x)k}\beta^y_{c(x)k} + \gamma^y)}{\sum_{y'} \exp(\sum_{k=1}^{K} \alpha^{y'}_{r(x)k}\beta^{y'}_{c(x)k} + \gamma^{y'})}$$

- For each rating value y, we keep a matrix $\alpha^y \in \mathbb{R}^{|R|\times K}$ of weights, and similarly for movies
  - Thus user r has an associated vector $\alpha^y_r \in \mathbb{R}^K$, so that

$$p(y|x;w) \propto \exp((\alpha^y_{r(x)})^T \beta^y_{c(x)} + \gamma^y)$$

  - We think of $\alpha^y_{r(x)}, \beta^y_{c(x)}$ as latent feature vectors, and so we call the model latent feature log-linear or LFL

LFL and matrix factorization

- The LFL model is a matrix factorization, but in log-odds space: if $P^{yy'}_{rc} := \log \frac{p(y|(r,c);w)}{p(y'|(r,c);w)}$, then

$$P^{yy'} = (\alpha^y)^T \beta^y - (\alpha^{y'})^T \beta^{y'}$$

- Fixing some y_0 as the base class with $\alpha^{y_0} \equiv \beta^{y_0} \equiv 0$:

$$Q^y := P^{yy_0} = (\alpha^y)^T \beta^y$$

  - Therefore, we have a series of factorizations, one for each possible rating y
  - We will combine these factorizations in a slightly different way than in standard collaborative filtering

Using the model: prediction and training

- The model's prediction, and in turn the training objective, both depend on whether the labels y_i are nominal or ordinal
- In both cases, as with the simple model, we can use stochastic gradient descent for large-scale optimization
- We'll study both cases in turn under the following setup:

  Input. Matrix X with observed entries $\mathcal{O}$, with X_{rc} being the training set label for dyad (r, c)

  Output. Prediction matrix $\hat{X}$ with unobserved entries filled in

Prediction and training: nominal labels

- For nominal labels, we predict the mode of the distribution:

$$\hat{X}_{rc} = \mathrm{argmax}_y \, p(y|(r,c);w)$$

- We use conditional log-likelihood as the objective, which does not impose any structure on the labels:

$$\mathrm{Obj}_{\mathrm{nom}} = \sum_{(r,c)\in\mathcal{O}} -\log p(X_{rc}|(r,c);w) + \sum_y \frac{\lambda_\alpha}{2}||\alpha^y||_F^2 + \frac{\lambda_\beta}{2}||\beta^y||_F^2$$

  - We use ℓ_2 regularization of parameters to prevent overfitting

Prediction and training: ordinal labels

- For ordinal labels, the previous objective does not consider that e.g. for a true label of 4 stars, predicting 1 star is worse than predicting 5 stars; all errors are considered equal
- Instead of using the mode, it is beneficial to predict the expected rating under the probability distribution:

$$\hat{X}_{rc} = \mathbb{E}_y[p(y|(r,c);w)] = \sum_y y p(y|(r,c);w)$$

- The objective we use depends on the performance measure on test data; typically, we use mean square error:

$$\text{Obj}_{\text{ord}} = \sum_{(r,c)\in\mathcal{O}} \left(X_{rc} - \hat{X}_{rc}\right)^2 + \sum_y \frac{\lambda_\alpha}{2}||\alpha^y||_F^2 + \frac{\lambda_\beta}{2}||\beta^y||_F^2$$

Reducing number of parameters in ordinal setting

- The model has one set of user/movie weights for each rating
  - Plausible that characteristics that make a movie likely to be 1 star are different to those that make it 5 stars
  - But intuitively, the parameters share a lot of structure
- We can cut down the number of parameters by assuming a decomposition of the model predictions:

$$(\alpha^y)^T \beta^y = \sum_{\ell=1}^{L} \phi_{\ell y} (\tilde{\alpha}^\ell)^T \tilde{\beta}^\ell$$

  - Each rating y imposes a series of scaling factors $\phi_{\ell y}$ on each latent vector
  - If $L \ll |\mathcal{Y}|$, we reduce the $\#$ of parameters being estimated
- Similar to the stereotype model for ordinal logistic regression

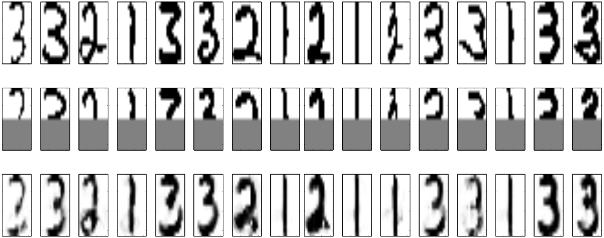
Outline

Experimental setup

- We present results on a range of dyadic prediction tasks, aiming to demonstrate:
  - Model richness via a general matrix completion problem
  - Handling nominal labels via a link prediction dataset
  - Incorporation of side-information in a cold-start setting
  - Respecting ordinal constraints via a collaborative filtering problem
- The aim of these experiments is to show the flexibility of the LFL model, and that it meets the desiderata we listed earlier
  - Not focussed on improving accuracy for collaborative filtering tasks, though that is an important problem

