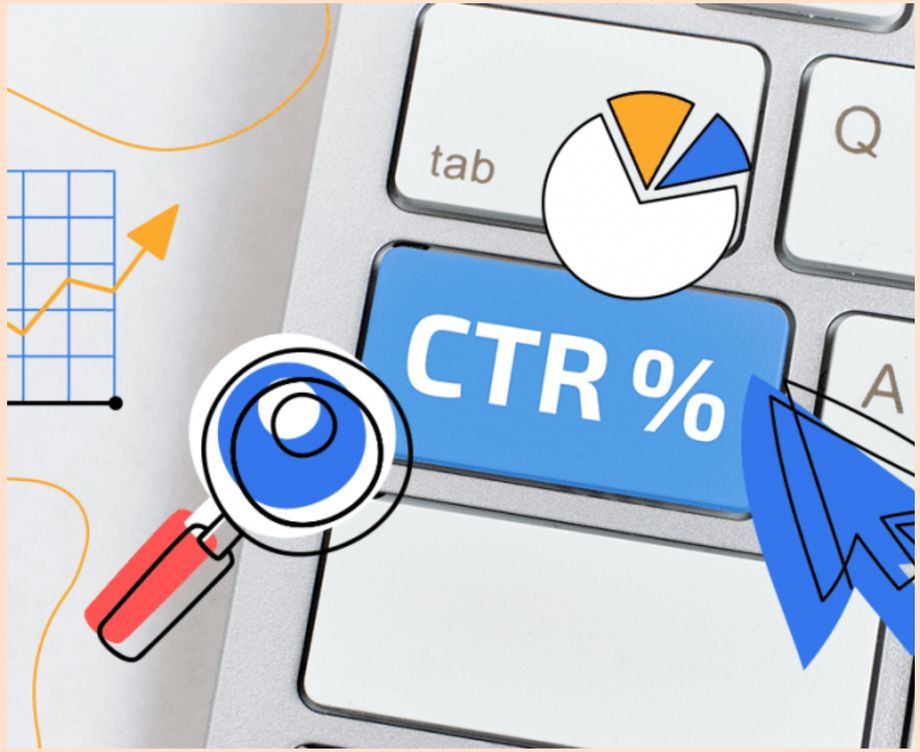
FIELD-AWARE FACTORIZATION MACHINES IN A REAL-WORLD ONLINE ADVERTISING SYSTEM



Elise Chin Mathilde Da Cruz Vincent Duchauffour

INTRODUCTION : PRÉSENTATION DE L'ARTICLE

1

a. Les auteurs
b. Définitions
c. Contributions
d. Contexte

LES AUTEURS

Papier publié en 2017 et présenté à la World Wide Web Conférence (IW3C2)

Yuchin Juan
Criteo Research

Olivier Chapelle
Google

Damien Lefortier
Facebook

DÉFINITIONS

- Computational advertising (CA)

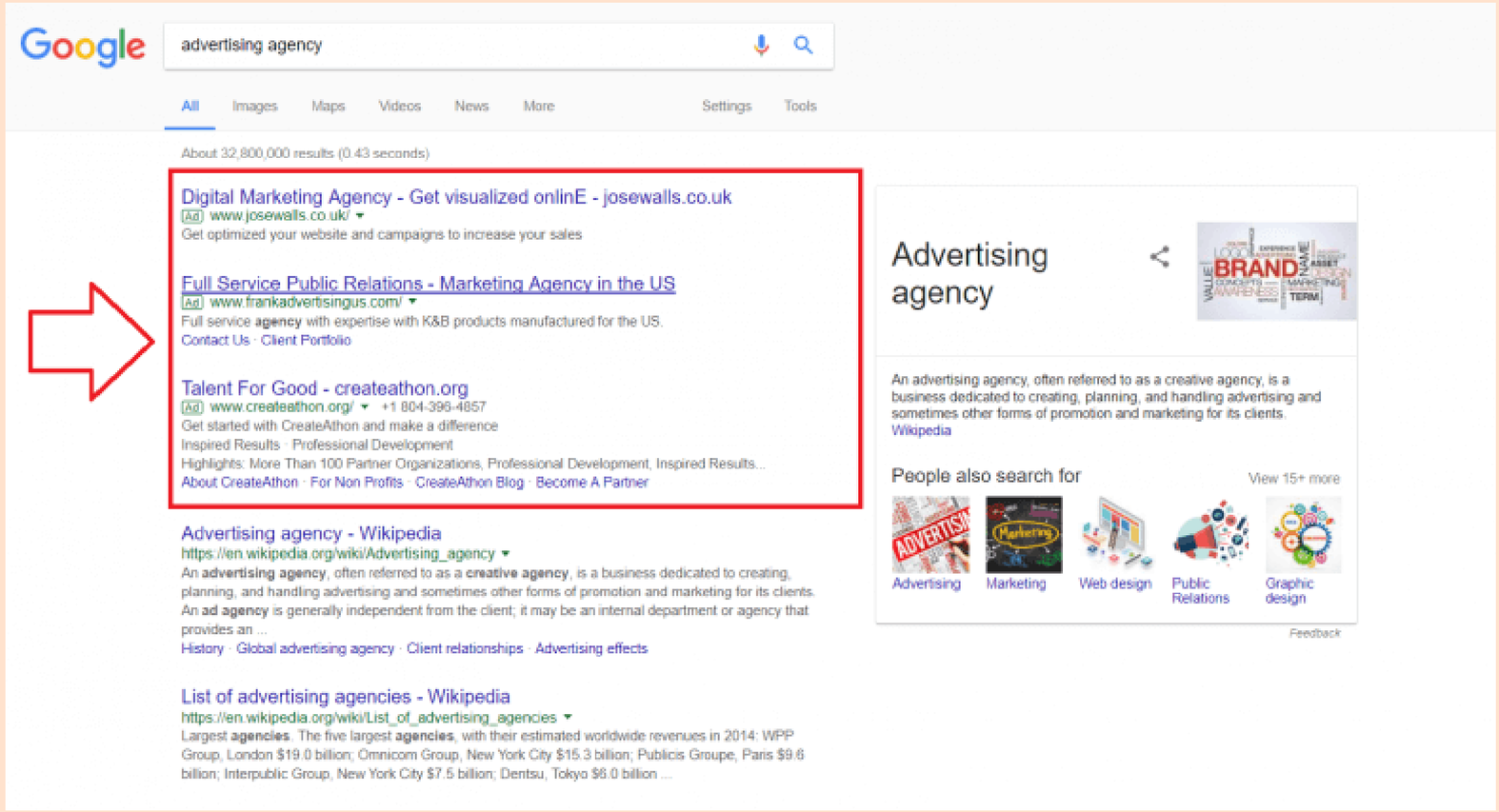
➡ Publicité la plus pertinente selon un contexte

