TP : Étude de la dynamique d'Adaboost

JÉRÉMIE MARY

2 mars 2006

Ce TP porte sur la dynamique d'Adaboost. Nous réutiliserons les notations introduites en cours.

Dans Adaboost, on peut distinguer deux cas :

1. Le cas Optimal ou le classifieur h_t appris à chaque étape est optimal dans le sens où il réalise le minimun pour l'erreur empirique (ε_t) parmis tous les classifieurs possibles.
2. Le cas Non-optimal ou le classifieur h_t appris n'est pas forcément celui ayant l'erreur empirique la plus faible possible. On se contente juste d'un «suffisament» bon.

1 Réécriture d'Adaboost

Ici, on se restreindra au premier cas. On suppose de plus que le nombre de classifieurs possible est fini et vaut n La matrice des classifieurs s'écrit comme dans le cadre de la théorie des jeux. On note m le nombre d'exemples d'apprentissage :

$$M = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix}$$

où $a_{i,j} = 2 * \mathbb{1}_{\{h_j(i)=y_i\}} - 1$. C'est-à-dire que $y_{i,j}$ vaut 1 si le classifieur i classe correctement l'exemple j et -1 sinon.

On note d_t le vecteur colonne représentant la répartition des poids sur les exemples

Question 1: Avec ce formalisme que représente $j_t \in \arg\max_j (d_t' M)_j$? (d' est la transposée de d)

Rappel : Adaboost converge vers un minimun de la fonction de perte empirique $F(\lambda) = \sum_{i=1}^{m} e^{-(M\lambda)_i}$

Question 2: Justifier le fait que dans l'algorithme 1, tous les λ soient initialisés à 0.

Pour des raisons pratiques, la présentation de l'algorithme 1 précédent n'est pas bien adaptée. En effet, nous allons regarder la dynamique d'Adaboost, ce qui implique de suivre l'évolution de d_t. On propose donc de remplacer l'itération principale par :

Question 3: En utilisant le fait que M est binaire, montrer que $e^{-(M_{i,j_t}\alpha_t)} = \left(\frac{1-M_{i,j_t}r_t}{1+M_{i,j_t}r_t}\right)^{\frac{1}{2}}$

On admettra que l'injection dans l'expression de d_{t+1} de cette égalité conduit à la forme donnée dans l'algoritme 2. (bon, si vous ne me croyez pas, ne vous gênez pas pour essayer)

Agorithme 1 Réécriture d'Adaboost

Input : Matrice M, T nombre total d'itérations
Initialisation : $\lambda_{1,j} = 0$ pour $j = 1, \dots, n$
Pour $t = 1 \dots T$ **Faire**
  Pour $i = 1, \dots, m$ **Faire**
    $d_{t,i} = e^{-(M\lambda_t)_i} / \sum_{\bar{i}=1}^{m} e^{(-M\lambda_t)_{\bar{i}}}$
    $j_t \in \arg\max_j (d_t' M)_j$
    $r_t = (d_t' M)_{j_t}$
    $\alpha_t = \frac{1}{2} \ln(\frac{1+r_t}{1-r_t})$
    $\lambda_{t+1} = \lambda_t + \alpha_t e_j t$ (e_{j_t} est un vecteur ligne valant 1 en position j_t et 0 ailleurs)
  Fin Pour
Fin Pour
Ouput : $\lambda_T / ||\lambda_T||_1$

Agorithme 2 Itérations d'Adaboost

$j_t \in \arg\max_j (d_t' M)_j$
$r_t = (d_t' M)_{j_t}$
Pour i=1,...,m **Faire**
  $d_{t+1,i} = \frac{d_{t,i}}{1+M_{i,j_t} r_t}$
Fin Pour

Question 4: Que devient la phase d'initialisation ?

