Our Solution on Track 1:

A Linear Ensemble of Individual and Blended Models for Music Rating Prediction

Po-Lung Chen, Chen-Tse Tsai, Yao-Nan Chen, Ku-Chun Chou, Chun-Liang Li, Cheng-Hao Tsai, Kuan-Wei Wu, Yu-Cheng Chou, Chung-Yi Li, Wei-Shih Lin, Shu-Hao Yu, Rong-Bing Chiu, Chieh-Yen Lin, Chien-Chih Wang, Po-Wei Wang, Wei-Lun Su, Chen-Hung Wu, Tsung-Ting Kuo, Todd G. McKenzie, Ya-Hsuan Chang, Chun-Sung Ferng, Chia-Mau Ni, Hsuan-Tien Lin, Chih-Jen Lin and Shou-De Lin

National Taiwan University

Three Properties of Track 1 Data

	track_1	track_2	album_3	author_4	$\cdots$	genre_I
user_1	$(100, t_{11})$	$(80, t_{12})$	$(70, t_{13})$	$(?, t_{14})$	$\cdots$	–
user_2	–	$(0, t_{22})$	$(?, t_{23})$	$(80, t_{24})$	$\cdots$	–
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
user_U	$(?, t_{U1})$	–	$(20, t_{U3})$	–	$\cdots$	$(0, t_{UI})$

similar to Netflix data, but with the following differences......

- scale: larger training and test sets

  training: study mature models that are computationally feasible;
  test: linearly combine many models w/o much overfitting

- taxonomy: relation graph of tracks, albums, authors and genres

  include as features for combining models nonlinearly

- time: detailed; training earlier than validation earlier than test

  include as features for combining models nonlinearly;
  respect time-closeness during training & with val.-set blending

Selected Ideas that Did Not Work: Deal with Zero-Variance Users

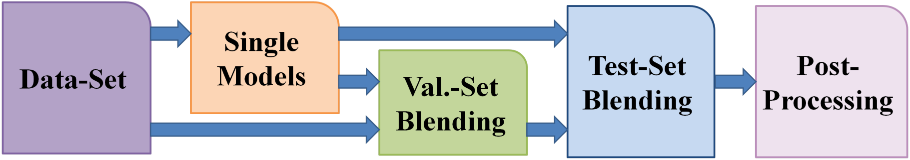
Background

- zero-variance users (7% of all users)
  —if a user gives 60, 60, 60, $\cdots$ in all training ratings, how'd she rate the next item?
- Occam's razor prediction: 60
  —only true for 80% of users, 20% changed their mind!

Idea

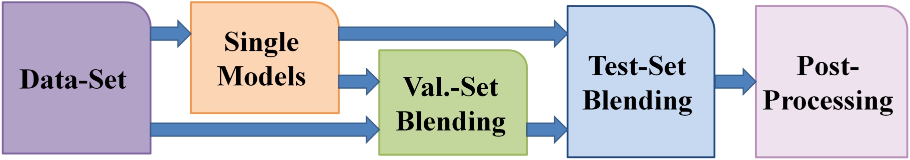
- conditionally (the 80%) post-process the predictions
- difficult to distinguish and thus failed

Framework of Our Solution



- single models—computationally feasible models that are diverse:
  - individual models: matrix factorization (& pPCA), pLSA
  - residual models: R-Boltz. machine, k-NN
  - derivative model: regression with statistical & model-based features
- validation-set blending:
  combine models nonlinearly while respecting time-closeness
- test-set blending:
  combine models linearly while fitting the leaderboard feedback
- post processing:
  polish predictions using findings during data analysis

