

Algorithme parallèle de Descente de gradient stochastique multi-classes pour la classification d'images

Réalisateur : NGUYEN Quoc Khai, promotion 17, IFI

Superviseurs : DO Thanh Nghi(CTU), PHAM Nguyen Khang(CTU) et HO Tuong Vinh(IFI)

Stage M2 réalisé à la laboratoire IDPL, CTU, Cantho, 2014



Plan de présentation

Introduction

- Motivation

- Contribution

État de l'art

- Processus de la classification d'images

- Apprentissage automatique

Algorithme SGD-SVM multi-classes

- Algorithme linéaire de SGD-SVM multi-classes

- Équilibrage de MC-SGD

- Parallélisation de MC-SGD

Expérimentation

- Jeux de données

- Résultat obtenu

Conclusion

Plan de présentation

Introduction

Motivation

Contribution

État de l'art

Processus de la classification d'images

Apprentissage automatique

Algorithme SGD-SVM multi-classes

Algorithme linéaire de SGD-SVM multi-classes

Équilibrage de MC-SGD

Parallélisation de MC-SGD

Expérimentation

Jeux de données

Résultat obtenu

Conclusion

Classification d'images

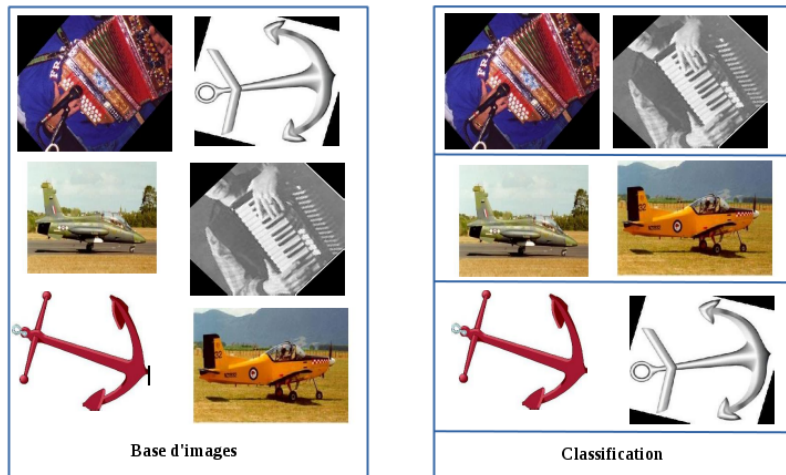


Figure 1: La classification d'image

Application de la classification d'images

La classification d'images est appliquée dans plusieurs domaines importants :

Application de la classification d'images

La classification d'images est appliquée dans plusieurs domaines importants :

- ▶ Reconnaissance des chiffres sur des chèques

Application de la classification d'images

La classification d'images est appliquée dans plusieurs domaines importants :

- ▶ Reconnaissance des chiffres sur des chèques
- ▶ Reconnaissance des codes postaux pour la classification automatique des courriers

Application de la classification d'images

La classification d'images est appliquée dans plusieurs domaines importants :

- ▶ Reconnaissance des chiffres sur des chèques
- ▶ Reconnaissance des codes postaux pour la classification automatique des courriers
- ▶ Reconnaissance des visages pour l'authentification

Difficulté de la classification d'images

Les données sorties des images sont complexes :

Difficulté de la classification d'images

Les données sorties des images sont complexes :

- ▶ Les dimension sont grandes

Difficulté de la classification d'images

Les données sorties des images sont complexes :

- ▶ Les dimension sont grandes
- ▶ Beaucoup de classes

Difficulté de la classification d'images

Les données sorties des images sont complexes :

- ▶ Les dimension sont grandes
- ▶ Beaucoup de classes

Solution

→ Il faut améliorer la vitesse

Contribution

Pendant ce stage, nous avons proposé :

Contribution

Pendant ce stage, nous avons proposé :

- ▶ Un algorithme SGD-SVM multi-classes

Contribution

Pendant ce stage, nous avons proposé :

- ▶ Un algorithme SGD-SVM multi-classes
- ▶ Une version parallèle de cet algorithme

Contribution

Pendant ce stage, nous avons proposé :

- ▶ Un algorithme SGD-SVM multi-classes
- ▶ Une version parallèle de cet algorithme
- ▶ Implémentation de MC-SGD-Toy pour étudier l'algorithme MC-SGD

Contribution

Pendant ce stage, nous avons proposé :

- ▶ Un algorithme SGD-SVM multi-classes
- ▶ Une version parallèle de cet algorithme
- ▶ Implémentation de MC-SGD-Toy pour étudier l'algorithme MC-SGD

Résultats expérimentaux :

- ▶ Égale au SVM pour la classification multi-classes
- ▶ Plus vite que le SVM de 26 fois à 3902 fois

