Deep Recommender Systems

Jill-Jênn Vie[1,3] Florian Yger[2] Ryan Lahfa[3] Hisashi Kashima[4]
Basile Clement[3] Kévin Cocchi[3] Thomas Chalumeau[3]

[1] Inria Lille

[2] Université Paris-Dauphine (France)

[3] Mangaki (Paris, France)

[4] RIKEN AIP, Tokyo & Kyoto University

LA RECOMMANDATION FRANCOPHONE D'ANIME ET MANGA

I – Collaborative filtering

Mangaki, recommendations of anime/manga

Rate anime/manga and receive recommendations

350,000 ratings by 2,000 users on 10,000 anime & manga

- myAnimeList
- AniDB
- AniList
- (soon) TVtropes

Build a profile

Steins;Gate

Suzumiya Haruhi no Shoushitsu

Suzumiya Haruhi no Yuuutsu

Terror in Resonance

The Night is Short, Walk On Girl

Time of Eve

5 centimètres par seconde

.hack//Liminality

.hack//Sign

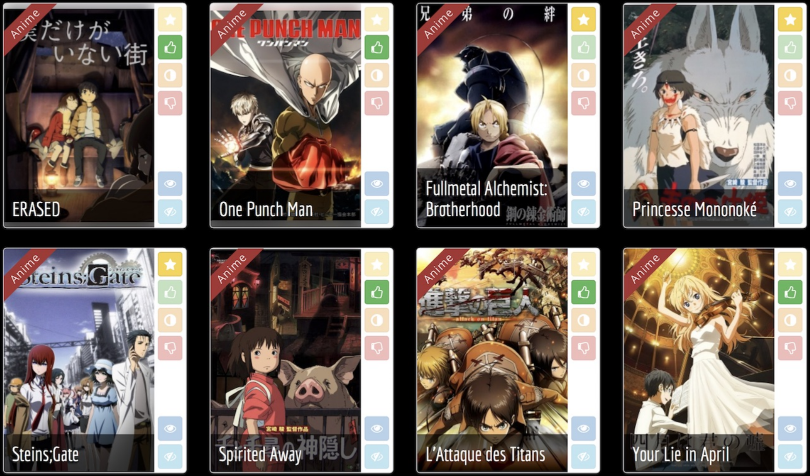
A Certain Scientific Railgun S

Mangaki prioritizes your watchlist

- Pokemon: Lucario and the Mystery of Mew
- Dimension W
- Haibane Renmei
- A Silent Voice
- Neon Genesis Evangelion
- Mind Game
- Record of Lodoss War
- Ghost in the Shell: Stand Alone Complex
- Neon Genesis Evangelion: The End of Evangelion
- Angel Beats!

Browse the rankings: top works

```
>>> from mangaki.models import Work
>>> Work.objects.filter(category__slug='anime').top()[:8]
```



Why nonprofit?

- Why should blockbusters get all the fun/clicks/money?
- Maybe there is one precious, unknown anime for you
  - and we can help you find it

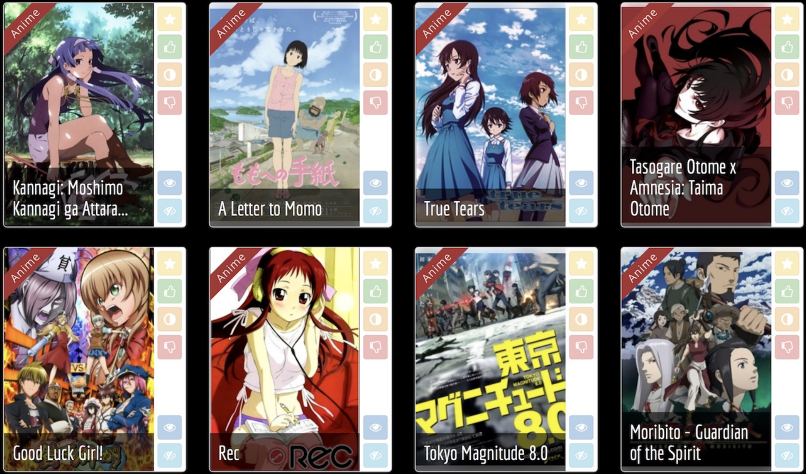
Driven by passion, not profit

- Everything is open source: github.com/mangaki
- Python (Django), Vue.js
- Many Jupyter notebooks (check 'em out!)

Awards: Microsoft Prize (2014) Japan Foundation (2016)

A simple idea: precious pearls

```
Work.objects.filter(category__slug='anime').pearls()[:8]
```



Recommender Systems

Problem

- Every user rates few items (1 %)
- How to infer missing ratings?

Example

Sacha	3	5	2	2
Ondine	4	1	4	5
Pierre	3	3	1	4
Joëlle	5	2	2	5

What is a machine learning algorithm?

Fit

Ondine	like	*Zootopia*
Ondine	favorite	*Porco Rosso*
Sacha	favorite	*Tokikake*
Sacha	dislike	*The Martian*

Predict

Ondine	?favorite	*The Martian*
Sacha	?like	*Zootopia*

What is a bad machine learning algorithm?

