Combining Factorization Model and Additive Forest for Recommendation

Presenter: Tianqi Chen

Team ACMClass@SJTU

August 11, 2012

Team ACMClass@SJTU

- Original team name: undergrads
- Members are students from ACMClass in SJTU
- All members are undergraduates, except the presenter:)

Outline

Overview of General Approach

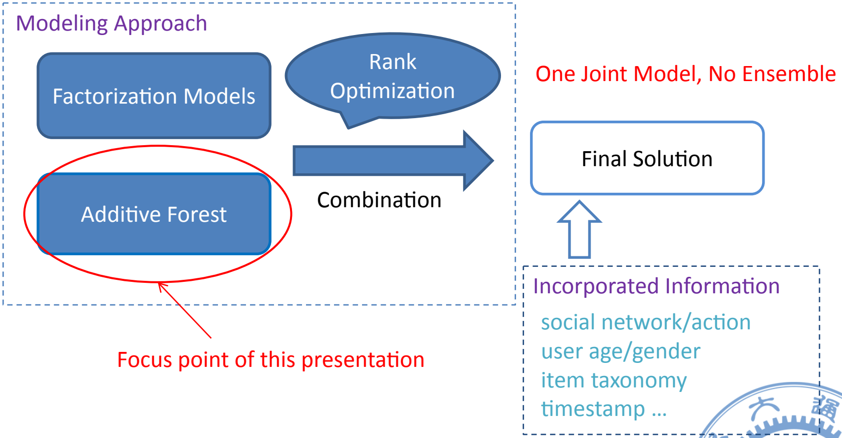
Go Beyond Factorization Models

More Example Models used in Solution

Results and Conclusion

数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Overview of Our Solution



Feature-based Matrix Factorization

$$\hat{r}_{ui} = \left(\sum_{c \in C(u)} \alpha_c^{(u)} \mathbf{p}_c\right)^T \left(\sum_{c \in C(i)} \beta_c^{(i)} \mathbf{q}_c\right) + \sum_{c \in C(u,i)} \gamma_c^{(u,i)} g_c \quad (1)$$

- $\Theta = \{\mathbf{p}, \mathbf{q}, g\}$, trained via stochastic gradient descent
- $\alpha_c^{(u)}$: user feature of user u: user social network/action, keyword/tag
- $\beta_c^{(i)}$: item feature weight of item(celeberity) i: item taxonomy/network
- $\gamma_c^{(u,i)}$: global feature related to interaction between u and i: user age/gender bias