Plan de présentation

Introduction

Motivation

Contribution

État de l'art

Processus de la classification d'images

Apprentissage automatique

Algorithme SGD-SVM multi-classes

Algorithme linéaire de SGD-SVM multi-classes

Équilibrage de MC-SGD

Parallélisation de MC-SGD

Expérimentation

Jeux de données

Résultat obtenu

Conclusion

Processus de la classification d'images

Processus de la classification d'images

1. Extraction des caractéristiques des images : obtenir des vecteurs de caractéristiques (SIFT)

Processus de la classification d'images

1. Extraction des caractéristiques des images : obtenir des vecteurs de caractéristiques (SIFT)
2. Construire un dictionnaire et calculer le fréquence de chaque mot apparaît dans chaque image : obtenir des histogrammes de chaque image (BOW)

Processus de la classification d'images

1. Extraction des caractéristiques des images : obtenir des vecteurs de caractéristiques (SIFT)
2. Construire un dictionnaire et calculer le fréquence de chaque mot apparaît dans chaque image : obtenir des histogrammes de chaque image (BOW)
3. Application SVM pour l'apprentissage automatique sur des histogrammes

Processus de la classification d'images (2)

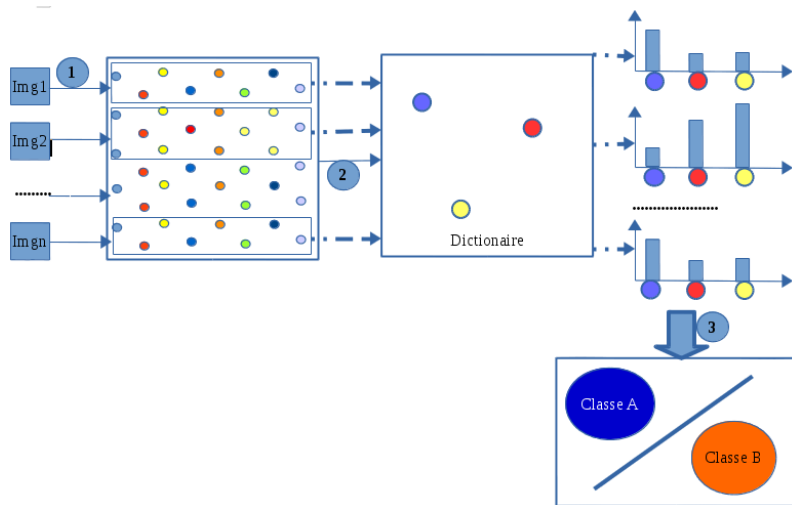


Figure 2: Processus de la classification d'image

Méthode SVM

- ▶ Méthode d'apprentissage supervisé importante
- ▶ Cherche l'hyperplan optimal séparant les données en deux classes

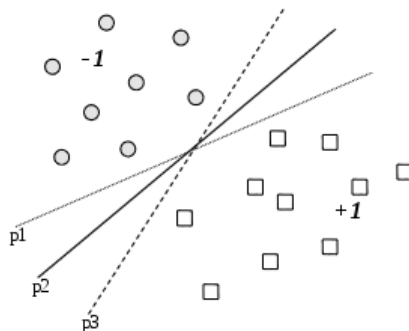


Figure 3: Classification linéaire

Méthode SVM - support hyperplan

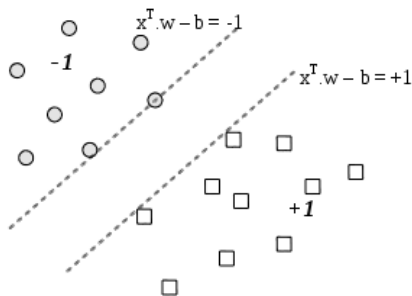


Figure 4: Support hyperplan

Méthode SVM - hyperplan optimal

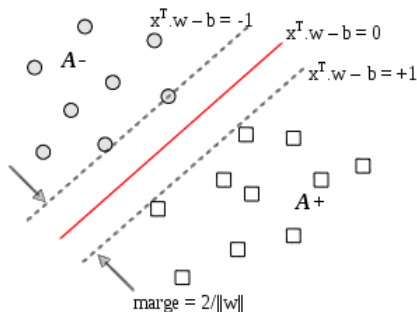


Figure 5: L'hyperplan optimal

Méthode SVM - problème

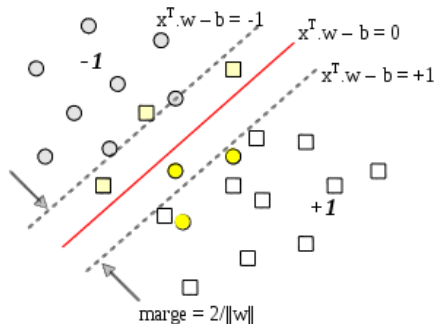


Figure 6: Erreurs de SVM

- SVM obtient w et b après avoir trouvé la solution du programme quadratique $f(w, b, z)$ (détaille dans le rapport)