Fit

Ondine	like	*Zootopia*
Ondine	favorite	*Porco Rosso*
Sacha	favorite	*Tokikake*
Sacha	dislike	*The Martian*

100% correct

Predict

Ondine	dislike	*The Martian* (was: favorite)
Sacha	neutral	*Zootopia* (was: like)

20% correct

Cannot generalize

What is a **good** machine learning algorithm?

Fit

Ondine	favorite	*Zootopia* (was: like)
Ondine	favorite	*Porco Rosso*
Sacha	favorite	*Tokikake*
Sacha	dislike	*The Martian*

90% correct

Predict

Ondine	like	*The Martian* (was: favorite)
Sacha	favorite	*Zootopia* (was: like)

90% correct

How to compare algorithms?

dislike	wontsee	neutral	willsee	like	favorite
-2	-0.5	0.1	0.5	2	4

Penalty

If I predict: favorite for favorite $\to$ 0 error

dislike for favorite $\to (4-(-2))^2 = 36$ error

like for favorite $\to$ 4 error

Error: Mean value of (difference)^2

RMSE: square root of that

Divide / **Fit** / **Predict**

A likes 1		C likes 1		E ?neutral 3
B likes 2	B dislikes 3	C likes 2	D ?wontsee 3	C ?willsee 2
	B likes 4		D ?wontsee 4	

Matrix factorization → reduce dimension to generalize

Idea: Do user2vec for all users, item2vec for all movies
such that users like movies that are in their direction.

Fit

- R ratings, U user vectors, W work vectors.

$$R = UW^T \qquad \hat{r}_{ij}^{ALS} = U_i \cdot W_j$$

Predict: Will user i like item j?

- Just compute $U_i \cdot W_j$ and you will find out!

Algorithm ALS: Alternating Least Squares (Zhou, 2008)

- Until convergence ($\sim$ 20 iterations):
  - Fix U (users) learn W (works)
    in order to minimize the error (+ something)
  - Fix W find U

Illustration of ALS

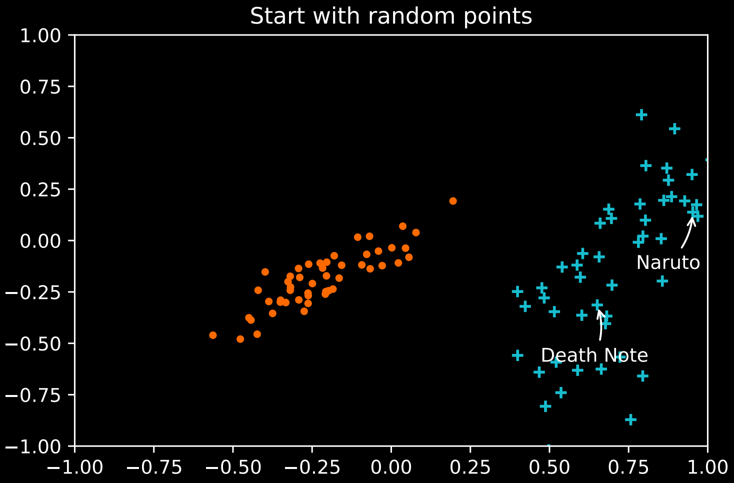
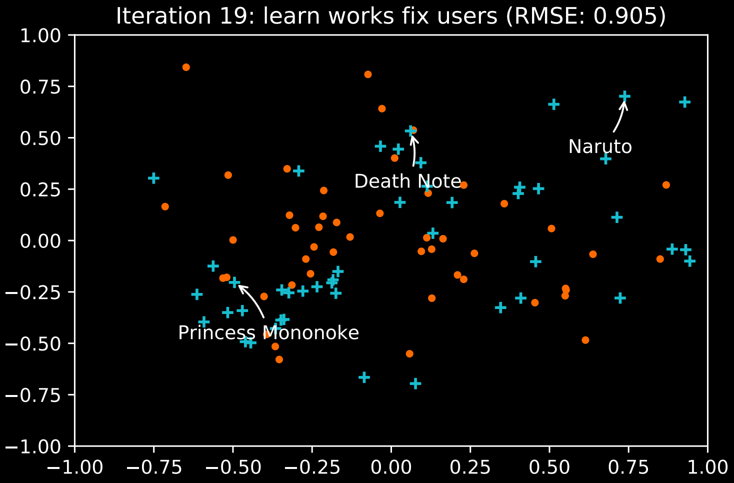


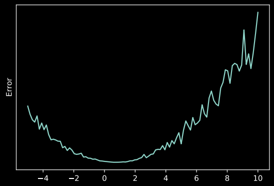
Illustration of ALS



Why + something? Regularize to generalize

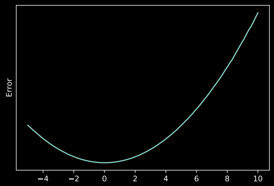
Just minimize RMSE

May not be optimal



Minimize RMSE + regularization:

$\Rightarrow$ easier to optimize



Visualizing all anime

What did we do, precisely?