➡ information retrieval, statistical modeling, machine learning, optimization large scale search, text analysis

  - Sponsored search
  - Contextual advertising
  - Display advertising

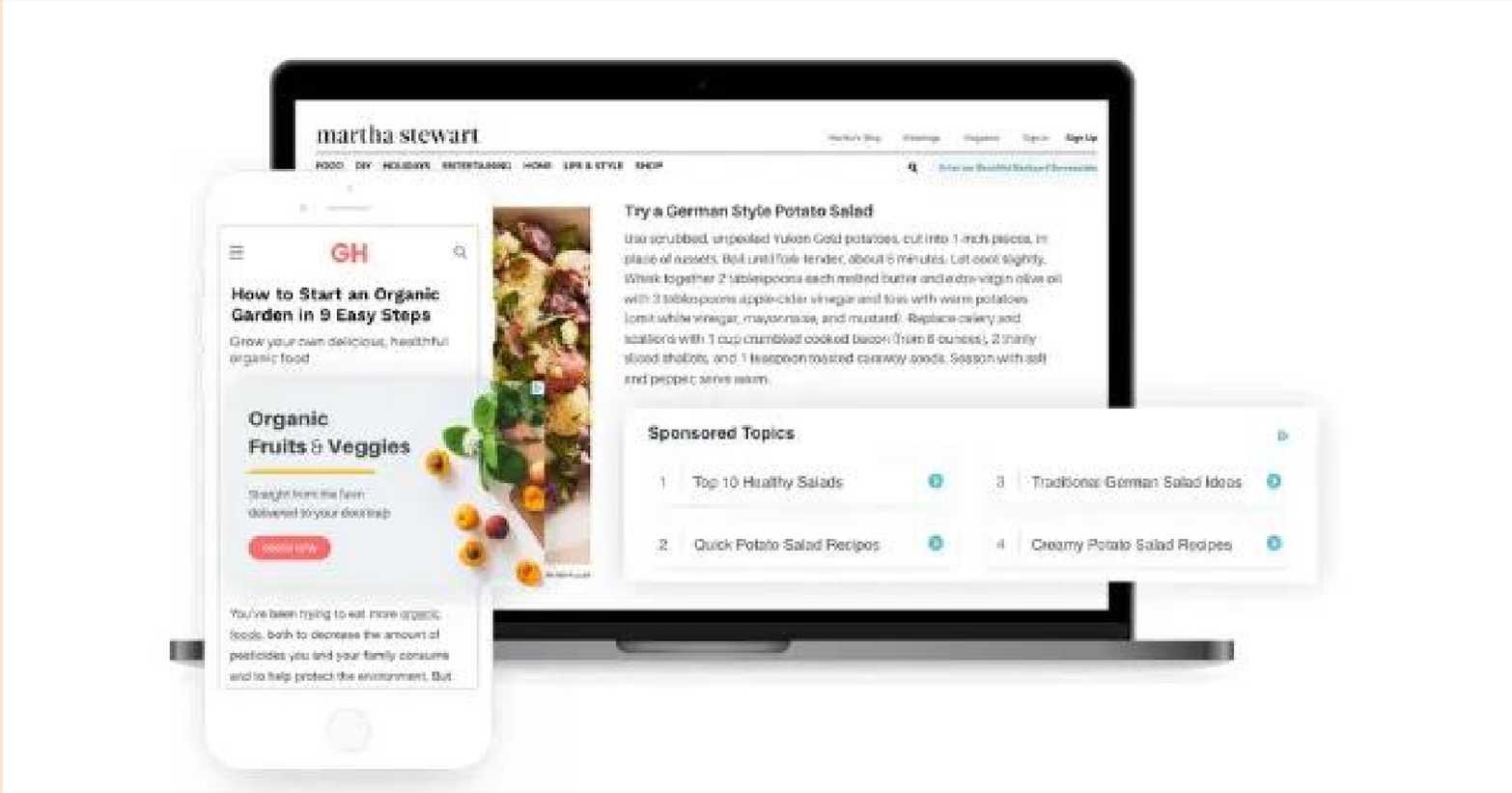