Question 5: Coder l'agorithme 2 (en n'oubliant pas la boucle pricipale et l'initialisation) dans le langage de votre choix. Mis-a part si vous maîtrisez des librairies gérant des matrice et des affichages de graphiques, je vous conseille d'utiliser `matlab` ou `octave` (équivalent libre, pensez a installer le package octave-forge). *Attention, j'ai volontairement introduit une notation tendancieuse $d_{t,i}$, a vous de vous débrouiller pour mettre les transposées au bon endroit... pour vos test n'utilisez pas des matrices carrées ou vous ferez n'importe quoi.*

2 En pratique

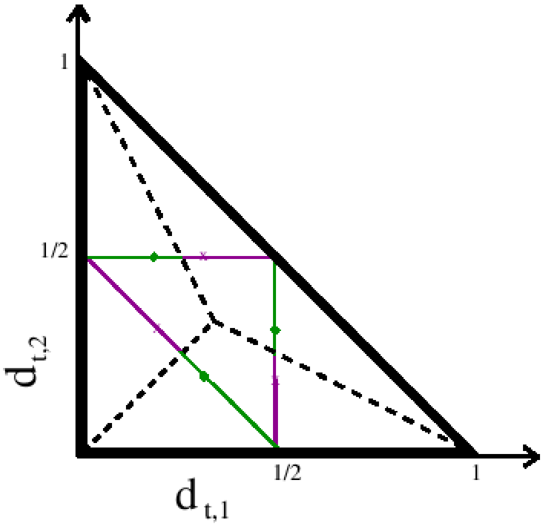
Question 6: Que se passe t-il si votre matrice M contient une ligne (resp. une colonne de 1) ? Même question avec des -1. Expliquez les résultats.

Faites tourner votre algorithme avec comme matrice :

$$M = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

Question 7: Observez l'évolution des densités sur les exemples (d_t) - par exemple en tracant $d_1(t)$ en fonction de $d_2(t)$ sur 100 pas de boosting -. Que remarquez vous ?

Question 8: Contrôlez votre code. N'y aurait-il pas un problème de déviation numérique ? Si oui, lequel ? pourquoi ? Proposez une rustine ou un mécanisme de contrôle sur les itérations.



Question 9: Sur la figure précédente, représentez avec des flèches l'évolution des densités en fonction des différentes zones. (pour la lisibilité il sera sans doute nécessaire de représenter plusieurs fois la figure).

Question 10: Étudiez maintenant seulement les 50 premières itérations. de nombreux points sont presque au même endroit. Proposez un mécanisme de représentation permettant de mieux lire la dynamique.

Regardez maintenant ce qui se passe avec les familles de matrices de la formes :

$$M = \begin{pmatrix} -1 & 1 & \cdots & 1 & 1 \\ 1 & -1 & 1 & & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & & & -1 & 1 \\ 1 & 1 & \cdots & 1 & -1 \end{pmatrix}$$

Question 11: Émettez une conjecture pour ces matrices.

Question 12: Essayez maintenant avec des matrices choisies par vos soins plus ou moins aléatoires. Essayez de mettre en évidence des comportements différents et des «attracteurs» pour les cycles. Essayez $n >> m$,$n << m$, variation de la proportion de 1...

3 Marges

Rappel : La marge est définie sur la base d'exemples E comme étant

$$\sum_{x_i \in E} \sum_{t=1}^{T} h_t(x_i) y_i = \sum_{i \in m} (M\lambda)_i$$

avec $||\lambda||_1 = 1$

Une explication qui a été avancée pour expliquer le bon fonctionnement d'Adaboost est sa «tendance» à optimiser la marge. Si Adaboost est dans un cycle, alors sa marge peut se calculer comme suit :

$$\text{marge} = \frac{-\ln \prod_{t=1}^{T_{cycle}} (1 - r_t^2)}{\ln \prod_{t=1}^{T_{cycle}} \frac{1 + r_t}{1 - r_t}}$$

Si n et m ne sont pas trop grands, on peut tenter de trouver le λ qui maximise la marge sans utiliser Adaboost.

Question 13: Écrire et coder un algorithme trouvant une approximation d'un λ optimal par une technique de force brute (vous pouvez utiliser une grille, une grille auto-adaptative ou des techniques basées sur la discrépance).

Question 14: Comparez les marges obtenues par force et brute et par Adaboost pour des matrices `8x8`. Que peut on en conclure pour répondre à la question : Adaboost maximise t-il la marge ?

Question 15: (bonus) Si vous avez encore du temps, réfléchissez à une façon de traiter le cas non-optimal.

Question 16: (bonus) Si vous avez encore davantage de temps, représentez r_t en fonction de T (attention cela requiert de faire attention au niveau numérique). Faites cette courbe pour différentes matrices et différentes initialisations de d. Vous devriez pourvoir trouver des alternances entre stabilisation dans un k-cycle et du chaos…