Newton's method
To find the zeroes of $f : \mathbf{R} \to \mathbf{R}$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Optimization
What if we want to minimize $\mathcal{L} : \mathbf{R}^n \to \mathbf{R}$?

$$x_{n+1} = x_n - \underbrace{H\mathcal{L}(x_n)}_{n \times n \text{ matrix}}{}^{-1} \nabla \mathcal{L}(x_n)$$

What if it is costly?

$$x_{n+1} = x_n - \gamma \nabla \mathcal{L}(x_n)$$

Oh, we just invented gradient descent.

Alternating Least Squares

find U_k that minimizes

$$f(U_k) = \sum_{i,j} (\underbrace{U_i \cdot W_j}_{pred} - \underbrace{r_{ij}}_{real})^2 + \underbrace{\lambda ||U_i||_2^2 + \lambda ||W_j||_2^2}_{regularization}$$

(by the way: the derivative of $u \cdot v$ with respect to u is v)

find the zeroes of

$$f'(U_k) = \sum_{j \text{ rated by } k} 2(U_k \cdot W_j - r_{kj})W_j + 2\lambda U_k = 0$$

can be rewritten $AU_k = B$ so $U_k = A^{-1}B$ (easy!)

Complexity: $O(n^3)$ where n is the number of items rated by U_k

Stochastic Gradient Descent

$$U_k \leftarrow U_k - \gamma f'(U_k)$$

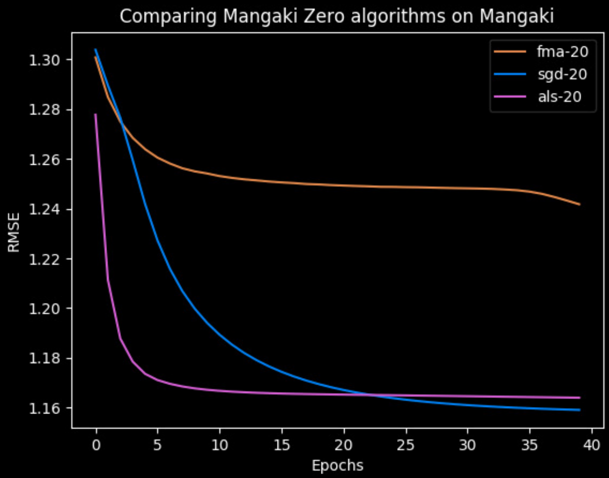
$$U_k \leftarrow (1 - 2\gamma\lambda) U_k - 2\gamma \sum_{j \text{ rated by } k} \underbrace{(U_k \cdot W_j - r_{kj})}_{\text{prediction error}} W_j$$

U_k is updated according to its neighbors W_j

Benchmarks

ALS: minimizing U then W then U then W

SGD: minimizing U and W at the same time



Drawback with collaborative filtering

Issue: Item Cold-Start

- If no ratings are available for a work j
  $\Rightarrow$ Its vector W_j cannot be learned :-(

No way to distinguish between unrated works.

But we have (many) posters!

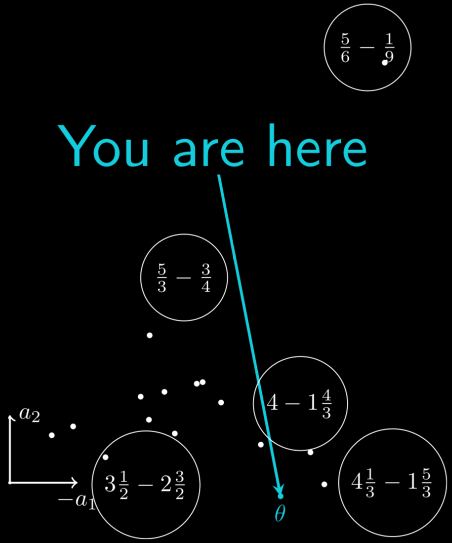
II – Factorization Machines

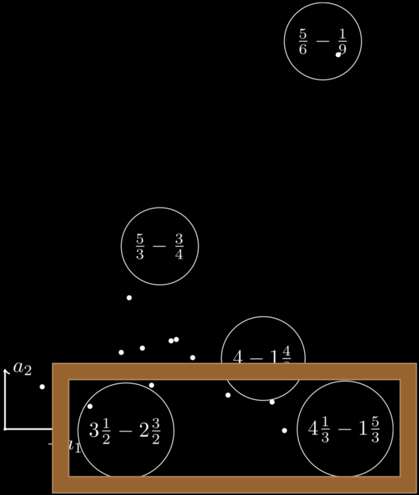
Learning multidimensional feature embeddings

Logistic Regression
Learn a bias for each feature (each user, item, etc.)

Factorization Machines
Learn a bias and an embedding for each feature

What can be done with multidimensional embeddings?