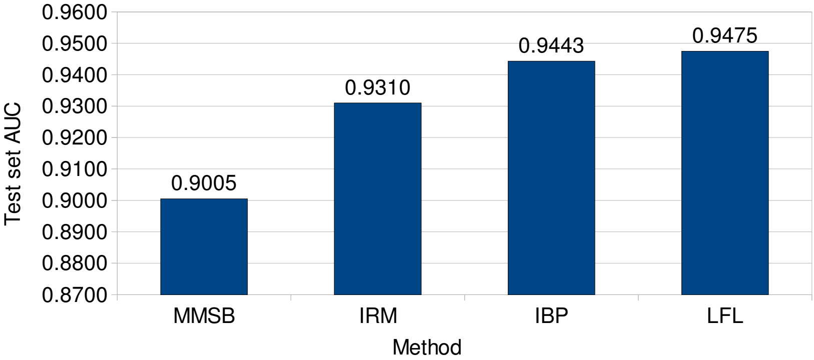
General matrix completion task

- Taking digits $\{1, 2, 3\}$ from the USPS dataset, we construct a dyadic dataset of image IDs by pixel positions
- If we occlude the bottom half of some images, can we reconstruct them given the rest of the data?
- Results of our model:



Experiments on nominal link prediction

- We took the alyawarra dataset, comprising relationships between 104 people
  - Each relationship is one of several kinship relations i.e. $\{$ Brother, Sister, Father, $\ldots\}$
- The multinomial LFL model achieves better AUC than previously proposed Bayesian methods:

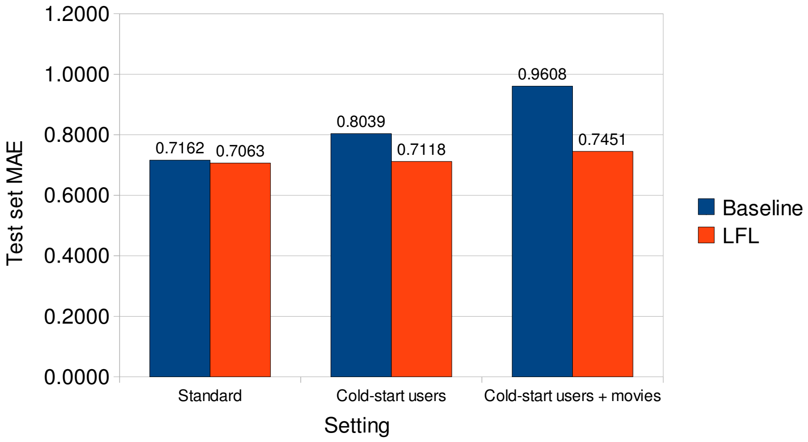


Experiments with side-information - I

- We check the usefulness of side-information in overcoming the cold-start problem
- We took the 100K movielens dataset and randomly discarded 50 users from the training set to act as the cold-start users
- We consider three scenarios:
  - The standard setting with no cold-start users/movies
  - The setting where there are 50 cold-start users
  - The setting where there are 50 cold-start users, and their test set movies are made cold-start also
- Baseline method is to just predict the average rating over the training set
- Side-information is user's age and gender, and movie's genre

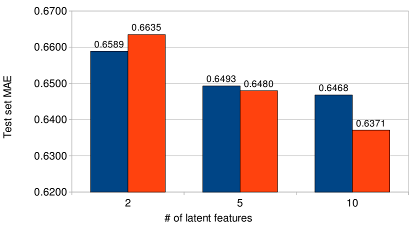
Experiments with side-information - II

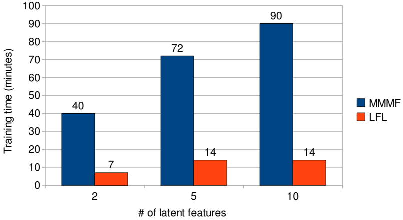
- Our model successfully exploits side-information to address the cold-start setting:



Experiments on collaborative filtering

- We ran experiments on the 1M movielens dataset, consisting of 6040 users and 3952 movies
  - For each user, a random rating is placed in the test set, and the rest are used for training
- Despite being more general, the LFL model is competitive with, yet faster than, the MMMF method:





Outline

Dyadic prediction: definition and goals

A simple log-linear model for dyadic prediction

Adding latent features to the log-linear model

Experimental results

Conclusion

Conclusion

- We presented a log-linear model with latent features for dyadic prediction
- The aim of the model is to address a range of desiderata, including:
  - predicting well-calibrated probabilities
  - handling nominal and ordinal labels, and
  - exploiting both side-information and unique identifiers
- The model is mathematically simple and easy to train
- Experimental results demonstrate its flexibility and good performance