XGBoost / Factorization machines (TP AVAZU)

Pascal Bianchi

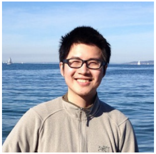
Le problème du Click Through Rate (CTR)

- Estimation de la probabilité de clic d'une banière (Criteo, Avazu, Outbrain, Wordstream, etc.)
- Central dans le pricing d'une annonce (coût par clic), l'évaluation de la qualité d'une campagne,
- Haute précision nécessaire : des améliorations même marginales de l'estimation induisent des gains importants
- Caractéristiques : données massives, features catégorielles

Données Avazu

- 6,3 GBytes, 11 jours de données Avazu, 40000000 exemples.
- Objectif : prédiction des clics

XGBoost



- Extreme Gradient Boosting (Tianqi Chen'16)
- Régulièrement dans les solutions gagnantes de challenge
  - Maksims Volkovs, Guangwei Yu and Tomi Poutanen, 1st place of the 2017 ACM RecSys challenge.
  - Vlad Sandulescu, Mihai Chiru, 1st place of the KDD Cup 2016 competition. paper.
  - Marios Michailidis, Mathias Müller and HJ van Veen, 1st place of the Dato Truely Native ? competition.
  - Vlad Mironov, Alexander Guschin, 1st place of the CERN LHCb experiment Flavour of Physics competition.
  - Josef Slavicek, 3rd place of the CERN LHCb experiment Flavour of Physics competition.
  - Mario Filho, Josef Feigl, Lucas, Gilberto, 1st place of the Caterpillar Tube Pricing competition.
  - Qingchen Wang, 1st place of the Liberty Mutual Property Inspection.
  - Chenglong Chen, 1st place of the Crowdflower Search Results Relevance.
  - Alexandre Barachant ("Cat") and Rafał Cycoń ("Dog"), 1st place of the Grasp-and-Lift EEG Detection.
  - Halla Yang, 2nd place of the Recruit Coupon Purchase Prediction Challenge.
  - Owen Zhang, 1st place of the Avito Context Ad Clicks competition.
  - Keiichi Kuroyanagi, 2nd place of the Airbnb New User Bookings.
  - Marios Michailidis, Mathias Müller and Ning Situ, 1st place Homesite Quote Conversion.

Présentation de XGBoost

- Généralités sur la régularisation
- Formalisme : le problème d'optimisation
- Apprentissage des arbres

XGBoost : modèle

Méthode d'ensemble :

$$\hat{y}_i(\theta, x_i) = \sum_{k=1}^{K} f_k(x_i)$$

où $f_1, \dots, f_K$ = arbres de régression (CART) et $\theta = (f_1, \dots, f_K)$.

Un arbre, c'est :

- une **structure** = un split à chaque noeud (= une question)
- des **scores** = valeurs portée par les feuilles

Scores et structure = paramètres du problème d'optimisation

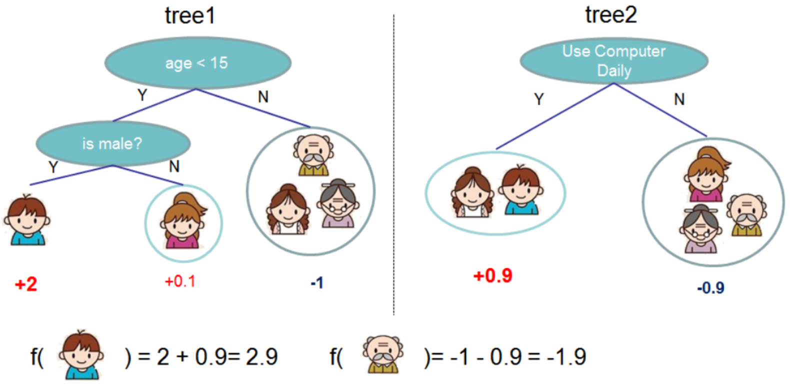


Figure : http ://homes.cs.washington.edu/ tqchen/pdf/BoostedTree.pdf

Apprentissage récursif

Le problème

$$\min_{\theta} \sum_{i=1}^{n} \ell \left(y_i, \sum_k f_k(x_i) \right) + \sum_{k=1}^{K} \Omega(f_k)$$

est intractable (minimisation conjointe sur plusieurs arbres).