RMSE Performance at Each Stage of Framework



- single models: 22.7915
  - individual models: best RMSE 22.9022 (MF)
  - residual models: best RMSE 22.7915 (k-NN + MF)
  - derivative model: best RMSE 24.1251 (but helps in later stages)
- validation-set blending: 21.3598 [improvement 1.4317]
- test-set blending: (estimated) 21.0253 [improvement 0.3345]
- post processing: 21.0147 [improvement 0.0106]

both blending stages: key to the system

Glance of Single Model RMSE

model	# used	best	average	worst	contribution
MF	81	22.90	23.92	26.94	**0.3645**
pPCA	2	24.46	24.61	24.75	0.0014
pLSA	7	24.83	25.53	26.09	0.0042
R-Boltz. machine	8	22.80	24.75	26.08	0.0314
k-NN	18	22.79	25.06	42.94	0.0298
regression	10	24.13	28.01	35.14	0.0261

- contribution (**before val.-set blending**):
  estimated RMSE diff. via leave-the-model-out in test-set blending
- MF: most important (absorbing pPCA)
- residual models: both quite important
- derivative model: individually weak but adds diversity

val.-set blending:
95 models, best 21.36, average 23.53, worst 31.70

Selected Ideas that Worked (1/5): Time Emphasis in Stochastic Gradient Descent

Background

SGD for minimizing sum of per-example $E_n(\boldsymbol{\theta})$ (say, for MF):

- randomly pick one example n
- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla E_n(\boldsymbol{\theta})$

Idea

- last M steps of SGD: effectively considering only the last M examples picked—**final θ *as if* biased towards those**
- need: $\boldsymbol{\theta}$ respects time-closeness to the test examples
- heuristic: deterministically pick the “newer” examples as last

consistent $\approx$ 0.05 RMSE improvement for MF

Selected Ideas that Worked (2/5): Gaussian RBM as Residual Model

Background

- RBM: a recursive NNet; can be used as an individual model by

discrete hidden factors	
↑	↑ ↓
per-user incomplete discrete ratings	predicted continuous ratings

- as individual: RMSE 24.7433, worse than MF (22.9974)

Idea

- MF (a first-order model) efficiently gets better performance, but can RBM digest something different?
- need: RBM that learns from the residuals of MF (continuous values)

Selected Ideas that Worked (2/5): Gaussian RBM as Residual Model

Background

discrete hidden factors	
↑	↑ ↓
per-user incomplete discrete ratings	predicted continuous ratings

Idea

- need: RBM that learns from the residuals of MF
- choice: Gaussian RBM (gRBM)

discrete hidden factors	
↑	↑ ↓
per-user incomplete continuous residuals	predicted continuous residuals

MF+gRBM: 22.8008;
better than individual MF (22.9974) or RBM (24.7433)

Selected Ideas that Worked (3/5): Multi-Feature and Multi-Stage Binned Lin. Reg.

Background

- Binned Linear Regression: a conditional aggregation model
- different model strength on different "types" of examples
- different blending weights for different types (**bins**) to utilize strength

bins	# rating $\leq \theta_1$	$\theta_1 <$ # rating $\leq \theta_2$	others
weight of MF-1	0.4	0.7	1.0
weight of RBM-1	0.5	0.1	0.0
weight of RBM-2	0.1	0.2	0.0

- a simplified regression tree with one level (on one feature)

Selected Ideas that Worked (3/5): Multi-Feature and Multi-Stage Binned Lin. Reg.

Background

- Binned Linear Regression
  —different blending weights for different (types) bins of examples

Idea: **multi-feature** BLR

- rationale: “type” more sophisticated than 1-feature bin
- a special multi-level decision tree
- prevent overfitting by limiting height and bin size
- heuristic algorithm instead of traditional decision tree:
  due to simplicity by extending from one-feature BLR

multi-feature	1-feature	4-feature	6-feature
RMSE	22.0829	21.8605	21.8128

Selected Ideas that Worked (3/5): Multi-Feature and Multi-Stage Binned Lin. Reg.