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Outline

Overview of General Approach

Go Beyond Factorization Models

More Example Models used in Solution

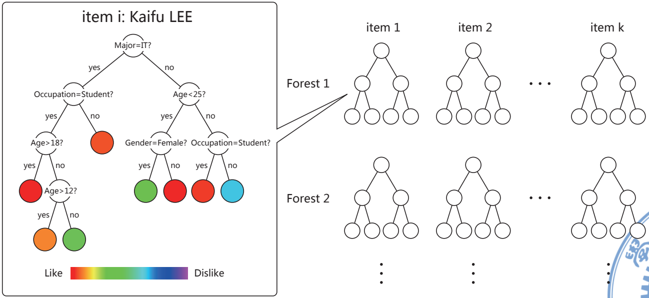
Results and Conclusion

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

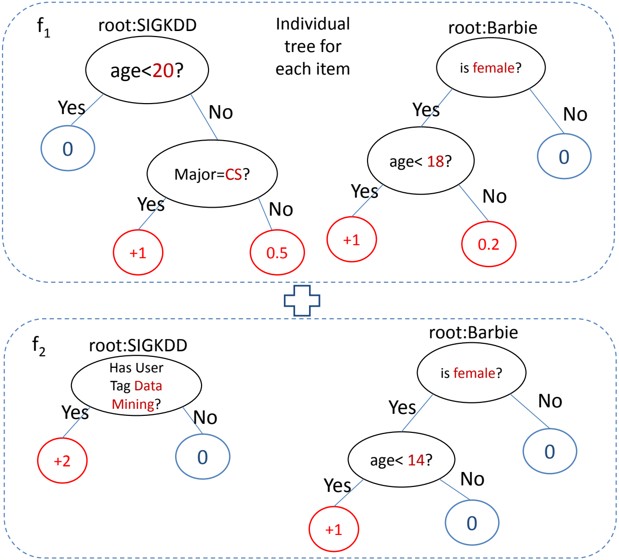
Additive Forest

$$\hat{r}_{ui} = \sum_{s=1}^{S} f_{s,root(i,s)}(x_u) \quad (2)$$

- x_u: property feature of user u
- $f_{s,root(i,s)}$: function defined by a regression tree
- Learning via gradient boosting algorithm



An Example of Additive Forest



Forest 1

Forest 2 is learned to complement Forest 1

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Factorization Model vs Additive Forest

	Factorization	**Additive Forest**
handling of sparse matrix data	very well	capable, not very well
combination of different information	linear combination	nonlinear composition
handling of continuous property	need predefined segmentation	automatic segmentation
model complexity control	regularization	feature selection, prunning

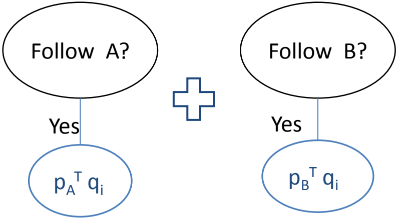
- Both models have their own advantages on different aspect
- Understanding their properties and knowing when to use which one is very important

Information Combination: User Social Network

Factorization Model

$$\hat{r}_{ui} = \left(\frac{1}{\sqrt{|F(u)|}} \sum_{j \in F(u)} \mathbf{p}_j \right)^T \mathbf{q}_i$$

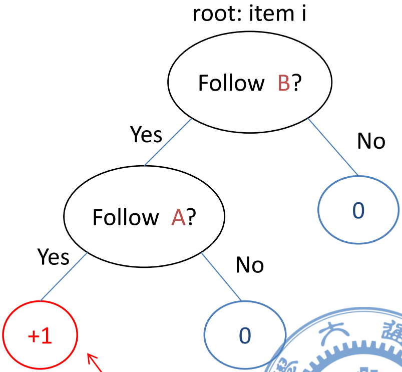
- $F(u)$: follow set of u
- Linear combination



Score for users who
"Follow both A and B" = $p_A^T q_i + p_B^T q_i$

Additive Forest

- Condition composition
- Feature selection



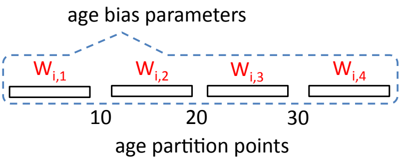
Specific score for
condition *"Follow A and B"*

Continuous Feature Handling: User Age

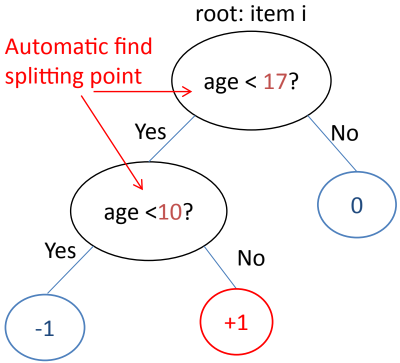
Factorization Model

$$\hat{r}'_{ui} = \mathbf{p}_u^T \mathbf{q}_i + W_{i,ag(u)} \qquad (3)$$

- $ag(u)$: age segment index
- Require predefined partition



Additive Forest



Factorization Model vs Additive Forest

	Factorization	Additive Forest
handling of sparse matrix data	very well	capable, not very well
combination of different information	linear combination	nonlinear composition
handling of continuous property	need predefined segmentation	automatic segmentation
model complexity control	regularization	feature selection, prunning

