Opera Solutions, LLC
12230 El Camino Real
Suite 330San Diego, California
92130+1 (858) 480 3750
telephone+1 (858) 480 3727
facsimile
www.operasolutions.com

KDD Cup 2012 Track 2:

Ensemble of Collaborative Filtering and Feature Engineered Models for Click Through Rate Prediction

—Methods of Opera Solutions

NOTICE: Proprietary and Confidential
This material is proprietary to Opera Solutions. It contains trade secrets and confidential information which is sole property of Opera Solutions. This material is solely for the Client's internal use. This material shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part, without the express written consent of Opera Solutions.
© 2012 Opera Solutions, LLC. All rights reserved.

The Dream Team

- Jeong-Yoon Lee
- Jingjing(Bruce) Deng
- Hang Zhang
- Jacob Spoelstra
- Andreas Töscher
- Michael Jahrer



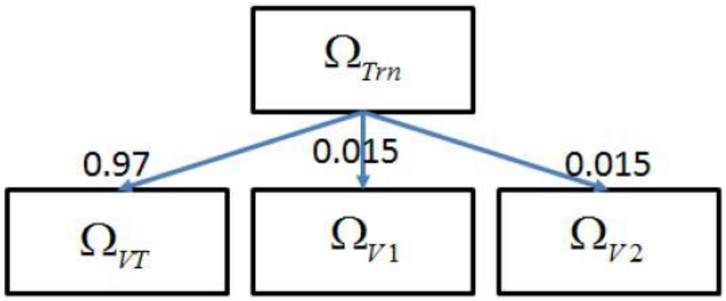



The Task

- Predicting the click-through rate (CTR) a search advertisement receives from a querying user
  - Search advertising has been one of the major revenue sources of the internet industry
  - Predicting CTR correctly helps search providers to rank/price ads correctly
  - Important to user experience improvements and revenue growth
  - Widely applicable to searching engines, online stores, online finance services, etc.
  - Evaluation metric: Area Under ROC Curve (AUC)

Preparing the data for learning

- We do some basic checks
- Decide to use random 3% of train as valid
  - Split 1.5% to Valid1
  - Split 1.5% to Valid2



- Main data table

clicks	impr	adUrlID	adID	adverID	depth	pos	queryID	keyWordID	titleID	descrID	userID
0	1	12673870462623600000	4242983	26519	2	1	47350	812	8842	25537	6023881
0	1	6399024617856670000	21299603	36491	2	2	546	113	3225	121	6023881
1	1	12673870462623600000	4242983	26519	1	1	47350	812	9164	7625	6023881
0	1	6877516134389990000	20053263	2332	2	1	23447	476	3547	3397	2583834
0	4	6360809004806360000	10164628	18209	2	2	4035850	947	74709	37226	2583834
0	4	11659373614241500000	20934246	34882	2	1	4035850	592434	1507528	4127	2583834
1	1	17697127834337800000	10484162	29135	2	1	1788197	147838	971553	628205	2583834
0	2	4660387735928840000	21313239	36540	1	1	6600	11342	10208	1785	4019508
1	1	2670952723278900000	20172874	23805	2	2	5	3	35	16	4019508
0	1	7771884441258270000	20108617	32367	1	1	1315	316	177	64	4019508

... 150M records !! - 10Gig raw csv file + keywords + userProfiles

Opera's Approaches

- Individual models
  - Collaborative filtering (Bias model, Factor models)
  - Naïve Bayesian classifiers (NBC)
  - Feature engineering and advanced statistical models
- Blending (mix the individuals)
  - Weighted sum (linear)
  - Neural network

Collaborative filtering

- Sparse matrix
- What is the matrix ?
- What is the target ?

clicks	impr	adUrlID	adID	adverID	depth	pos	queryID	keyWordID	titleID	descrID	userID
0	1	12673870462623600000	4242983	26519	2	1	47350	812	8842	25537	6023881
0	1	6399024617856670000	21299603	36491	2	2	546	113	3225	121	6023881
1	1	12673870462623600000	4242983	26519	1	1	47350	812	9164	7625	6023881
0	1	6877516134389990000	20053263	2332	2	1	23447	476	3547	3397	2583834
0	4	6360809004806360000	10164628	18209	2	2	4035850	947	74709	37226	2583834
0	4	11659373614241500000	20934246	34882	2	1	4035850	592434	1507528	4127	2583834
1	1	17697127834337800000	10484162	29135	2	1	1788197	147838	971553	628205	2583834
0	2	4660387735928840000	21313239	36540	1	1	6600	11342	10208	1785	4019508
1	1	2670952723278900000	20172874	23805	2	2	5	3	35	16	4019508
0	1	7771884441258270000	20108617	32367	1	1	1315	316	177	64	4019508

