# Scorecard with Latent Factor Models for User Follow Prediction Problem

## KDD cup 2012 – track 1

Xing Zhao, Ph.D.
Senior Scientist
Model Builder Product Development

**August, 2012**

# Agenda

» Problem description

» Scorecard introduction

» Feature generation

» Latent factor models

» Results and discussions



2
© 2009 Fair Isaac Corporation. Confidential.

# Background

FICO Model Builder

- FICO’s internal tool used to build models
- software product for sale to customers

Use Model Builder for KDD cup problem?

- Performance to handle big size data?
- Scorecard to ensemble scores from CF models?

Self-motived project

- First two months, after work, fun
- Last three weeks, work time, tough competition

# Problem Description

Task

- predict if a user will follow a recommended item

Evaluation

- average precision

Data

- rec_log_train, rec_log_test
- user_profile, user_key_word, user_sns, user_actions, item

Solution

- Generate predictive features
- Train latent factor models
- Train scorecard model to ensemble latent factor model scores and predictive features.

# Development Environment

Model Builder scripting environment on linux

Groovy scripts for simple processing

Java for latent factor model training and feature generation

Frequently used Model Builder functions

- dataset, data analysis
- binning, scorecard,
- model performance, reporting

# Training Data Setup

Raw train data: 31 days, 70M records

remove duplicates and dummy users => 36M

Split => train1 (first 22 days, 26M) + train2 (last 9 days, 10M)

User based split for train2 => train2_a (9M) + trian2_b (1M)

train1 + train2_a (35M)

- training data for latent factor models
- feature generation

train2_b (1M)

- validation for latent factor models
- training data for scorecard

# Average Precision

Introduce record weight to address average precision indirectly

- For each user, balance the weight of follow records and recommend records
- Adjust user's importance based on user's total follow records.

$H_u$: the count of follow records for user u in training data

$L_u$: the count of recommend records for user u in training data.

$W_u$: record weight used in latent factor model and scorecard

$1/H_u$ for follow record

$1/L_u$ for recommend record

# Scorecard

## Miniature Example of a Scorecard

| Charact. j | Bin k | Description | Score Weight |
|---|---|---|---|
| 1 | | *# Late Payments last 9 months* | |
| | 1 | 0 | 20 |
| | 2 | 1 | 10 |
| | 3 | 2 or more | 5 |
| 2 | | *Age of account* | |
| | 1 | below 1 year | 5 |
| | 2 | 1-2 years | 10 |
| | | etc. | |
| 3 | | *Debt Ratio* | |
| | 1 | 0-30 | 15 |
| | 2 | 30-50 | 10 |
| | 3 | 50-70 | 5 |
| | | etc. | |



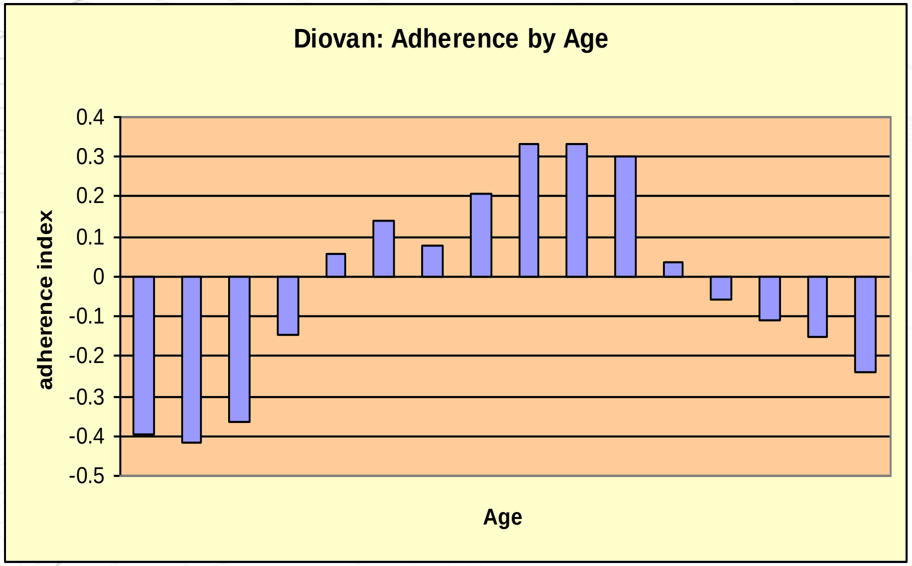
* Simulated figures for illustrative purposes only