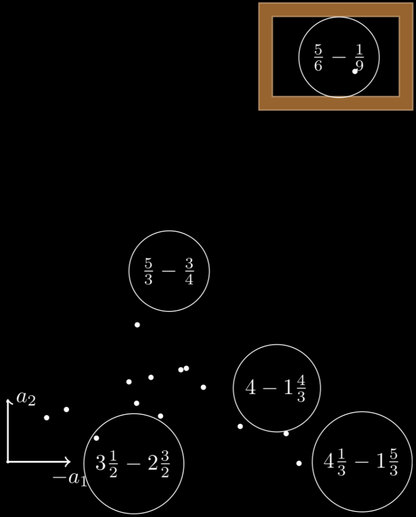




Items that discriminate only over one dimension

$3\frac{1}{2} - 2\frac{3}{2}$	$4\frac{1}{3} - 2\frac{4}{3}$	$4\frac{1}{3} - 1\frac{5}{3}$
$b = 0.13$	$b = -0.46$	$b = -1.99$
$-a_1 = 2.01$	$-a_1 = 4.65$	$-a_1 = 5.66$
$a_2 = -0.03$	$a_2 = -0.02$	$a_2 = 0.00$

Interpreting the components



Items that highly discriminate over both dimensions

$\frac{3}{4} - \frac{3}{8}$	$\frac{5}{6} - \frac{1}{9}$
$b = 1.09$	$b = -0.28$
$-a_1 = 5.54$	$-a_1 = 5.29$
$a_2 = 6.22$	$a_2 = 6.44$

How to model pairwise interactions with side information?

If you know user i watched item j on TV (not theatre)
How to model it?

y: rating of user i over item j

Biases

$$y = \theta_i + e_j$$

Collaborative filtering

$$y = \theta_i + e_j + \langle \boldsymbol{v}_{\text{user } \boldsymbol{i}}, \boldsymbol{v}_{\text{item } \boldsymbol{j}} \rangle$$

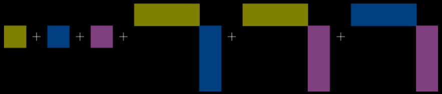
With side information

$$y = \theta_i + e_j + w_{\text{TV}} + \langle \boldsymbol{v}_{\text{user } \boldsymbol{i}}, \boldsymbol{v}_{\text{item } \boldsymbol{j}} \rangle + \langle \boldsymbol{v}_{\text{user } \boldsymbol{i}}, \boldsymbol{v}_{\text{TV}} \rangle + \langle \boldsymbol{v}_{\text{item } \boldsymbol{j}}, \boldsymbol{v}_{\text{TV}} \rangle$$

Factorization Machines

Just pick features (ex. user, item, skill) and you get a model

Each feature k is modeled by bias w_k and embedding $\boldsymbol{v_k}$.



$$\operatorname{logit} p(\boldsymbol{x}) = \mu + \underbrace{\sum_{k=1}^{N} w_k x_k}_{\text{logistic regression}} + \underbrace{\sum_{1 \leq k < l \leq N} x_k x_l \langle \boldsymbol{v_k}, \boldsymbol{v_l} \rangle}_{\text{pairwise relationships}}$$

Jill-Jênn Vie and Hisashi Kashima. "Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing". In: *33th AAAI Conference on Artificial Intelligence*. 2019. URL: http://arxiv.org/abs/1811.03388

Regression with sparse features (very elegant!)

$\boldsymbol{x}$ concatenation of one-hot vectors (ex. at positions s and t)

$\langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_i w_i x_i = w_s + w_t$

$||V\boldsymbol{x}||^2 = \sum_{i,j} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \geq 0$

$\frac{1}{2}(||V\boldsymbol{x}||^2 - \mathbf{1}^T(V \circ V)(\boldsymbol{x} \circ \boldsymbol{x})) = \sum_{i<j} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = \langle \boldsymbol{v}_s, \boldsymbol{v}_t \rangle$

Factorization machines (Rendle 2012)

$P(\langle \boldsymbol{x}, \boldsymbol{v}_i \rangle)$ for a polynomial P

The Blondel Trilogy

- Polynomial networks and FMs (ICML 2016)
- Multi-output polynomial networks and FMs (NIPS 2017)
- Higher-order FMs (NIPS 2016)

III – Binary factorization

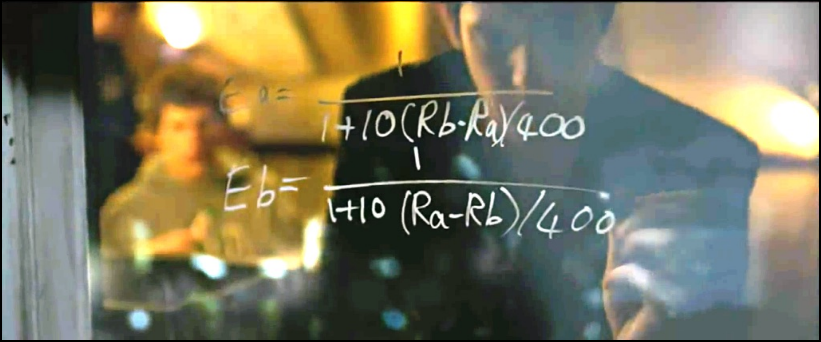
Chess players have Elo ratings

Elo ratings are updated after each match

If player 1 (550) beats player 2 (600)