- We have 10 ID sources (adUrlID, adID, advertiserID, depth, pos, queryID, keyWID, titleID, descrID, userID)
- userID x adUrlID ?
- userID x adID ?
- userID x advertiserID ?
- …
- …
- …

45 combinations

- Target: clicks/impressions

Bias model

clicks	impr	adUrlID	adID	adverID	depth	pos	queryID	keyWordID	titleID	descrID	userID
0	1	1267387046262360000	4242983	26519	2	1	47350	812	8842	25537	6023881
0	1	6399024617856670000	21299603	36491	2	2	546	113	3225	121	6023881
1	1	1267387046262360000	4242983	26519	1	1	47350	812	9164	7625	6023881
0	1	6877516134389990000	20053263	2332	2	1	23447	476	3547	3397	2583834
0	4	6360809004806360000	10164628	18209	2	2	4035850	947	74709	37226	2583834
0	4	1165937361424150000	20934246	34882	2	1	4035850	592434	1507528	4127	2583834
1	1	1769712783433780000	10484162	29135	2	1	1788197	147838	971553	628205	2583834
0	2	4660387735928840000	21313239	36540	1	1	6600	11342	10208	1785	4019508
1	1	2670952723278900000	20172874	23805	2	2	5	3	35	16	4019508
0	1	7771884441258270000	20108617	32367	1	1	1315	316	177	64	4019508

- Biases for every unique ID
  - approx. 50M biases
- Prediction is sum of M=10 biases

$$\boxed{\hat{p}_i = \sum_{m=1}^{M} b_k^m}$$

where $k = d_i^m$

Value of column=m and row=i in data

$d_1^1 = 1267387046262360000$ adUrlID, line=1

$d_1^2 = 4242983$ adID, line=1

$d_1^3 = 26519$ adverID, line=1

$d_1^4 = 2$ depth, line=1

...

- Training with stochastic gradient descent
  - Minimizing MSE
  - Small learning rate, L2 regularization (both optimized)
  - Public Leaderboard AUC: 0.76461

Bias model improved #1

- Same model

$$\widehat{p_i} = \sum_{m=1}^{M} b_{k}^{m} \quad \text{where } k = d_{i}^{m}$$

- **Separate learning rates η_m and regularizations λ_m for each of the 10 ID sources**

ID NAME	η	λ
ADURLID	0.000013	0.01
ADID	0.0001	0.0135
ADVERTISERID	0.0001	0.0379
DEPTH	0.000013	0.0379
POSITION	0.009	0.002
QUERYID	0.0025	0.0379
KEYWORDID	0.0001	0.002
TITLEID	0.0001	0.0135
DESCRIPTIONID	0.0001	0.137
USERID	0.0025	0.0075

- Training with stochastic gradient descent
  - Minimizing MSE
  - **Public Leaderboard AUC: 0.77336**

+0.009 AUC improvement

Bias model improved #2

+0.015 AUC improvement

- Same model

$$\widehat{p_i} = \sum_{m=1}^{M} b_{k}^{m} \quad \text{where } k = d_i^m$$

- Separate learning rates η_m and regularizations λ_m for each of the 10 ID sources

ID NAME	η	λ
ADURLID	0.000013	0.01
ADID	0.0001	0.0135
ADVERTISERID	0.0001	0.0379
DEPTH	0.000013	0.0379
POSITION	0.009	0.002
QUERYID	0.0025	0.0379
KEYWORDID	0.0001	0.002
TITLEID	0.0001	0.0135
DESCRIPTIONID	0.0001	0.137
USERID	0.0025	0.0075

- **Training with pairwise stochastic gradient descent**
  - Minimizing MSE on pairs – related to AUC maximization directly
  - **Public Leaderboard AUC: 0.788**

FOR e = 1...maxEpochs

FOR n = 1...N (all samples, e.g. N=150M for train set)