Background

- Binned Linear Regression
  —different blending weights for different (types) bins of examples

Idea: **multi-stage** BLR

- rationale: more diverse but good models before test-set blending

bins	1	2	3
weight of MF-1	...	...	...
weight of RBM-1	...	...	...
weight of RBM-2	...	...	...
weight of BLR-1	...	...	...
weight of BLR-2	...	...	...

multi-stage	1-stage	2-stage	3-stage
RMSE	21.7140	21.4591	21.4287

Selected Ideas that Worked (4/5): Offline Test Performance Predictor

Background

- given: columns $\mathbf{z}_m$ = test-set prediction of model m
- test-set linear regression:

$$\mathbf{w}(\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M, \lambda) = (\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I})^{-1}\mathbf{Z}^T\mathbf{r}$$

- true ratings $\mathbf{r}$ unknown but $\mathbf{z}^T\mathbf{r}$ can be estimated by

$$\begin{aligned} 2\mathbf{z}^T\mathbf{r} &= \mathbf{z}^T\mathbf{z} + \mathbf{r}^T\mathbf{r} - (\mathbf{z}-\mathbf{r})^T(\mathbf{z}-\mathbf{r}) \\ &\approx \mathbf{z}^T\mathbf{z} + N \cdot \text{RMSE}(\mathbf{0})^2 - N \cdot \text{RMSE}(\mathbf{z})^2 \end{aligned}$$

- common technique for RMSE ever since Netflix competition

Selected Ideas that Worked (4/5): Offline Test Performance Predictor

Background

$$\begin{aligned} 2\mathbf{z}^T\mathbf{r} &= \mathbf{z}^T\mathbf{z} + \mathbf{r}^T\mathbf{r} - (\mathbf{z}-\mathbf{r})^T(\mathbf{z}-\mathbf{r}) \\ &\approx \mathbf{z}^T\mathbf{z} + N \cdot \text{RMSE}(\mathbf{0})^2 - N \cdot \text{RMSE}(\mathbf{z})^2 \end{aligned}$$

Idea

- want: decide which $\mathbf{z}_m$'s and λ to use
- restriction: one submission every eight hours
- solution: estimate RMSE of $\mathbf{w}$ without submitting more than $\mathbf{z}_m$

$$N \cdot \text{RMSE}(\mathbf{w})^2 = (\mathbf{Z}\mathbf{w}-\mathbf{r})^T(\mathbf{Z}\mathbf{w}-\mathbf{r}) = \mathbf{w}^T\mathbf{Z}^T\mathbf{Z}\mathbf{w} - 2\mathbf{w}^T\mathbf{Z}^T\mathbf{r} + \mathbf{r}^T\mathbf{r}$$

compute the contribution of models;
choose 221 from ≈ 300 models & decide $\lambda = 10^{-6}$ offline

Selected Ideas that Worked (5/5): Clipping for Old Four-Star Days

Background

- some very different rating systems observed during data analysis:
  - four-star rating? $\{0, 30, 50, 70, 90\}$
  - five-star rating? $\{0, 20, 40, 60, 80, 100\}$
  - 100-point scale
- suspect changes in the user interface of Yahoo! Music

Idea

- existing: in five-star or 100-point scale, clip prediction to $[0, 100]$
- new: for four-star, clip prediction to $[0, 90]$
- what dates? $[3365, 5982]$ (7 years) or $[4281, 6170]$ (5 years)

$\approx$ 0.02 RMSE improvement on most models

Summary

- NTU team: 1 class, 19 students, 3 TAs, 3 professors
- shared techniques between two tracks:
  - models: MF, k-NN, pLSA
  - concept of diversity and blending
  - taxonomy information (more for track 2)
- special techniques in track 2:
  - construct suitable learning problems and (new) models from raw data
  - sample proper validation sets
- special techniques in track 1:
  - respect time-closeness
  - blend deeply with validation set and broadly with test set

Acknowledgments

We truly thank

- organizers for designing a successful competition
- NTU EECS college and CSIE department for support

Thank you. Questions?