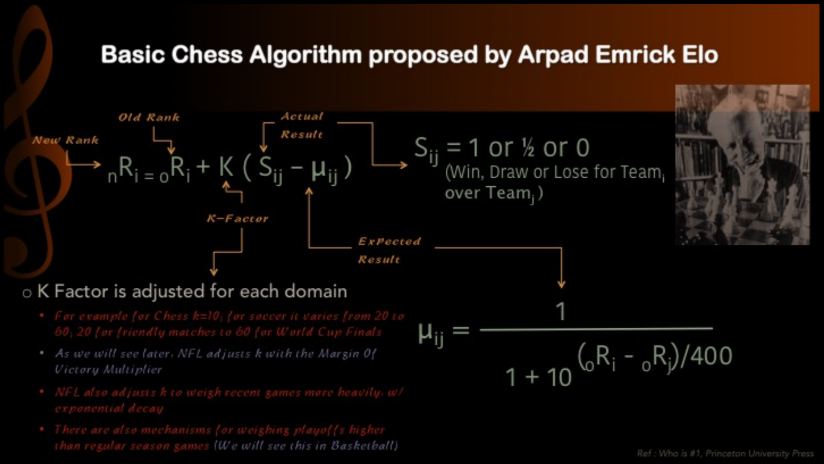
Then player 1 will ↑ (560) and player 2 will ↓ (590)

Let's ask Harvard students



(*The Social Network*)

K-Factor???



(Not *The Social Network*)

Old models still used today

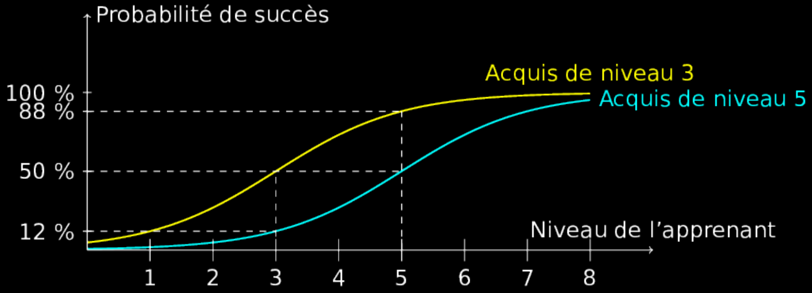
Elo (1960–1978)

$$P(\theta_i \text{ beats } \theta_j) = \frac{1}{1 + 10^{(\theta_j - \theta_i)/400}}$$

Item response theory (1960)

$$P(\theta_i \text{ solves } d_j) = \frac{1}{1 + e^{-(\theta_i - d_j)}}$$

Examples



Used in PISA, GMAT, Pix.

Maximum likelihood estimation

Given outcomes $r \in \{0, 1\}$, how to estimate θ?

$p = \frac{1}{1+e^{-(\theta-d)}} = \sigma(\theta - d)$

Thanks to logistic function: $p' = p(1 - p)$

$L(\theta) = \log p^r(1-p)^{1-r} = r \log p + (1-r) \log(1-p)$

$\nabla_\theta L = \frac{\partial L}{\partial \theta} = r - p$

$\theta_{t+1} = \theta_t + \gamma \underbrace{\nabla_\theta L}_{r-p}$

Thus it is online gradient ascent! K-factor $= \gamma =$ learning rate.

> *The chess statistician Jeff Sonas believes that the original $K = 10$ value (for players rated above 2400) is inaccurate in Elo's work.*

Evolving over time

Players ability increase as they win matches over other players

So players may have an optimistic strategy to plan their matches

Factorization: learning vectors

From some R_{ij} infer other R_{ij}

Collaborative filtering
Learn model U, V such that $R \simeq UV \quad \widehat{r_{ij}} = \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle$

Optimize regularized least squares

$\sum_{i,j}(\widehat{r_{ij}} - r_{ij})^2 + \lambda(||U||_F^2 + ||V||_F^2)$

Binary version
Learn model U, V such that $R \simeq \sigma(UV) \quad \widehat{r_{ij}} = \sigma(\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle)$

Optimize likelihood

EM algorithm via MCMC: sample U, optimize V (Cai, 2010)

Slow, $d \leq 6$

Scaling to big data

Gradient descent
For each example update parameters

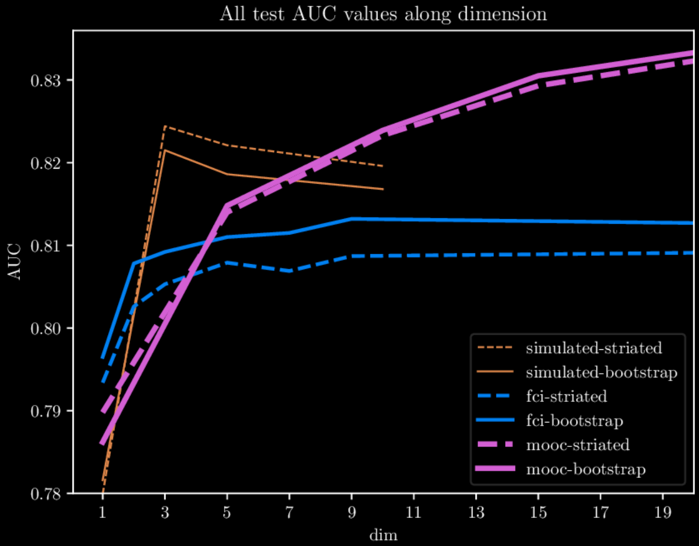
Batch gradient descent
Compute the gradient on all examples and update parameters

Stochastic gradient descent
Sample examples and update parameters

Minibatch gradient descent
Sample a minibatch of examples and update parameters

Scaling to high dimension

$\theta_{t+1} = \theta_t - \gamma \nabla_\theta \mathcal{L} \Rightarrow$ Replace $\nabla_\theta \mathcal{L}$ with an unbiased estimate $\tilde{\nabla}_\theta \mathcal{L}$



IV – Deep Factorization

Drawback with collaborative filtering

Issue: Item Cold-Start

- If no ratings are available for a work j

  $\Rightarrow$ Its vector W_j cannot be learned :-(

No way to distinguish between unrated works.