Select a sample: a=index to positive sample

Select b sample: b=index to negative sample

$\widehat{p_a} = \sum_{m=1}^{M} b_{d_a^m}^{m}$ a sample prediction

$\widehat{p_b} = \sum_{m=1}^{M} b_{d_b^m}^{m}$ b sample prediction

$\Delta_{pred} = \widehat{p_a} - \widehat{p_b}$ difference of predictions

$\Delta_{target} = t_a - t_b$ difference of targets

$error = \Delta_{pred} - \Delta_{target}$ the error

FOR m = 1...M (all 10 ID sources)

$k_a = d_a^m \qquad k_b = d_b^m$

$b_{k_a}^m = b_{k_a}^m - \eta_m \cdot (error + \lambda_m \cdot b_{k_a}^m)$ update the a and b sample biases

$b_{k_b}^m = b_{k_b}^m - \eta_m \cdot (-error + \lambda_m \cdot b_{k_b}^m)$

Bias model improved #3

- Same model

$$\widehat{p_i} = \sum_{m=1}^{M} b_k^m \quad \text{where } k = d_i^m$$

- Unroll the training set based on impressionCnt
  - From 150M to 235M training samples (+56% more training samples)
  - Use only 1 (+) or 0 (-) as targets

clicks	impr	adUrlID	adID	adverID	depth	pos	queryID	keyWordID	titleID	descrID	userID
0	1	1267387046262360000	4242983	26519	2	1	47350	812	8842	25537	6023881
0	1	6399024617856670000	21299603	36491	2	2	546	113	3225	121	6023881
1	1	1267387046262360000	4242983	26519	1	1	47350	812	9164	7625	6023881
0	1	6877516134389990000	20053263	2332	2	1	23447	476	3547	3397	2583834
0	4	6360809004806360000	10164628	18209	2	2	4035850	947	74709	37226	2583834
0	4	1165937361424150000	20934246	34882	2	1	4035850	592434	1507528	4127	2583834
1	1	1769712783433780000	10484162	29135	2	1	1788197	147838	971553	628205	2583834
0	2	4660387735928840000	21313239	36540	1	1	6600	11342	10208	1785	4019508
1	1	2670952723278900000	20172874	23805	2	2	5	3	35	16	4019508
0	1	7771884441258270000	20108617	32367	1	1	1315	316	177	64	4019508

0	4	6360809004806360000	10164628	1
0	4	6360809004806360000	10164628	1
0	4	6360809004806360000	10164628	1
0	4	6360809004806360000	10164628	1

e.g. if impressionCnt=4
-> unroll 1 data sample
to 4 +/- samples

- Gives also improvement
  - Unfortunately, we have no detailed notes

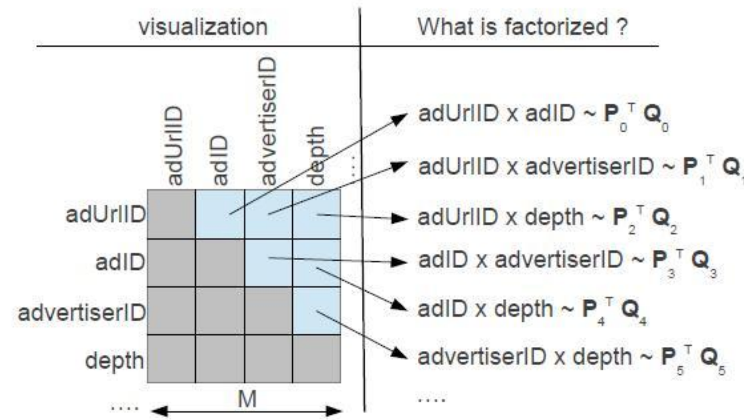
Factorized model

- Again, d_i^m is the value of the data at
  - m = the sourceID (1…10)
  - i = the sampleID (1…150M)
- The prediction is a sum of all dot products !