DÉFINITIONS

Sponsored search



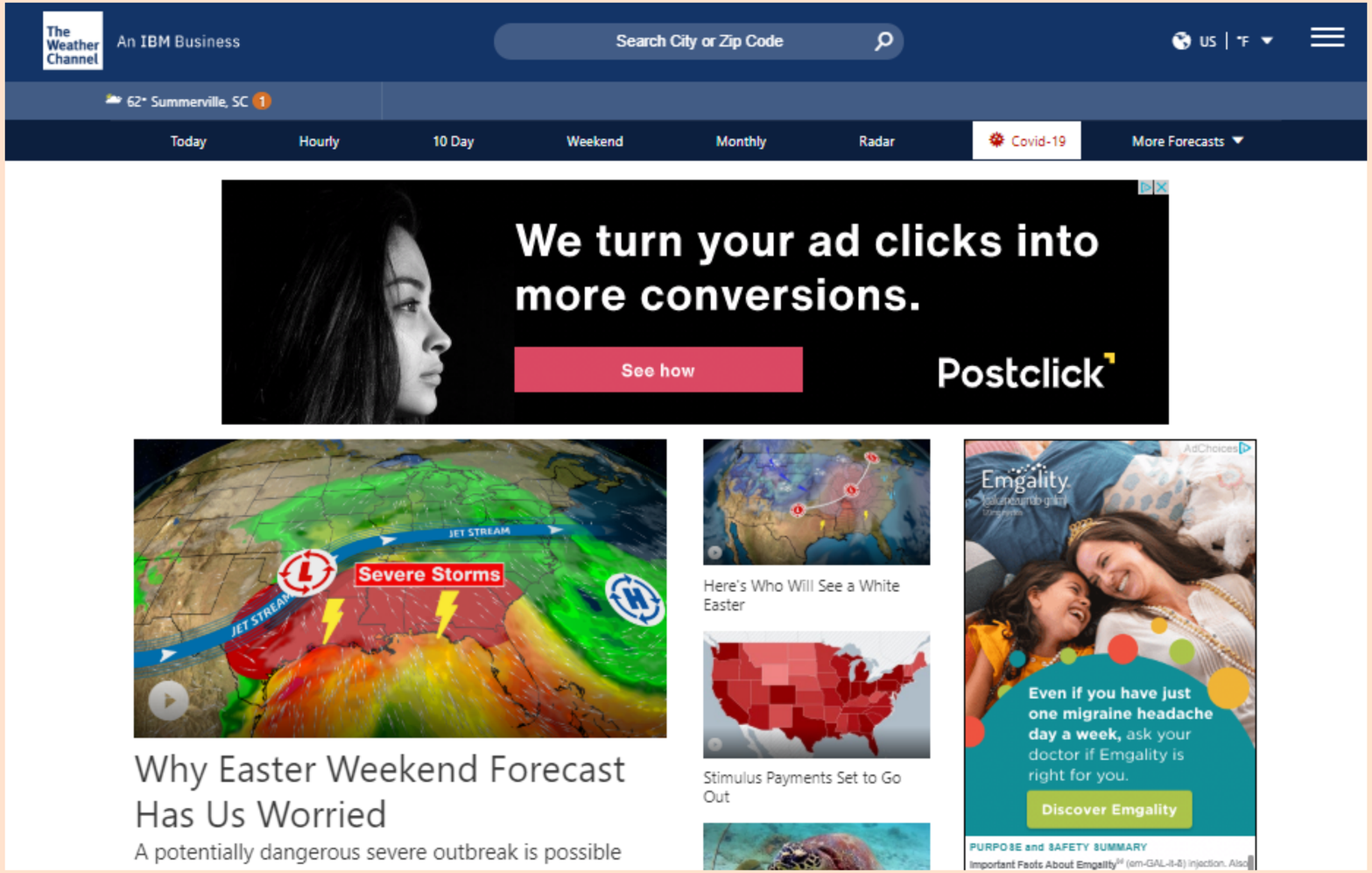
DÉFINITIONS

Contextual Advertising



DÉFINITIONS

Display Advertising



DÉFINITIONS

- Computational advertising (CA)

