Collaborative Filtering

Nicholas Ruozzi
University of Texas at Dallas

based on the slides of Alex Smola & Narges Razavian

Collaborative Filtering

- Combining information among collaborating entities to make recommendations and predictions
  - Can be viewed as a supervised learning problem (with some caveats)
  - Because of its many, many applications, it gets a special name

Examples

- Movie/TV recommendation (Netflix, Hulu, iTunes)
- Product recommendation (Amazon)
- Social recommendation (Facebook)
- News content recommendation (Yahoo)
- Priority inbox & spam filtering (Google)
- Online dating (OK Cupid)

Netflix Movie Recommendation

Training Data

user	movie	rating
1	14	3
1	200	4
1	315	1
2	15	5
2	136	1
3	235	3
4	79	3

Test Data

user	movie	rating
1	50	?
1	28	?
2	94	?
2	32	?
3	11	?
4	99	?
4	54	?

Recommender Systems

- Content-based recommendations
  - Recommendations based on a user profile (specific interests) or previously consumed content
- Collaborative filtering
  - Recommendations based on the content preferences of similar users
- Hybrid approaches

Collaborative Filtering

- Widely-used recommendation approaches:
  - k-nearest neighbor methods
  - Matrix factorization based methods
- Predict the utility of items for a user based on the items previously rated by other like-minded users

Matrix Factorization

- There could be a number of latent factors that affect the recommendation
  - Style of movie: serious vs. funny vs. escapist
  - Demographic: is it preferred more by men or women
- View CF as a matrix factorization problem

Matrix Factorization

- Express a matrix $M \in \mathbb{R}^{m \times n}$ approximately as a product of factors $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{p \times n}$

$$M \sim A \cdot B$$

- Approximate the user $\times$ items matrix as a product of matrices in this way
  - Similar to SVD decompositions that we saw earlier (SVD can't be used for a matrix with missing entries)
  - Think of the entries of M as corresponding to an inner product of latent factors

Matrix Factorization

users

items

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

~

~

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

•

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

[from slides of Alex Smola]

Matrix Factorization

users

items

1		3			5			5		4	
		5		?		4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

~

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

•

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

[from slides of Alex Smola]

Matrix Factorization

users

items

1		3			5			5		4	
		5		?		4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

~

items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

●

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

[from slides of Alex Smola]

Matrix Factorization

- We can express finding the “closest” matrix as an optimization problem

$$\min_{A,B} \sum_{(u,i)\ observed} \left(M_{u,i} - \langle A_{u,:}, B_{:,i} \rangle\right)^2 + \lambda(\|A\|_F^2 + \|B\|_F^2)$$

Matrix Factorization

- We can express finding the "closest" matrix as an optimization problem

$$\min_{A,B} \sum_{(u,i)\ observed} \left(M_{u,i} - \langle A_{u,:}, B_{:,i} \rangle\right)^2 + \lambda(\|A\|_F^2 + \|B\|_F^2)$$

Computes the error
in the approximation
of the observed
matrix entries

Matrix Factorization

- We can express finding the "closest" matrix as an optimization problem

$$\min_{A,B} \sum_{(u,i)\ observed} \left(M_{u,i} - \langle A_{u,:}, B_{:,i} \rangle\right)^2 + \lambda\left(\|A\|_F^2 + \|B\|_F^2\right)$$

Regularization preferences matrices with small Frobenius norm

Matrix Factorization

- We can express finding the "closest" matrix as an optimization problem

$$\min_{A,B} \sum_{(u,i)\ observed} \left(M_{u,i} - \langle A_{u,:}, B_{:,i} \rangle\right)^2 + \lambda(\|A\|_F^2 + \|B\|_F^2)$$

- How to optimize this objective?

Matrix Factorization

- We can express finding the "closest" matrix as an optimization problem

$$\min_{A,B} \sum_{(u,i)\ observed} \left(M_{u,i} - \langle A_{u,:}, B_{:,i} \rangle\right)^2 + \lambda(\|A\|_F^2 + \|B\|_F^2)$$

- How to optimize this objective?

    - (Stochastic) gradient descent!

Extensions

- The basic matrix factorization approach doesn't take into account the observation that some people are tougher reviewers than others and that some movies are over-hyped
  - Can correct for this by introducing a bias term for each user and a global bias

$$\min_{A,B,\mu,b} \sum_{(u,i)\ observed} \left(M_{u,i} - \mu - b_i - b_u - \langle A_{u,:}, B_{:,i} \rangle\right)^2$$

$$+\lambda(\|A\|_F^2 + \|B\|_F^2) + \nu \left(\sum_i b_i^2 + \sum_u b_u^2 \right)$$

End of course content…

Supervised Learning

- Regression & classification
- Discriminative methods
  - k-NN
  - Decision trees
  - Perceptron
  - SVMs & kernel methods
  - Logistic regression
  - Neural networks
- Parameter learning
  - Maximum likelihood estimation
  - Expectation maximization

Bayesian Approaches

- Maximum likelihood estimation
- MAP estimation
- Prior/posterior probabilities
- Naive Bayes

Unsupervised Learning

- Clustering
  - k-means
  - Spectral clustering
  - Hierarchical clustering
- Expectation maximization
  - Soft clustering
  - Mixtures of Gaussians

Learning Theory

- PAC learning
- VC dimension
- Bias/variance tradeoff
- Chernoff bounds
- Sample complexity

Optimization Methods

- Gradient descent
  - Stochastic gradient descent
  - Subgradient methods
- Coordinate descent
- Lagrange multipliers and duality

Matrix Based Methods

- Dimensionality Reduction
  - PCA
  - Matrix Factorizations

Ensemble Methods

- Bootstrap sampling
- Bagging
- Boosting

Other Learning Topics

- Reinforcement learning

Questions about the course content?

For the final...

- You should understand the basic concepts and theory of all of the algorithms and techniques that we have discussed in the course
- There is no need to memorize complicated formulas, etc.
  - For example, if I ask for the sample complexity of a scheme, I will give you the generic formula
- However, you should be able to derive the algorithms and updates
  - E.g., Lagrange multipliers and SVMs, the EM algorithm, etc.

For the final...

- No calculators, books, notes, etc. will be permitted
  - As before, if you need a calculator, you have done something terribly wrong
- The exam will be in roughly the same format
  - Expect true/false questions, short answers, and two-three long answer questions
- Exam will emphasize the new material, but ALL material will be tested
- Take a look at the practice exams!

Final Exam

Monday, 12/11/2023

11:00AM - 1:45PM

GR 2.530

ML Related Researchers at UTD

- Vincent Ng (NLP)
- Vibhav Gogate (MLNs, Sampling, Graphical Models)
- Sriraam Natarajan (Graphical Models & Reinforcement Learning)
- Sanda Harabagiu (NLP & Health)
- Nicholas Ruozzi (Graphical Models & Approx. Inference)
- Rishabh Iyer (Submodular Optimization)

…

ML Related Researchers at UTD

- Yu Xiang (Computer Vision/Robotics)
- Yapeng Tian (Audio-visual Scene Understanding)
- Jessica Ouyang (NLP)
- Yunhui Guo (Computer Vision/Transfer Learning)
- Feng Chen (Data mining/Graph mining)
- Xinya Du (NLP)

And More!

Please evaluate the course!

eval.utdallas.edu