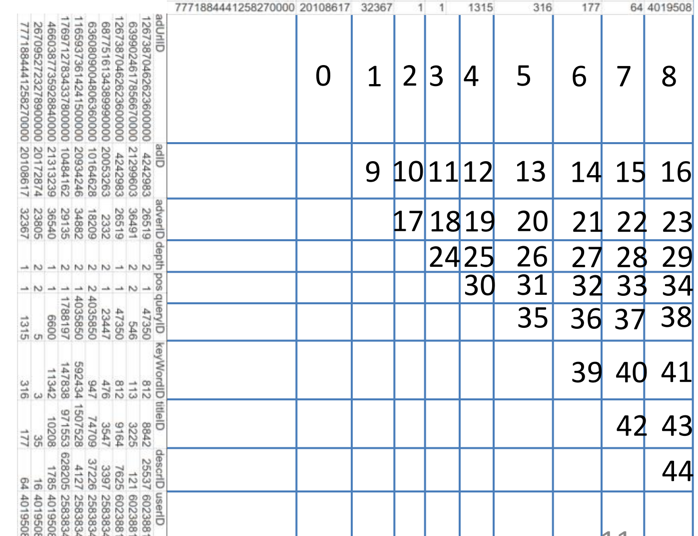
$$\boxed{\widehat{p_i} = \sum_{m=1}^{M} \sum_{n=m+1}^{M} p_{\left(d_i^m\right)}^{c\ T} \cdot q_{\left(d_i^n\right)}^{c}}$$

$$\widehat{p_i} = \underbrace{p_0^T q_0 + p_1^T q_1 + p_2^T q_2 + \ \ldots + p_{44}^T q_{44}}_{\text{45 dot products}}$$

- On every cell we have a feature matrix: F x $|d_*^m|$
  - F = number of features
  - e.g. P_0=F x 26272 P_1=F x 641706
  - Huge number of features !



adUrlID	adID	adverID	depth	pos	queryID	keyWordID	titleID	descrID	userID
12673870462623600000	4242983	26519	2	1	47350	812	8842	25537	6023881
6399024617856670000	21299603	36491	2	2	546	113	3225	121	6023881
12673870462623600000	4242983	26519	1	1	47350	812	9164	7625	6023881
6877516134389990000	20053263	2332	2	1	23447	476	3547	3397	2583834
6360809004806360000	10164628	18209	2	2	4035850	947	74709	37226	2583834
11659373614241500000	20934246	34882	2	1	4035850	592434	1507528	4127	2583834
17697127834337800000	10484162	29135	2	1	1788197	147838	971553	628205	2583834
4660387735928840000	21313239	36540	1	1	6600	11342	10208	1785	4019508
2670952723278900000	20172874	23805	2	2	5	3	35	16	4019508
7771884441258270000	20108617	32367	1	1	1315	316	177	64	4019508



Factorized model #2

- Very HUGE memory consumption
  - We were only able to train models with F=2 features
- Problems with overfitting
  - Error is minimal after 1 epoch of training !
  - High L2-regularization does not help
  - Too less time to do careful analysis

- Training with pairwise stochastic gradient descent
  - Minimizing pairwise MSE
  - Small learning rate, L2 regularization (both optimized)
  - **Public Leaderboard AUC: 0.7913**

Factorized model #3

- Added an 11th ID based on token overlap
  - # same tokens per instance: queryTokens -> {keywordTokens,titleTokens,descriptionTokens}
  - **Public Leaderboard AUC: 0.7945**
- Tried 12th ID based on
  - #pairs in tokens: hurts the model (but inside ensemble)

+0.003 AUC improvement

Other Collaborative filtering models tried

- KNN
  - Tried a few tweaks, but didn‘ help
- AFM
  - Uses features in „test set“ to learn !
  - Helps a little (0.0001 in blend)
  - Bad performance itsself (public leaderboard AUC 0.74xx)