➡ Publicité la plus pertinente selon un contexte

➡ information retrieval, statistical modeling, machine learning, optimization large scale search, text analysis

  - Sponsored search
  - Contextual advertising
  - Display advertising

- Click-through / Click-through Rate (CTR)

➡ nombre de clics / nombre de vues

DÉFINITIONS

- Computational advertising (CA)

➡ Publicité la plus pertinente selon un contexte

➡ information retrieval, statistical modeling, machine learning, optimization large scale search, text analysis

  - Sponsored search
  - Contextual advertising
  - Display advertising

- Click-through / Click-through Rate (CTR)

➡ nombre de clics / nombre de vues

- Conversion / Conversion rate

➡ Action spécifique de l'utilisateur

CONTRIBUTIONS

Utiliser le FFM dans un système de production pour prédire le CTR et CR.

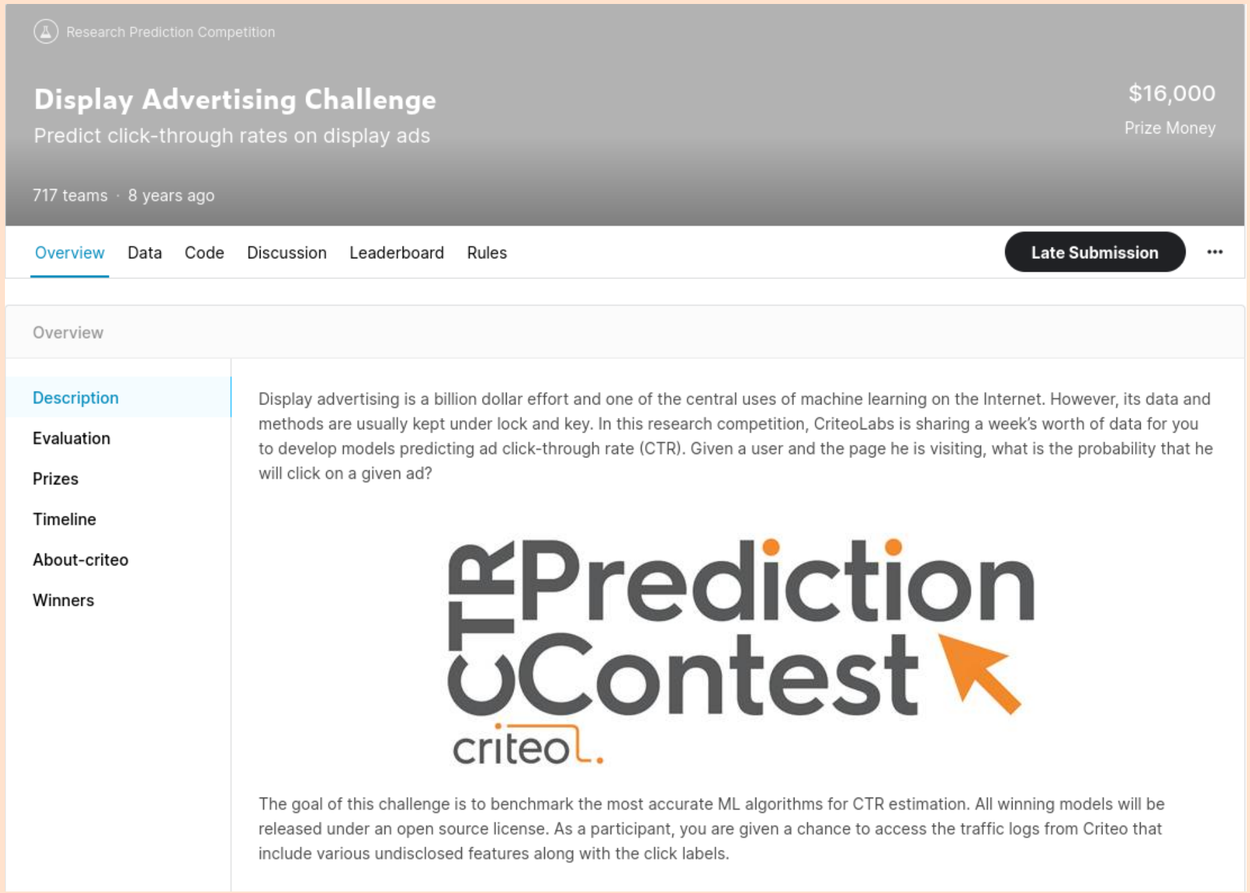
Cette méthode est-elle aussi efficace dans le monde réel que dans des compétitions kaggle ?

1. Application de FFM dans un cadre industriel

2. Entrainement distribué

3. Pré mature Warm Start

CONTEXTE



Kaggle compétition : Predict ad CTR

"Given a user and the page he is visiting, what is the probability that he will click on a given ad ?"