But we have (many) posters!

Illustration2Vec (Saito and Matsui, 2015)

Prediction results

#	General Tag	Confidence
1.	1girl	86.1%
2.	thighhighs	84.0%
3.	solo	79.2%
4.	red hair	73.1%
5.	long hair	66.4%
6.	breasts	53.7%
7.	gloves	38.0%
8.	weapon	34.0%
9.	elbow gloves	28.3%
10.	high heels	14.0%
11.	tattoo	10.9%
#	**Character Tag**	
#	**Copyright Tag**	
#	**Rating**	
1.	safe	68.4%
2.	questionable	29.3%
3.	explicit	1.92%

- CNN (VGG-16) pretrained on ImageNet (photos)
- Retrained on Danbooru (1.5M manga illustrations with tags)
- 502 most frequent tags kept, outputs tag weights

LASSO for sparse linear regression

T matrix of 15000 works $\times$ 502 tags (T_j: tags of work j)

Fit

- Each user is described by its preferences over tags P_i
- LASSO constraint: user likes/hates few tags
- Learn user preferences P_i such that

$$\hat{r}_{ij}^{LASSO} = P_i \cdot T_j.$$

Predict: Will user i like work j?

- Here is a new work with a poster and tags T_j
- Just compute $P_i \cdot T_j$ and you will find out!

Interpretation and explanation of user preferences

- *You seem to like magical girls but not blonde hair*

  $\Rightarrow$ *Look! All of them are brown hair! Buy now!*

Combine models

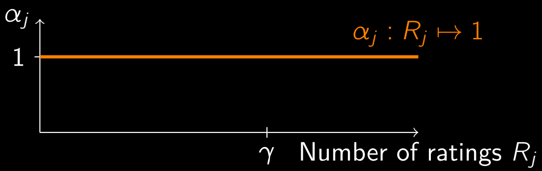
Which model should we choose between ALS and LASSO?

Answer Both!

Methods boosting, bagging, model stacking, blending.

Idea find α_j s.t. $\hat{r}_{ij} \triangleq \alpha_j \hat{r}_{ij}^{ALS} + (1 - \alpha_j)\hat{r}_{ij}^{LASSO}$.
If popular, listen to ALS more than LASSO

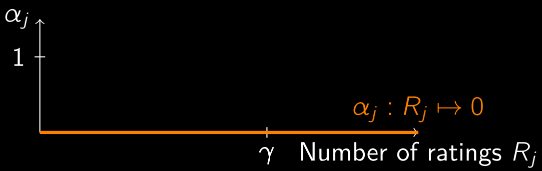
Examples of α_j



Mimics ALS

$$\hat{r}_{ij} \triangleq 1\hat{r}_{ij}^{ALS} + 0\hat{r}_{ij}^{LASSO}.$$

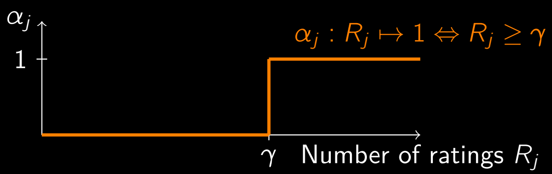
Examples of α_j



Mimics LASSO

$$\hat{r}_{ij} \triangleq 0\hat{r}_{ij}^{ALS} + 1\hat{r}_{ij}^{LASSO}.$$

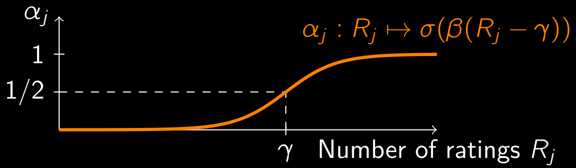
Examples of α_j



$$\hat{r}_{ij}^{BALSE} = \begin{cases} \hat{r}_{ij}^{ALS} & \text{if item } j \text{ was rated at least } \gamma \text{ times} \\ \hat{r}_{ij}^{LASSO} & \text{otherwise} \end{cases}$$

But we can't: Not differentiable!

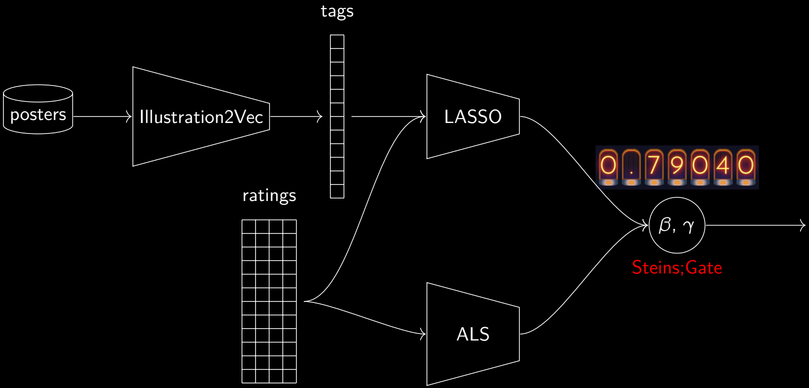
Examples of α_j



$$\hat{r}_{ij}^{BALSE} = \sigma(\beta(R_j - \gamma))\hat{r}_{ij}^{ALS} + (1 - \sigma(\beta(R_j - \gamma)))\,\hat{r}_{ij}^{LASSO}$$

β and γ are learned by stochastic gradient descent.