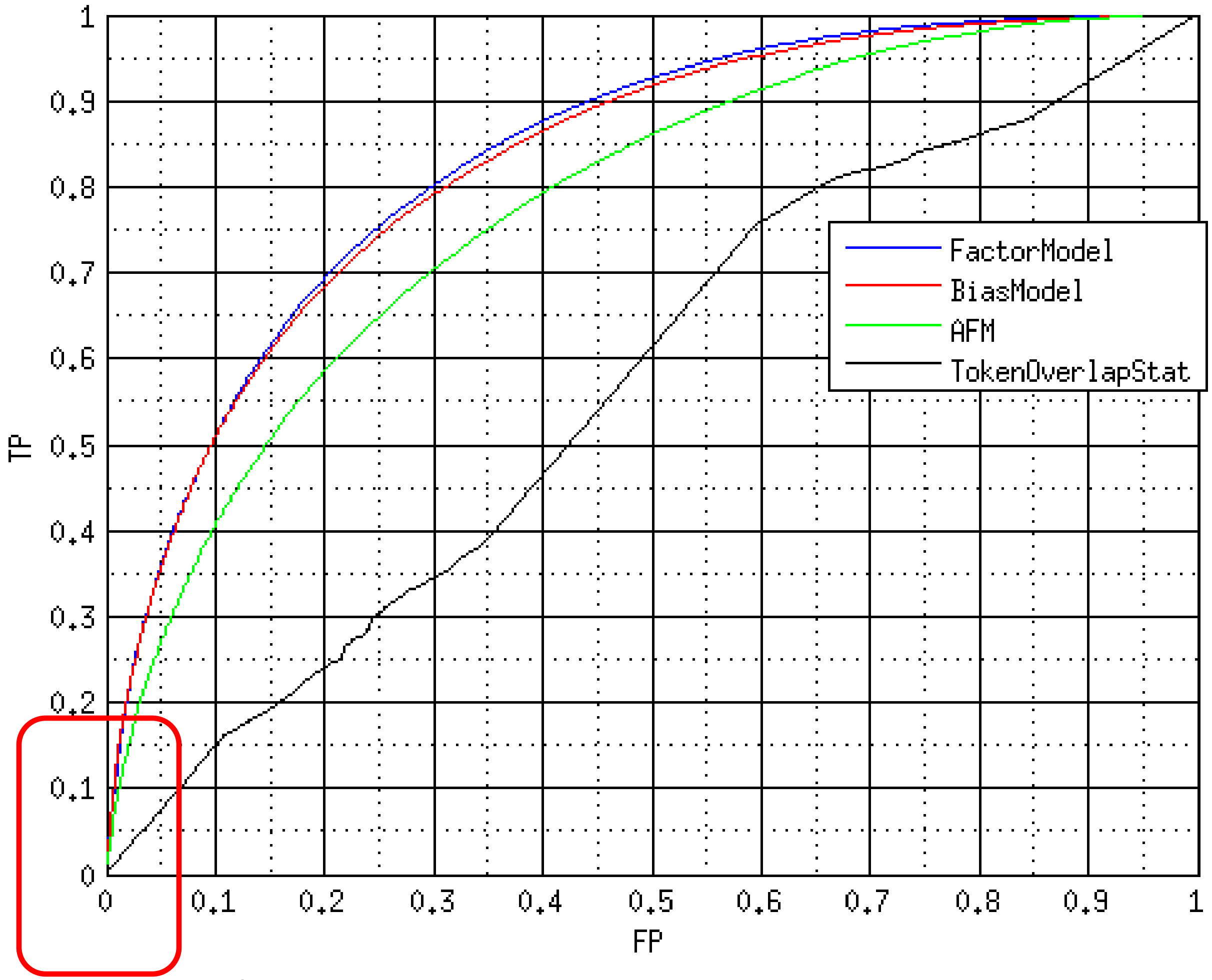
The prediction of a sample i was

$$\begin{aligned} \hat{r}_i &= \overbrace{\mathbf{p}^T}^{\text{"item feature"}} \cdot \overbrace{\mathbf{q}}^{\text{"user feature"}} \\ \mathbf{p} &= \{sum\ of\ 7\ features\ of\ sample\ i\} \\ \mathbf{q} &= \mathbf{q}_u + \sum_{j \in N(u)} \sum_{k=1}^{7} \{sum\ of\ 7\ features\ of\ sample\ j\} \end{aligned}$$

<u>7 features are:</u>

- adUrlID
- adID
- advertiserID
- queryID
- keyWordID
- titleID
- descriptionID

ROC curves comparisons



For classifiers, this is the important region -> operating point
But for Track2 unimportant, just area under the curve

<u>Pub. Leaderboard AUC's</u>
FactorModel: 0.795
BiasModel: 0.788
AFM: 0.74
TokenOverlapStat:0.57