7 jours de données : prédire le jour suivant

1. Field-aware Factorization Machines

2/3 ? Logistic Regression with cross-features

CONTEXTE

Système de production = Contraintes et objectifs spécifiques ! Différent d'une compétition académique.

Cas du Netflix Prize (2006)

➡ $1 million à l'équipe qui améliorerait l'accuracy de leur système de recommandation d'au moins 10%

➡ Codes gagnants jamais utilisés par Netflix

✖ Code écrit pour 100M notes (VS +5Milliards en réalité)

✖ Code ne s'adapte pas à de nouvelles notes

✖ Trop de "Engineering efforts" pour le gain

✖ Shift des recommandations (transition DVD streaming)

= données obsolètes

CONTEXTE

SOTA Kaggle challenge = SOTA real world ?

Différences entre la compétition et le monde réel :

- Prédire si clic ou non
- Données limitées
- Tout prédire avant la fin de la compétition
- etc.

- Prédire Click-Through Rate ET Conversion Rate
- Données continues
- Prédire chaque valeur au bon moment = TEMPS LIMITÉ !
- etc.

MODÈLES

2

a. Linéaire

b. Polynomial de degré 2 (Poly2) et Factorization Machine (FM)

d. Field-aware Factorization Machine (FFM) : SOTA

DATASET

$$\{\mathbf{x_i}, y_i\}_{i=1}^{m}$$

y_i label

$\mathbf{x_i}$ vecteur de features de dimension n

Terminologie :

- **Champ** : {Publisher, Advertiser}
- **Feature** : {ESPN, Vogue, NBC, Nike, ...}

		Publisher	Advertiser
+80	−20	ESPN	Nike
+10	−90	ESPN	Gucci
+0	−1	ESPN	Adidas
+15	−85	Vogue	Nike
+90	−10	Vogue	Gucci
+10	−90	Vogue	Adidas
+85	−15	NBC	Nike
+0	−0	NBC	Gucci
+90	−10	NBC	Adidas

Table 1: An artificial CTR data set, where + (−) represents the number of clicked (unclicked) impressions.

LINÉAIRE

Formulation :

$$\phi_{LM}(\mathbf{w}, \mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{j \in C_1} w_j x_j$$

où C1 est l'ensemble des éléments non nuls dans x

		Publisher	Advertiser
+80	−20	ESPN	Nike
+10	−90	ESPN	Gucci
+0	−1	ESPN	Adidas
+15	−85	Vogue	Nike
+90	−10	Vogue	Gucci
+10	−90	Vogue	Adidas
+85	−15	NBC	Nike
+0	−0	NBC	Gucci
+90	−10	NBC	Adidas

Table 1: An artificial CTR data set, where + (−) represents the number of clicked (unclicked) impressions.

Exemple :

$$\phi_{LM}(\mathbf{w}, \mathbf{x}) = w_{ESPN} x_{ESPN} + w_{Vogue} x_{Vogue} + w_{Gucci} x_{Gucci} + \dots$$

Limite : ne permet pas de représenter l'effet d'une feature sur une autre (e.g. taux de clics plus élevé pour (Vogue, Gucci) que (ESPN, Gucci))

POLY2

Formulation :

$$\phi_{Poly2}(\mathbf{w}, \mathbf{x}) = \sum_{j_1, j_2 \in C_2} w_{j_1, j_2} x_{j_1} x_{j_2}$$

où C2 est l'ensemble des couples d'éléments non nuls dans **x**

FACTORIZATION MACHINE (FM)

Formulation :

$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = \sum_{j_1, j_2 \in C_2} < \mathbf{w}_{j_1}, \mathbf{w}_{j_2} > x_{j_1} x_{j_2}$$

où :

- w_j1 et w_j2 sont deux vecteurs de dimension k, k étant un hyperparamètre
- C2 est l'ensemble des couples d'éléments non nuls dans **x**

		Publisher	Advertiser
+80	−20	ESPN	Nike
+10	−90	ESPN	Gucci
+0	−1	ESPN	Adidas
+15	−85	Vogue	Nike
+90	−10	Vogue	Gucci
+10	−90	Vogue	Adidas
+85	−15	NBC	Nike
+0	−0	NBC	Gucci
+90	−10	NBC	Adidas

Table 1: An artificial CTR data set, where + (−) represents the number of clicked (unclicked) impressions.