Alternative : Estimer les arbres un à un

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

Problème d'optimisation revisité

A **chaque** itération t, on résoud le sous-problème

$$\begin{aligned}\min_{f_t} \; & \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i^t\right) + \sum_{k=1}^{t} \Omega(f_k) \\ & = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + \text{cste}\end{aligned}$$

On approxime l'attache aux données par son développement à l'ordre deux

$$\ell(y_i, \hat{y}_i^{(t-1)} + \delta) \simeq \ell(y_i, \hat{y}_i^{(t-1)}) + \underbrace{\partial_{\hat{y}} \ell(y_i, \hat{y}_i^{(t-1)})}_{g_i} \delta + \underbrace{\partial_{\hat{y}}^2 \ell(y_i, \hat{y}_i^{(t-1)})}_{h_i} \frac{\delta^2}{2}$$

Régularisation

Le problème est finalement

$$\min_{f_t} \sum_{i=1}^{n} \left(g_i f_t(x_i) + \frac{h_i}{2} f_t(x_i)^2 \right) + \Omega(f_t)$$

Appelons $w = (w_1, \ldots, w_T)$ les scores des T feuilles de l'arbre f_t. Formellement, l'arbre f_t s'écrit

$$f_t(x) = w_{q(x)}$$

où $q(x) \in \{1, \ldots, T\}$ est la fonction qui à toute donnée x associe l'indice de la feuille dans laquelle x est classé.

$$\Omega(f_t) = \lambda \frac{\|w\|^2}{2} + \gamma T$$

Optimisation des scores

À structure $q(\,.\,)$ fixée, les scores optimaux minimisent

$$\begin{aligned}\mathrm{obj}^{(t)} &= \sum_{i=1}^{n}[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}[\underbrace{(\sum_{i\in I_j} g_i)}_{G_j} w_j + \frac{1}{2}(\underbrace{\sum_{i\in I_j} h_i}_{H_j} + \lambda) w_j^2] + \gamma T\end{aligned}$$

où $I_j =$ ensemble des exemples i classés dans la $j$ème feuille.
La solution w_j^* est explicite ! Et la fonction objective ne dépend plus de de q :

$$\mathcal{J}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$$

Evaluer le coût $\mathcal{J}(q)$ d'une structure

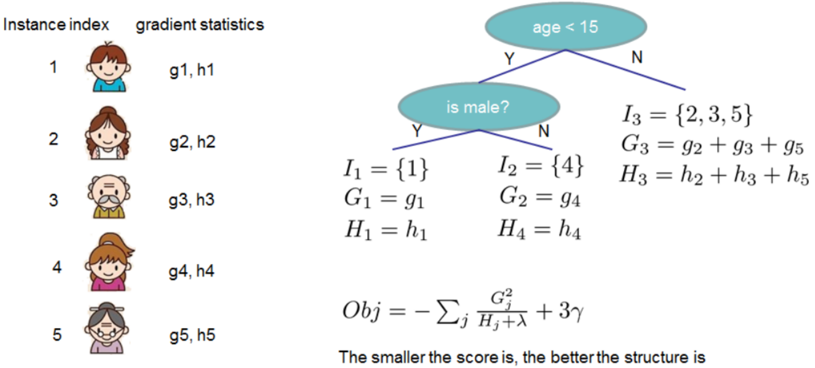


Figure : http ://homes.cs.washington.edu/ tqchen/pdf/BoostedTree.pdf

Optimisation de la structure

Il reste à minimiser $\mathcal{J}(q)$ pour obtenir f_t.

Approche greedy :

- Initialisation : $T = 1$, $J(1) = -G^2/2(H+\lambda) + \gamma$
- Pour chaque split, calculer le nouveau score $\mathcal{J}(q)$
- Appliquer le meilleur split (si amélioration il y a)
- Itérer

Remarque sur le terme γT

- on “paye” un coût γ à chaque ajout d'un feuille.
- Si le meilleur split n'apporte pas un gain supérieur à γ sur l'attache aux données, il n'est pas appliqué.
- Assure un élagage automatique de l'arbre.