CF observations and model tweaks

- Construct a 11th ID
  - tokenMatchID
  - Use it in bias model and factor model
- >50% of userIDs in the test set are unknown
  - Bad for user-based models
- Never clip predictions to 0...1
  - Can hurt in the final blend
- Every model is re-trained on the whole data before making predictions on the testset
- Use the tokenIDs in factor models
  - queryTokens, keywordTokens, titleTokens, descriptionTokens
  - Very small improvements in the blend
- Use gender and age codes
  - Very small improvements in the blend, if all
  - Hurts if we add this as new ID source in factor models
- We have problems with overfitting in the factor model, even if regularization is high
  - Back to F=1 features

Engineered Features

- Risk Features
  - 1D: conditional probability of click given an ID was present in a record.

$$Pr(Y=1|ID_i) = \frac{\sum_{j=1}^{n}(c_j+N_1)\times I(ID_i \in R_j)}{\sum_{j=1}^{n}(n_j+N_2)\times I(ID_i \in R_j)}$$

  - 2D: conditional probabilty of click given two IDs were present in a record.
  - 8 1D-risk features for adUrlID, adID, advertiserID, depth, position, userID, gender, age
  - 8 2D-risk features for {adID, advertiserID, depth, position} x {gender, position}

- Similarity Features
  - Overlap between tokens of queryID (ID1) and keywordID/titleID/descriptionID (ID2).
    - The proportion of the tokens in ID1 that are present in ID2 tokens.
    - The proportion of the 2-consecutive tokens in ID1 that are present in ID2.
    - If there exist common tokens between ID1 and ID2, their earliest position in ID2.
    - If there exist common 2-consecutive tokens between ID1 and ID2, their earliest position in ID2
  - 12 similarity features.

Feature Engineered Models

- Built on the engineered features
- Gradient Boosting Machine (GBM)
  - “gbm“ package in R was used.
  - Number of trees, shrinkage, and depth were chosen based on the validation errors.
  - AUC: 0.757
- Support Vector Machine (SVM)
  - SVM_perf was used.
  - AUC loss function, linear kernel, c = 500.
  - AUC: 0.764
- Neural Network (NN)
  - NN with AUC optimization was implemented in C.
  - Single hidden layer.
  - Other parameters were chosen based on the validation errors.
  - AUC: 0.765

Blending with a linear model

- Inputs
  - P Predictors (models) as a matrix with elements p_{nj}
  - Targets as a vector t
  - Features (pos, gender, age, tokenOverlaps, supports)
- Model
  - Weights w_j
  - $\widehat{p_i} = \sum_{j=1}^{P} w_j p_{nj} + w_0$ (w_0=0, because of pairwise ranking)
- Training
  - Gradient descent on pairs of samples
  - **Public Leaderboard AUC: 0.8030**

FOR e = 1...maxEpochs

FOR n = 1...N (all samples, e.g. N=3,430,641 for upsampled Valid1)

Select a positive sample: a=index to positive sample $t_{(+)}$=1

Select a negative sample: b=index to negative sample $t_{(-)}$=0

$\widehat{p_{(+)}} = \sum_{j=1}^{P} w_j p_{aj}$ (+) sample prediction

$\widehat{p_{(-)}} = \sum_{j=1}^{P} w_j p_{bj}$ (-) sample prediction

$\Delta_{pred} = \widehat{p_{(+)}} - \widehat{p_{(-)}}$ difference of predictions

$\Delta_{target} = t_{(+)} - t_{(-)}$ difference of targets

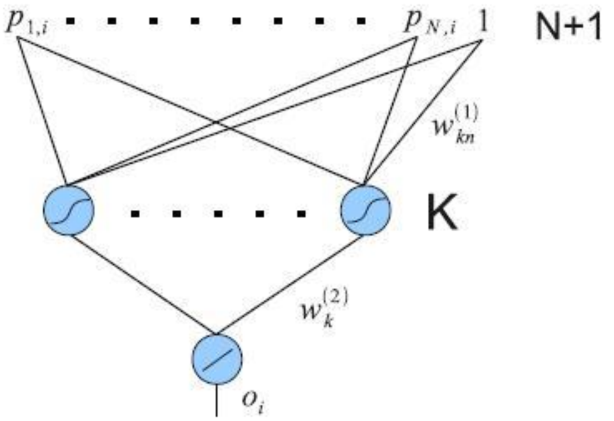
$error = \Delta_{pred} - \Delta_{target}$ the error