- Both models have their own advantages on different aspect
- Understanding their properties and knowing when to use which one is very important

Outline

数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Time-aware Model

Traditional Time Bin Model

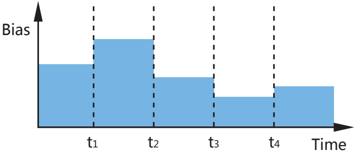
$$\hat{r}'_{ui}(t) = \hat{r}_{ui} + b_{i,binid(t)}$$

- $binid(t)$: time bin index of t

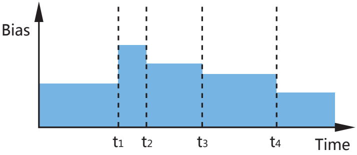
Our Time-aware Model

$$\hat{r}'_{ui}(t) = \hat{r}_{ui} + \sum_{s=1}^{S} f_{s,i}(t)$$

- $f_{s,i}(t)$: k-piece step function



(a) Item Time Bin



(b) K-piece Step Function

Figure: Comparison of Two Temporal Models

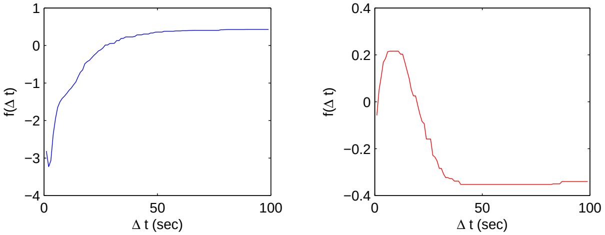
APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

User Sequential Pattern

$$\hat{r}'_{ui}(t) = \hat{r}_{ui} + \sum_{s=1}^{S} f_s(x_{seq}) \tag{4}$$

Features include in x_{seq}:

- time difference between clicks
- average click speed of current user



(a) $\Delta t = t_{next} - t_{curr}$ (b) $\Delta t = t_{curr} - t_{prev}$

Figure: Single Variable Pattern $\sum_{s=1}^{S} f_s(\Delta t)$

数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Final Model

$$\hat{r}_{ui} = \left(\sum_{c \in C(u)} \alpha_c^{(u)} \mathbf{p}_c\right)^T \left(\sum_{c \in C(i)} \beta_c^{(i)} \mathbf{q}_c\right) + \sum_{c \in C(u,i)} \gamma_c^{(u,i)} g_c + \sum_{s=1}^{S} f_{s,root(s,i)}(x_{ui}) \tag{5}$$

- Combination of all the factorization model and additive forest
- Boosting from result of factorization part

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Outline

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Experiment Results

ID	model	public	private	Δ_{public}	$\Delta_{private}$
1	item bias	34.6%	34.0%		
2	1 + user follow/action	36.7%	35.8%	2.1%	1.8%
3	2 + user age/gender	38.0%	37.2%	1.3%	1.4%
4	3 + user tag/keyword	38.5%	37.6%	0.5%	0.4%
5	4 + item taxonomy	38.7%	37.8%	0.2%	0.2%
6	5 + time-aware model	39.0%	37.9%	0.3%	0.1%
7	6 + age/gender(forest)	39.1%	38.0%	0.1%	0.1%
8	7 + sequential patterns	44.2%	42.7%	5.1%	4.7%

Table: MAP@3 of different methods

- User Modeling and Sequential Patterns contributes the most
- Time-aware model is more effective in public data
- **All of them** are important for winning

数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Summary

- Seems Ensemble methods **do not** work in our experiment
- Choose right methods to utilize different kinds of data
  - Factorization models are powerful, but also have drawbacks
  - Additive forest can automatic cut the continuous features, sometimes smarter than human
- Use automatic cutting to build robust time-aware model
- Fully utilize the available information
- Source code: *svdfeature.apexlab.org*

Thank You, Questions?