Conclusion

Les ingrédients principaux :

- Des scores intégrés aux paramètres d'optimisation
- Une régularisation
- Un développement à l'ordre deux du risque
- Une implémentation parallèle efficace

Outline

XGBoost

Field-aware Factorization Machines

Modèle de régression logistique

Données : $x_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$, $i = 1, \dots n$

Problème : Optimisation du coût logistique

$$\min_w \; \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \hat{y}(w, x_i))$$

où ℓ est le coût logistique

$$\ell(y, \hat{y}) = y \log(1 + e^{-\hat{y}}) + (1 - y) \log(1 + e^{\hat{y}})$$

et $\hat{y}(w, x_i)$ est le score

$$\hat{y}(w, x_i) = w_0 + w_1 x_i^1 + \dots + w_d x_i^d \;\; = x_i^T w \,.$$

Caractéristiques du problème de CTR

Le features sont des dummies :

- Grand nombre d de features
- Chaque ligne x_i^T est un vecteur **sparse** de 0 et 1

Les interactions entre features sont importantes :

Features	Country-USA	Country-China	Day-Thanksgiving	Day-14july
x_i^T	1	0	1	0

Co-occurence de “Country-USA” et “Day-Thanksgiving” doit être prise en compte.

Idem pour Avazu : interactions `site_category et C14 par ex.`

Prendre en compte les interactions

Première solution : une nouvelle feature par paire de features

Une nouvelle feature "Country-USA-Day-Thanksgiving"

Nouveau modèle de score :

$$\hat{y}(w, x_i) = \sum_{k,\ell} W_{k,\ell}\, x_i^k x_i^\ell + w^T x_i$$

- Dimension intractable du modèle
- Même si le problème était faisable, fort risque d'overfit (grand nombre de paramètres)

Factorization Machines

Poser $W_{k,\ell} = \langle v_k, v_\ell \rangle$:

$$\hat{y}(V, x_i) = \sum_{k,\ell} \langle v_k, v_\ell \rangle \, x_i^k x_i^\ell + w^T x_i$$

où $V = (v_1, \ldots, v_d)$ matrice $L \times d$

- v_k vecteur de dimension L
- Agit comme une représentation de l'interaction que la $k$ème feature a avec les autres variables.
- L'évaluation de la double somme est en fait linéaire en d car

$$\hat{y}(V, x_i) = \| V x_i \|^2 + w^T x_i$$

Field-aware Factorization Machines

Chaque feature appartient à un champ :

Field	Country		Day	
Features	Country-USA	Country-China	Day-Thanksgiving	Day-14july
x_i^T	1	0	1	0

On labélise les champs et les features

Field index	1		2		3				
Feature index	1	2	3	4	5	6	7	8	9
x_i^T	1	0	1	0	1	0	0	1	1

Soit f la fonction qui à chaque feature-index associe son champ :

$$f_1 = 1,\ \ f_2 = 1,\ \ f_3 = 2, f_4 = 2,\ \ f_5 = 3,\ \ \text{etc.}$$

Modèle FFM

Chaque feature k interagit différemment avec les features du champ 1, celles du champ 2, etc. On a autant de vecteurs d'interaction par feature qu'il y a de champs :

$$(v_k^1, \dots, v_k^F) \quad \text{où } F = \text{nombre de champs}$$

Modèle :

$$\hat{y}(V, x_i) = \sum_{k,\ell} \langle v_k^{f_\ell}, v_\ell^{f_k} \rangle \, x_i^k x_i^\ell + w^T x_i$$

Stochastic Gradient Descent (SGD)

$$\min_{v} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \hat{y}(V, x_i)) + \frac{\lambda}{2} \sum_{f} \sum_{k} (v_k^f)^2$$

Au temps t,

- choisir $i \in \{1 \dots n\}$. On a un ligne au format :

  $$\text{field}_1\text{:feature}_1\text{: }1, \;\; \text{field}_2\text{:feature}_2\text{: }1, \;\; \text{field}_3\text{:feature}_3\text{: }1, \;\; \text{etc.}$$

- pour chaque couple de features k, ℓ *présentes* dans i,

$$\begin{aligned} v_k^{f_\ell} &\leftarrow v_k^{f_\ell} - \gamma g_k^{f_\ell} \\ v_\ell^{f_k} &\leftarrow v_\ell^{f_k} - \gamma g_\ell^{f_k} \end{aligned}$$

où $g_k^{f_\ell} = \partial_{\hat{y}} \ell(y_i, \hat{y}(V, x_i)) v_\ell^{f_k} + \lambda v_k^f$