FOR j = 1...P (all predictors, e.g. P=57)

$w_j = w_j - \eta \cdot (error \cdot (p_{aj} - p_{bj}) + \lambda \cdot w_j)$ update the weights

Blending with a neural network

- Inputs
  - P Predictors (models) as a matrix with elements p_{nj}
  - Targets as a vector t
  - Features (pos, gender, age, tokenOverlaps, supports)
- Model
  - A single neural network, 1 hidden layer, K=20 units
  - $\widehat{p_i} = calcNN(p_{n*})$
- Training
  - Normalization of inputs to -1...+1
  - Gradient descent on pairs of samples
  - **Public Leaderboard AUC: approx. 0.80524 (0.80824 on private)**



FOR e = 1...maxEpochs

FOR n = 1...N (all samples, e.g. N=3,430,641 for upsampled Valid1)

Select a positive sample: a=index to positive sample $t_{(+)}$=1

Select a negative sample: b=index to negative sample $t_{(-)}$=0

$\widehat{p_{(+)}} = calcNN(p_{aj})$ (+) sample prediction

$\widehat{p_{(-)}} = calcNN(p_{bj})$ (-) sample prediction

$\Delta_{pred} = \widehat{p_{(+)}} - \widehat{p_{(-)}}$ difference of predictions

$\Delta_{target} = t_{(+)} - t_{(-)}$ difference of targets

$error = \Delta_{pred} - \Delta_{target}$ the error

Update the NN with both (+) and (-) sample
Using backprob rule

+0.002 AUC improvement to linear blending

Summary of Results

Model name	Performance on public leaderboard
Bias model (rank optimization)	0.788
Factor model (rank optimization)	0.795
AFM	0.745
NBC	0.77847
ANN optimizing AUC on feature metrics	0.76535

Ensemble methods	Performance on public leaderboard
Neural Network rank blend (1x20 neurons)	0.80524
Linear rank blend	0.803

It was very close on the private leaderboard !

KDD Cup 2012, Track 2

Information Data Forum **Results**

Public Leaderboard Private Leaderboard

This leaderboard is calculated on approximately 42% of the test data. The final results will be based on the other 58%, so the final standings may be different.

* in the money

#	Δ1w	Team Name	Kdd Ctr Auc	Entries	Last Submission UTC (Best Submission - Last)
1	-	Catch up *	0.80697	141	Fri, 01 Jun 2012 23:57:26 (-0.1h)
2	↑28	Opera Solutions *	0.80524	90	Fri, 01 Jun 2012 23:59:54 (-0.2h)
3	↑14	dsa! *	0.80508	84	Fri, 01 Jun 2012 19:56:49 (-13.6h)
4	new	Chinese Academy of Sciences	0.80343	25	Fri, 01 Jun 2012 23:53:17 (-0.1h)
5	↓3	Birutas & LARCA & Team DL	0.79872	200	Fri, 01 Jun 2012 23:37:41 (-0.7h)

KDD Cup 2012, Track 2

Information Data Forum **Results**

Public Leaderboard **Private Leaderboard**

This competition has completed, this leaderboard reflects the preliminary final standings. The results will become final after the competition organizers verify the results.

* in the money

#	Δ1w	Team Name	Kdd Ctr Auc	Entries	Last Submission UTC (Best Submission - Last)
1	-	Catch up *	0.80893	141	Fri, 01 Jun 2012 23:57:26 (-0.1h)
2	↑34	Opera Solutions *	0.80824	90	Fri, 01 Jun 2012 23:59:54 (-0.2h)
3	new	Chinese Academy of Sciences *	0.80303	25	Fri, 01 Jun 2012 23:53:17 (-0.1h)
4	↑2	Steffen	0.80178	50	Fri, 01 Jun 2012 22:48:52
5	↓2	Birutas & LARCA & Team DL	0.80166	200	Fri, 01 Jun 2012 23:37:41 (-0.4h)

Conclusions

- Was a challenge to handle this HUGE dataset
- Collaborative filtering methods (for sparse data)
  - Pairwise-rank training
  - Unroll the data (150M -> 235M +/- samples)
- Feature engineering + supervised models
- Blending (mix models) is the key for accuracy
  - Pairwise rank SGD -> optimized the AUC
  - Neural network perform better than linear models

Thank you for the attention !