Blended Alternate Least Squares with Explanation (BALSE)



Comparing algorithms: cross-validation

- 80% of the ratings are used for training
- 20% of the ratings are kept for testing

Different sets of items:

- Whole test set of works
- 1000 works least rated (1.5%)
- Cold-start: works not seen in the training set (only the posters)

Results

RMSE	Test set	1000 least rated (1.5%)	Cold-start items
ALS	1.157	1.299	1.493
LASSO	1.446	1.347	1.358
BALSE	**1.150**	**1.247**	**1.316**

Summing up

We presented BALSE, a model that:

- uses information in the ratings (collaborative filtering)
- uses information in the posters using CNNs (content-based)
- combine them in a nonlinear way

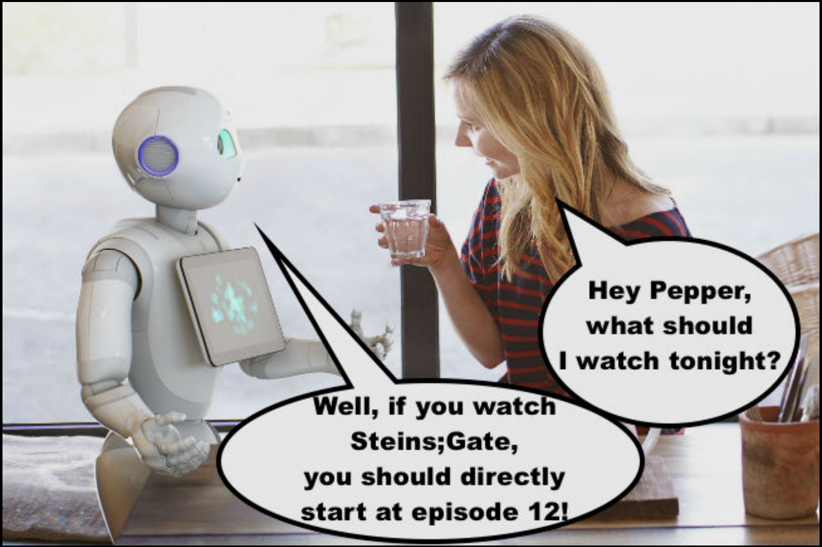
to improve the recommendations, and explain them.

Future work: Make your neural network watch the anime

Extract frames from episodes

Cowboy Bebop EP 23 “Brain Scratch”, Sunrise

Coming soon: Watching assistant



Deep Factorization Machines

Learn layers $W^{(\ell)}$ and $b^{(\ell)}$ such that:

$$
\begin{aligned}
\boldsymbol{a}^0(\boldsymbol{x}) &= (\boldsymbol{v}_{\texttt{user}}, \boldsymbol{v}_{\texttt{item}}, \boldsymbol{v}_{\texttt{skill}}, \ldots) \\
\boldsymbol{a}^{(\ell+1)}(\boldsymbol{x}) &= \mathrm{ReLU}(W^{(\ell)} \boldsymbol{a}^{(\ell)}(\boldsymbol{x}) + \boldsymbol{b}^{(\ell)}) \quad \ell = 0, \ldots, L-1 \\
y_{DNN}(\boldsymbol{x}) &= \mathrm{ReLU}(W^{(L)} \boldsymbol{a}^{(L)}(\boldsymbol{x}) + \boldsymbol{b}^{(L)})
\end{aligned}
$$

$$
\mathrm{logit}\, p(\boldsymbol{x}) = y_{FM}(\boldsymbol{x}) + y_{DNN}(\boldsymbol{x})
$$

Jill-Jênn Vie. "Deep Factorization Machines for Knowledge Tracing". In: *The 13th Workshop on Innovative Use of NLP for Building Educational Applications*. 2018. URL: https://arxiv.org/abs/1805.00356

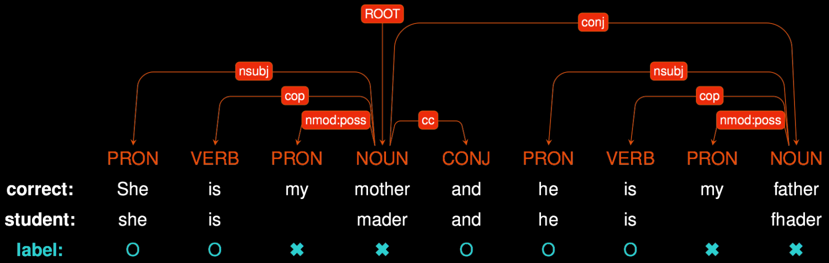
Comparison

- FM: y_{FM} factorization machine with $\lambda = 0.01$
- Deep: y_{DNN}: multilayer perceptron
- DeepFM: $y_{DNN} + y_{FM}$ with shared embedding
- Bayesian FM: $w_k, v_{kf} \sim \mathcal{N}(\mu_f, 1/\lambda_f)$
  $\mu_f \sim \mathcal{N}(0, 1)$, $\lambda_f \sim \Gamma(1, 1)$ (trained using Gibbs sampling)