Appendix

- The rest parts of the slides are appendix

Objective Function

- Loss function of Pairwise Ranking: AUC optimization

$$L_u = \frac{1}{|\{(i,j)|r_{ui} > r_{uj}\}|} \sum_{(i,j):r_{ui}>r_{uj}} \mathcal{C}(\hat{r}_{ui} - \hat{r}_{uj}) \qquad (6)$$

- Pseudo loss function of LambdaRank: MAP optimization

$$L_u = \frac{1}{|\{(i,j)|r_{ui} > r_{uj}\}|} \sum_{(i,j):r_{ui}>r_{uj}} |\Delta_{ij}MAP|\mathcal{C}(\hat{r}_{ui} - \hat{r}_{uj}) \qquad (7)$$

  - $\Delta_{ij}MAP$ is MAP change when we swap i and j in current list
- $\mathcal{C}(x)$ is a surrogate convex loss function
  - logistic loss(BPR): $\mathcal{C}(x) = \ln(1 + e^{-x})$
  - hinge loss(maximum margin): $\mathcal{C}(x) = max(0, 1 - x)$
- L_u is normalized by number of pairs($|\{(i,j)|r_{ui} > r_{uj}\}|$): **Balance** over all users is important

BiLinear Model

$$\hat{r}_{ui} = \mathbf{x}_u{}^T \mathbf{W} \mathbf{y}_i \tag{8}$$

- $\mathbf{W}$: weight matrix
- $\mathbf{x}_u$: property vector of user u
- $\mathbf{y}_i$: property vector of item i

Example: Social aware Model

$$\hat{r}_{ui} = \frac{1}{\sqrt{|F(u)|}} \sum_{c \in F(u)} W_{c \to i},\ x_{uc} = \begin{cases} \frac{1}{\sqrt{|F(u)|}} & c \in F(u) \\ 0 & c \notin F(u) \end{cases},\ y_{uc} = \mathbf{e}_i \tag{9}$$

- $W_{c \to i}$: confidence of rule u follows $c \to u$ accept i

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Factorization Model vs BiLinear Model

BiLinear

$$\hat{r}_{ui} = \mathbf{x}_u{}^T \mathbf{W} \mathbf{y}_i$$

Factorization

$$\hat{r}_{ui} = \mathbf{x}_u{}^T \mathbf{P}^T \mathbf{Q} \mathbf{y}_i$$

- Feature-based matrix factorization can be viewed as a *factorized* version of bilinear model.
- Advantage of $\mathbf{W}$: direct modeling effect of $c \rightarrow i$
- Advantage of $\mathbf{P}^T\mathbf{Q}$: less parameter, topic level matching
  - When $\mathbf{W}$ is large and with sparse data support, use factorization
  - When $\mathbf{W}$ is small and with dense data support, use bilinear

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

User Social Network and Action

$$\hat{r}_{ui} = \left(\frac{1}{\sqrt{|F(u)|}} \sum_{j \in F(u)} \mathbf{p}_j + \frac{1}{\|\alpha_u\|_2} \sum_{j \in A(u)} \alpha_{u,j} \mathbf{y}_j \right)^T \mathbf{q}_i + b_i \quad (10)$$

- $F(u)$: set of items user u followed
- $A(u)$: set of items user u has action with
- α_u: weight by action count

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Item Taxonomy and Social Network

Taxonomy

$$\mathbf{q}'_i = \mathbf{q}_i + \mathbf{q}_{c^1(i)} + \mathbf{q}_{c^2(i)} + \mathbf{q}_{c^3(i)} + \mathbf{q}_{c^4(i)} \tag{11}$$

- Taxonomy aware parameter sharing
- $c^k(i)$: k-th level category of item i belongs to

Social Network

$$\mathbf{q}'_i = \mathbf{q}_i + \sum_{j \in cofollow(i)} \mathbf{q}_j \tag{12}$$