Exemple :

$$\phi_{Poly2}(\mathbf{w}, \mathbf{x}) = w_{ESPN,Gucci} \times x_{ESPN} x_{Gucci} + w_{Vogue,Gucci} \times x_{Vogue} x_{Gucci} + \ldots$$

$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = < \mathbf{w}_{ESPN}, \mathbf{w}_{Gucci} > \times x_{ESPN} x_{Gucci} + < \mathbf{w}_{Vogue}, \mathbf{w}_{Gucci} > \times x_{Vogue} x_{Gucci} +$$

FIELD-AWARE FACTORIZATION MACHINE (FFM)

Formulation :

$$\phi_{FFM}(\mathbf{w}, \mathbf{x}) = \sum_{j_1, j_2 \in C_2} < \mathbf{w}_{j_1, f_2}, \mathbf{w}_{j_2, f_1} > x_{j_1} x_{j_2}$$

où :

- f1 et f2 sont respectivement les champs des features j1 et j2
- w_j1, f2 et w_j2, f1 sont deux vecteurs pour les features j1 et j2 dans le contexte des champs f2 et f1 respectivement
- C2 est l'ensemble des couples d'éléments non nuls dans **x**

Exemple :

Clicked	Publisher (P)	Advertiser (A)	Gender (G)
Yes	ESPN	Nike	Male

FM $\quad \boldsymbol{w}_{\text{ESPN}} \cdot \boldsymbol{w}_{\text{Nike}} + \boldsymbol{w}_{\text{ESPN}} \cdot \boldsymbol{w}_{\text{Male}} + \boldsymbol{w}_{\text{Nike}} \cdot \boldsymbol{w}_{\text{Male}}$

FFM $\quad \boldsymbol{w}_{\text{ESPN,A}} \cdot \boldsymbol{w}_{\text{Nike,P}} + \boldsymbol{w}_{\text{ESPN,G}} \cdot \boldsymbol{w}_{\text{Male,P}} + \boldsymbol{w}_{\text{Nike,G}} \cdot \boldsymbol{w}_{\text{Male,A}}$

FFM APPLIQUÉ DANS UN CADRE INDUSTRIEL

3

a. Approches Offline vs Online

b. Résultats

APPROCHES OFFLINE VS ONLINE

Offline learning :
Apprentissage fondé sur un dataset complet

Online learning :
La complétude des données d'entraînement n'est pas assurée. Complétion des données effectué selon un plan programmé

Méthode générale :
Comparaison du FFM avec une régression logistique comme baseline (L-BFGS avec warn start).
Utilisation du hashing trick pour réduire la dimension des données
Les variables prédites par les modèles sont les CTR et CR.

APPROCHES OFFLINE

Protocole :

- validation progressive
- réduction de la taille de l'espace de hashing afin de garantir le même nombre de paramètres pour la LR et le FFM

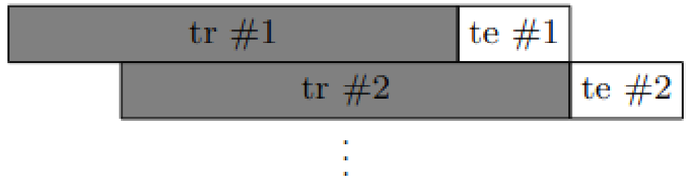


Figure 1: Progressive validation

$$\mathbf{LL}(p) = -\sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \qquad (2)$$

$$\mathbf{NLL}(p) = \frac{\mathrm{LL}(\bar{p}) - \mathrm{LL}(p)}{\mathrm{LL}(\bar{p})} \qquad (3)$$

$$\mathbf{Utility} = \sum_i \int_0^{p(\mathbf{x}_i)v_i} (y_i \cdot v_i - \tilde{c}) \Pr(\tilde{c} \mid c_i) d\tilde{c} \qquad (4)$$

APPROCHES ONLINE

Protocole :

- A/B test
- entraînement sur données live (~5B de display)
- rafraîchissement synchronisé des données pour LR et FFM
- utilisation d'un ROI comme métrique de comparaison avec la méthode offline

RÉSULTATS

Prediction model with FFM	NLL on all advertisers	NLL on small advertisers
CTR	+3.71%▲	+5.9%▲
CTR + CR	+1.21%▲	+6.2%▲

Prediction model with FFM	$\text{Utility}_{\beta=10}$ on all advertisers	$\text{Utility}_{\beta=10}$ on small advertisers	$\text{Utility}_{\beta=1000}$ on all advertisers	$\text{Utility}_{\beta=1000}$ on small advertisers
CTR	+6.29%▲	+9.70%▲	+2.22%▲	+4.39%▲
CTR + CR	+11.42%▲	+38.44%▲	+5.43%▲	+18.34%▲

Observation :

Meilleurs résultats sur les petits annonceurs

Prediction model with FFM	ROI on all advertisers	ROI on small advertisers
CTR + CR	+0.97%▲	+2.61%▲

RÉDUCTION DU TEMPS D'ENTRAÎNEMENT

4

a. Entraînement distribué

b. Pre-mature warm-start

ENTRAÎNEMENT DISTRIBUÉ

Entrainement trop long

↓

Performances réduites en production

↓

Impossible d'utilisation !