Various types of side information

- first: `<discrete>` (`user`, `token`, `countries`, etc.)
- last: `<discrete>` + `<continuous>` (`time` + `days`)
- pfa: `<discrete>` + `wins` + `fails`

Duolingo dataset




```
# user:D2inSf5+  countries:MX  days:1.793  client:web  session:lesson  format:reverse_translate  time:16
8rgJEAPw1001  She       PRON   Case=Nom|Gender=Fem|Number=Sing|Person=3|PronType=Prs|fPOS=PRON++PRP    nsubj      4  0
8rgJEAPw1002  is        VERB   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBZ    cop        4  0
8rgJEAPw1003  my        PRON   Number=Sing|Person=1|Poss=Yes|PronType=Prs|fPOS=PRON++PRP$              nmod:poss  4  1
8rgJEAPw1004  mother    NOUN   Degree=Pos|fPOS=ADJ++JJ                                                 ROOT       0  1
8rgJEAPw1005  and       CONJ   fPOS=CONJ++CC                                                           cc         4  0
8rgJEAPw1006  he        PRON   Case=Nom|Gender=Masc|Number=Sing|Person=3|PronType=Prs|fPOS=PRON++PRP   nsubj      9  0
8rgJEAPw1007  is        VERB   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBZ    cop        9  0
8rgJEAPw1008  my        PRON   Number=Sing|Person=1|Poss=Yes|PronType=Prs|fPOS=PRON++PRP$              nmod:poss  9  1
8rgJEAPw1009  father    NOUN   Number=Sing|fPOS=NOUN++NN                                               conj       4  1

# user:D2inSf5+  countries:MX  days:2.689  client:web  session:practice  format:reverse_translate  time:6
oMGsnnH/0101  When      ADV    PronType=Int|fPOS=ADV++WRB                                              advmod     4  1
oMGsnnH/0102  can       AUX    VerbForm=Fin|fPOS=AUX++MD                                               aux        4  0
oMGsnnH/0103  I         PRON   Case=Nom|Number=Sing|Person=1|PronType=Prs|fPOS=PRON++PRP               nsubj      4  1
oMGsnnH/0104  help      VERB   VerbForm=Inf|fPOS=VERB++VB                                              ROOT       0  0
```

Results

Model	d	epoch	train	first	last	pfa
Bayesian FM	20	500/500	–	0.822	–	–
Bayesian FM	20	500/500	–	–	0.817	–
DeepFM	20	15/1000	0.872	0.814	–	–
Bayesian FM	20	100/100	–	–	0.813	–
FM	20	20/1000	0.874	0.811	–	–
Bayesian FM	20	500/500	–	–	–	0.806
FM	20	21/1000	0.884	–	–	0.805
FM	20	37/1000	0.885	–	0.8	–
DeepFM	20	77/1000	0.89	–	0.792	–
Deep	20	7/1000	0.826	0.791	–	–
Deep	20	321/1000	0.826	–	0.79	–
LR	0	50/50	–	–	–	0.789
LR	0	50/50	–	0.783	–	–
LR	0	50/50	–	–	0.783	–

Duolingo ranking

Rank	Team	Algo	AUC
1	SanaLabs	RNN + GBDT	.857
2	singsound	RNN	.854
2	NYU	GBDT	.854
4	CECL	LR + L1 (13M feat.)	.843
5	TMU	RNN	.839
7 (off)	JJV	Bayesian FM	.822
8 (off)	JJV	DeepFM	.814
10	JJV	DeepFM	.809
15	Duolingo	LR	.771

Burr Settles et al. "Second language acquisition modeling". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. 2018, pp. 56–65. URL: http://sharedtask.duolingo.com

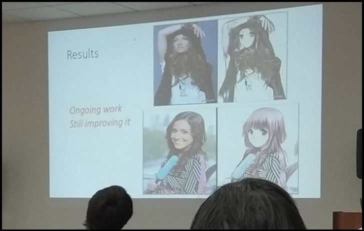
 jill-jenn.vie@inria.fr

Try our recommender system: mangaki.fr

Any questions?

Know more

- AI for Manga & Anime: research.mangaki.fr



- Burr Settles et al. "Second language acquisition modeling". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. 2018, pp. 56–65. URL: http://sharedtask.duolingo.com.
- Jill-Jênn Vie. "Deep Factorization Machines for Knowledge Tracing". In: *The 13th Workshop on Innovative Use of NLP for Building Educational Applications*. 2018. URL: https://arxiv.org/abs/1805.00356.
- Jill-Jênn Vie and Hisashi Kashima. "Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing". In: *33th AAAI Conference on Artificial Intelligence*. 2019. URL: http://arxiv.org/abs/1811.03388.