# Scorecard

- Scorecard is additive among variables

$$score = w_0 + f_1(x_1) + f_2(x_2) + \ldots + f_N(x_N)$$

- f(x) can capture nonlinear relation from each variable



# Variable Interactions

Scorecard is additive,  but real relationship may have interactions

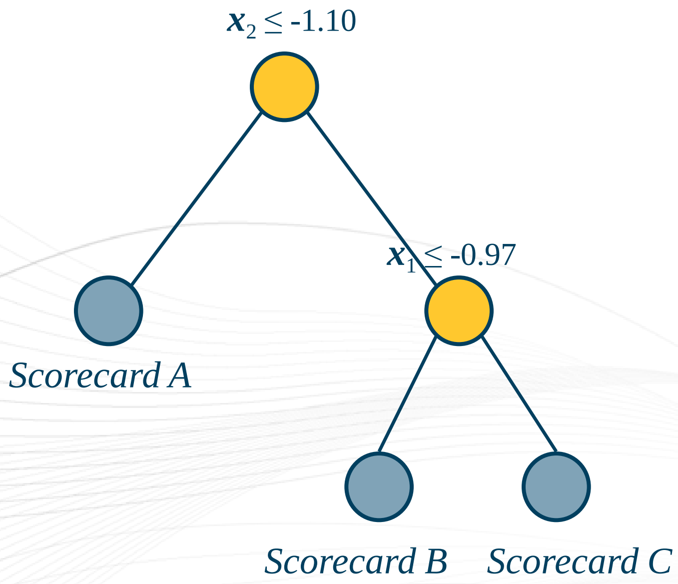
$$score = w_0 + f_1(x_1) + f_2(x_2) + \ldots + f_N(x_N)$$

$$\text{target} = f(x_1, x_2, \ldots, x_N)$$

=>  Segmented Scorecards

=>  Scorecard with cross binnings

# Segmented Scorecard

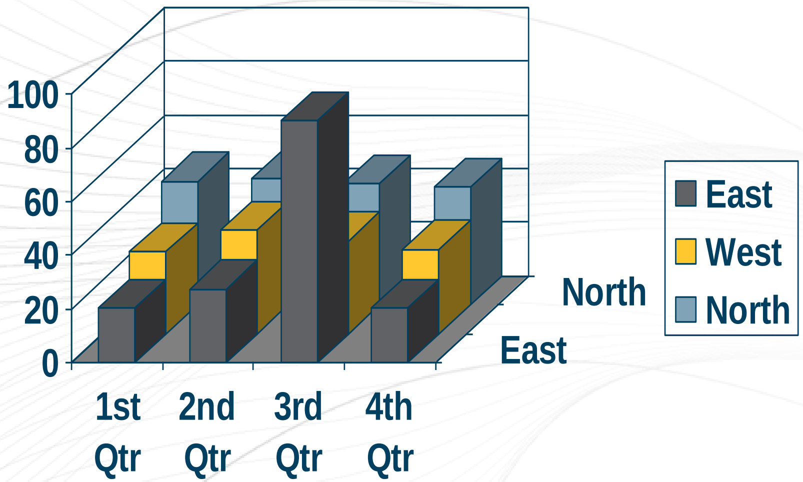



Segmented scorecard can handle variable interactions

FICO score is built using segmented scorecard

# Scorecard with Cross Binning

cross binning for pairwise interaction $f_{ij}(x_i, x_j)$




$$score = w_0 + f_1(x_1) + f_2(x_2) + \ldots + f_N(x_N) + \sum_{(i,j)} f_{ij}(x_i, x_j)$$

# Pairwise Interaction Selection

For each variable pair, build two models

$$scoreA = w_0 + f_{ij}(x_i, x_j)$$

$$scoreB = w_0 + f_i(x_i) + f_j(x_j)$$

Model performance difference is because of variable interaction

Select variable pairs with bigger performance difference

# Predictive Features

Session variables based on time stamp, batch and session

- pre_batch:  time from the last batch of records
- next_batch:  time to the next batch of records
- session_start: time to the start of the session
- session_end: time to the end of the session
- session_start_batches:  batches to the start of the session
- session_end_batches:  batches to the end of the session

next_batch is a very strong predictor

Pairwise interactions detected

- (next_batch, pre_batch)
- (next_batch, session_end_batches)
- (next_batch, session_start_batches)

# Predictive Features

Item popularity based on age-gender group

- users are divided into 15 groups based on age and gender combination.
- group based item popularity is a stronger predictor

Some other features that are weak predictors

Item based KNN

Item category

Keyword match

# Latent Factor Models

Each user is represented through other elements indirectly.

- Followed_items
- Action_users
- Keywords
- Tags

Latent factors introduced for these elements and items

$$\hat{r}_{ui} = u + u_u + u_i + q_i^T \left( \frac{1}{\sqrt{|I(u)|}} \sum_{j \in I(u)} q_j^0 \right)$$

# Latent Factor Models

Record weights for average precision

Age-gender group based item average

Average precision measure for validation data

Combo latent factor models

- followed_item + key_word+ tag
- key_word + tag,   followed_item + key_word …

# Results

Features only: 0.390

Add latent factor models: 0.422

Add cross binnings: 0.428

Best single predictor: 0.384

- combo latent factor model (followed_item + keyword + tag)
- age-gender group based item average

# Summary and Discussions

Scorecard

- ensemble features and latent factor model scores
- deal with pairwise feature interactions

Discussions

- better way to handle average precision?
- tuning for latent factor models?
- train latent factor models on data segments?

# Thanks

FICO Model Builder product development team for their support

organizers for a successful competition