↓

Système distribué

ENTRAÎNEMENT DISTRIBUÉ

2 méthodes pour distribuer
l'algorithme de descente du gradient

Distribution synchrone :
Agrégation des modèles après une certaines quantités de données traités (en général 1 epoch)
➡ **Iterative Parameter Mixing (IPM)**

Distribution asynchrone :
Des machines sont dédié au stockage du modèle tandis que d'autres met à jour le modèle avec leurs copies locales
➡ **Parameter Server Approach**

ENTRAÎNEMENT DISTRIBUÉ

$$\text{speed-up} = \#\text{machines} \times \frac{\#\text{epochs with multiple machines}}{\#\text{epochs with one machine}}$$

Le choix se porte sur la distribution synchrone du gradient car :

- implémentation technique simple
- suppose que le temps de calcul est équitablement distribué entre les machines
- le coût de communication entre les machines est négligeable

Cependant :

- la convergence réduit à mesure que l'on augmente le nombre de machines
- il faut donc également augmenter la valeur du learning rate

η	#epochs	log loss
0.2	157	0.44585
0.5	70	0.44569
1.0	37	0.44590
2.0	26	0.44622
3.0	21	0.44654
4.0	19	0.44688
5.0	18	0.44721

ENTRAÎNEMENT DISTRIBUÉ

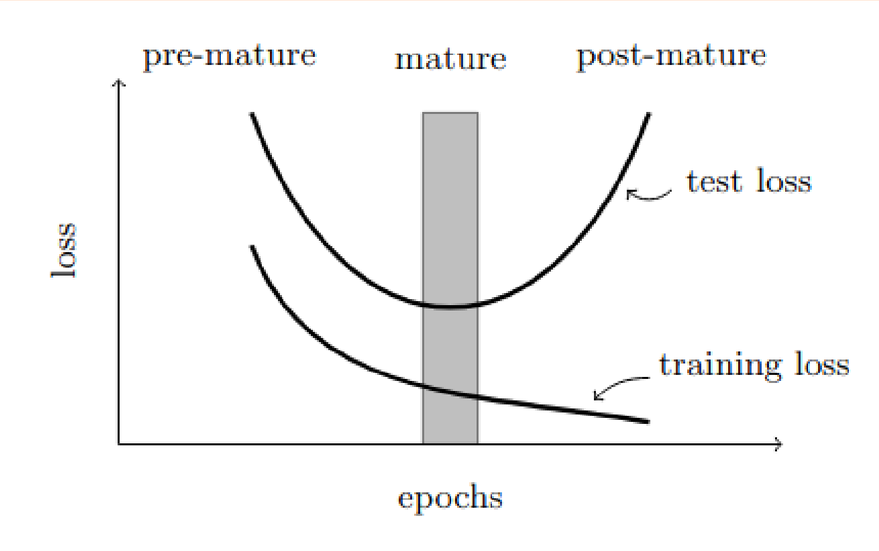
En modifiant l'algorithme IPM de telle façon à ce que le gradient soit le résultat d'une agrégation, on peut avoir un learning rate élevé et une loss faible.

On peut ainsi accélérer l'entraînement de :
32 * (8/22) soit 12 fois

Algorithm 2 Improved IPM for AdaGrad

Spread m data points into k machines
Initialize $\boldsymbol{w}$
Initialize $G \leftarrow I$
for $t \in \{1, \cdots, T\}$ **do** ▷ T: number of epochs
  Let $\boldsymbol{w}_i \leftarrow \boldsymbol{w} \quad \forall i \in \{1, \cdots, k\}$
  Let $G_i \leftarrow G \quad \forall i \in \{1, \cdots, k\}$
  for $i \in \{1, \cdots, k\}$ *parallel* **do**
    for each data point **do**
      Calculate the gradient $\boldsymbol{g}$
      Update G_i: $G_i \leftarrow G_i + \mathrm{diag}(\boldsymbol{g}\boldsymbol{g}^T)$
      Update $\boldsymbol{w}_i$: $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i - \eta G_i^{-1/2}\boldsymbol{g}$
  $\boldsymbol{w} \leftarrow \sum_{i=1}^{k} \boldsymbol{w}_i / k$
  $G \leftarrow \sum_{i=1}^{k} G_i$

PRE-MATURE WARM-START



Warm-start :
initialisation d'un modèle avec les poids d'un autre modèle
Early-stopping :
Arret automatique d'un entraînement lorsqu'une métrique ne présente aucune amélioration

État pre-mature : le modèle est entraîné avec trop peu d'epoch
État mature : le modèle est entraîné avec un nombre suffisant d'epoch
État post-mature : le modèle est entraîné avec un trop grand nombre d'epoch

PRE-MATURE WARM-START

Algorithm 4 Our proposed "pre-mature" warm-start

Require: an initial model $\mathbf{w}_{-1}$

$\mathbf{w} \leftarrow \mathbf{w}_0 \leftarrow \mathbf{w}_{-1}$

calculate the validation loss L_0

for $t \in \{1, \ldots, T\}$ **do**

  update $\mathbf{w}$

  $\mathbf{w}_t \leftarrow \mathbf{w}$

  calculate the validation loss L_t

  if $L_t > L_{t-1}$ **then**

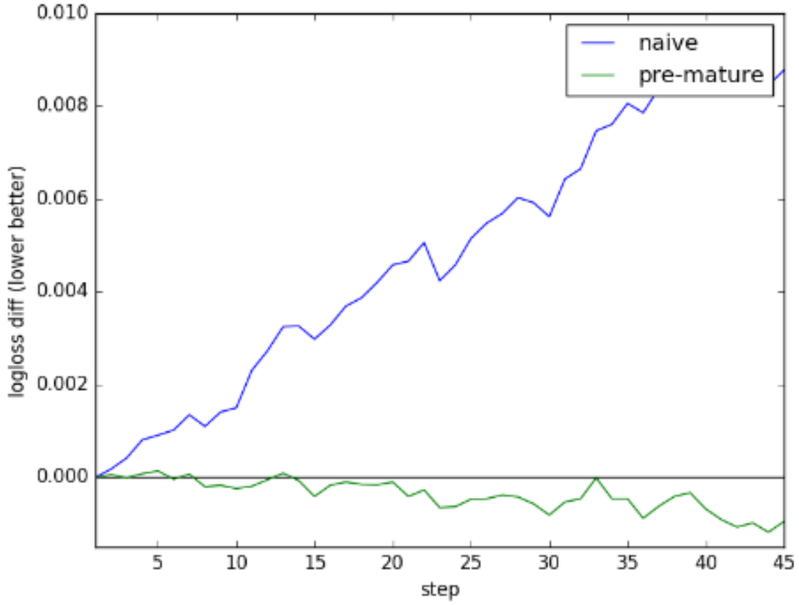
    return $(\mathbf{w}_{t-1}, \mathbf{w}_{t-2})$

Naive warm-start :

- Initialisation : modèle "mature"
- Après entraînement : obtention d'un modèle "post-mature", donc overfitting sur les anciennes données
- Observation : dégradation des performances au fur et à mesure des apprentissages.

Pre-mature warm-start (solution) :

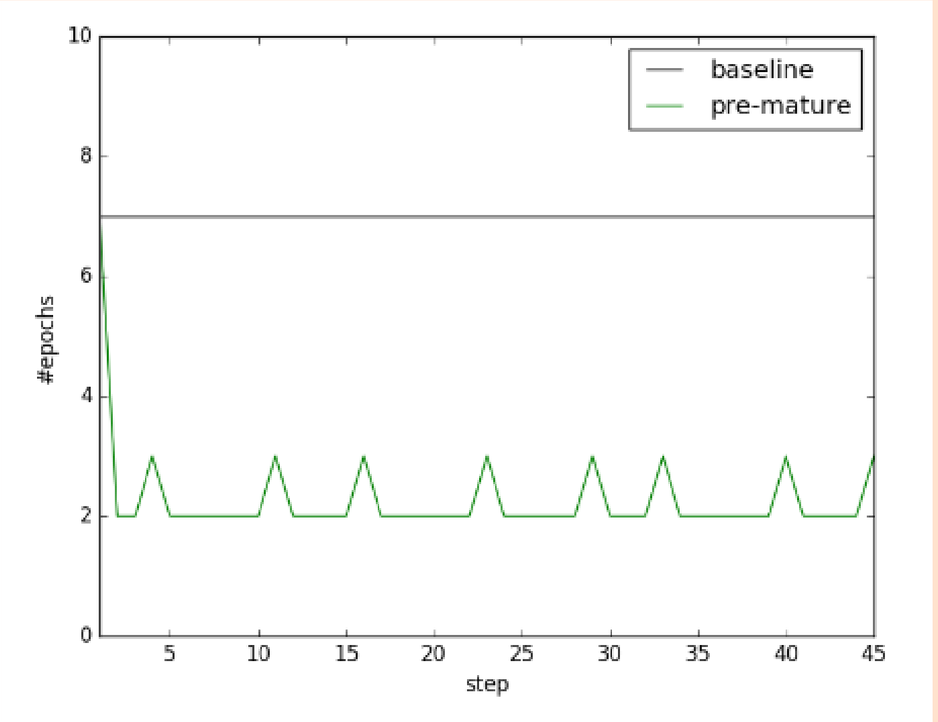
- Initialisation : modèle "pre-mature"
- Après entraînement : obtention d'un modèle "mature", donc apprentissage des nouvelles données sans sur-apprentissage sur les anciennes données.
- Nouveau modèle mature pour la prédiction, nouveau modèle pre-mature pour l'initialisation du prochain modèle



PRE-MATURE WARM-START

Le pre-mature warm start permet ainsi :

- de réduire significativement le nombre d'epoch nécessaire (et donc le temps d'entraînement)
- de conserver des performances équivalentes



CONCLUSION

Field-aware Factorization Machines peut être déployé avec succès dans des systèmes de publicité en ligne à grande échelle !

- Amélioration des "business metrics"

- Meilleure généralisation que la regression logistique (petits annonceurs)

- L'entrainement a été accéléré efficacement

  - distributed learning
  - warm start