Méthode SGD-SVM

- Ignorer b dans le programme quadratique $f(w, b, z)$ pour obtenir le programme $\Psi(w, [x, y])$ (détaille dans le rapport)

Méthode SGD-SVM

- ▶ Ignorer b dans le programme quadratique $f(w, b, z)$ pour obtenir le programme $\Psi(w, [x, y])$ (détaille dans le rapport)
- ▶ w est mis à jour en T étapes

Méthode SGD-SVM

- ▶ Ignorer b dans le programme quadratique $f(w, b, z)$ pour obtenir le programme $\Psi(w, [x, y])$ (détaille dans le rapport)
- ▶ w est mis à jour en T étapes
- ▶ A chaque étape t , SGD prend aléatoirement n exemples (x_i, y_i) pour calculer le sous-gradient et met à jour w_{t+1}

Méthode SGD-SVM

- ▶ Ignorer b dans le programme quadratique $f(w, b, z)$ pour obtenir le programme $\Psi(w, [x, y])$ (détaille dans le rapport)
- ▶ w est mis à jour en T étapes
- ▶ A chaque étape t , SGD prend aléatoirement n exemples (x_i, y_i) pour calculer le sous-gradient et met à jour w_{t+1}

$$w_{t+1} = w_t - \eta_t \nabla_w \Psi(w_t, [x_i, y_i]) \quad (1)$$

Méthode SGD-SVM

- ▶ Ignorer b dans le programme quadratique $f(w, b, z)$ pour obtenir le programme $\Psi(w, [x, y])$ (détaille dans le rapport)
- ▶ w est mis à jour en T étapes
- ▶ A chaque étape t , SGD prend aléatoirement n exemples (x_i, y_i) pour calculer le sous-gradient et met à jour w_{t+1}

$$w_{t+1} = w_t - \eta_t \nabla_w \Psi(w_t, [x_i, y_i]) \quad (1)$$

Algorithm 5 L'algorithm d'apprentissage SGD-SVM binaire

```
1: procedure trainBinaireSGDSVM( $X, y, \lambda, T, n$ )
2:   Initialiser  $w_1 = 0$ 
3:   for  $t = 0; t < T; t++$  do
4:      $\eta_t = \frac{1}{\lambda t}$ 
5:     Choisir  $n$  exemples aléatoire dans  $X$ 
6:      $w_{t+1} = w_t - \eta_t \nabla_w \Psi(w_t, [x_i, y_i])$ 
7:   Return  $w_t$ 
```


Plan de présentation

Introduction

Motivation

Contribution

État de l'art

Processus de la classification d'images

Apprentissage automatique

Algorithme SGD-SVM multi-classes

Algorithme linéaire de SGD-SVM multi-classes

Équilibrage de MC-SGD

Parallélisation de MC-SGD

Expérimentation

Jeux de données

Résultat obtenu

Conclusion

Méthode SGD - propriété

- ▶ SGD résout le problème de classification de 2 classes

Méthode SGD - propriété

- ▶ SGD résout le problème de classification de 2 classes
→ Comment résoudre le problème de multi-classes ?

Méthode SGD - propriété

- ▶ SGD résout le problème de classification de 2 classes
→ Comment résoudre le problème de multi-classes ?
- ▶ Diviser d'abord le problème en plusieurs problèmes de 2 classes

Méthode SGD - propriété

- ▶ SGD résout le problème de classification de 2 classes
→ Comment résoudre le problème de multi-classes ?
- ▶ Diviser d'abord le problème en plusieurs problèmes de 2 classes
- ▶ Chaque problème de 2 classes est traité par la classification binaire
- ▶ One-versus-one [3] et one-versus-all [4].

Méthode MC-SGD - 1-vs-1

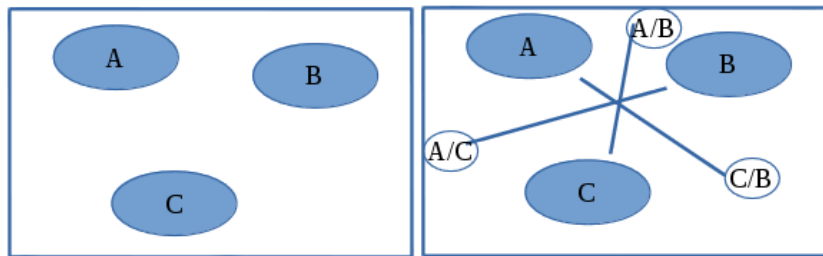


Figure 7: One-versus-one

- Construire $k(k-1)/2$ (k est le nombre de classes) classificateurs

Méthode MC-SGD - 1-vs-1

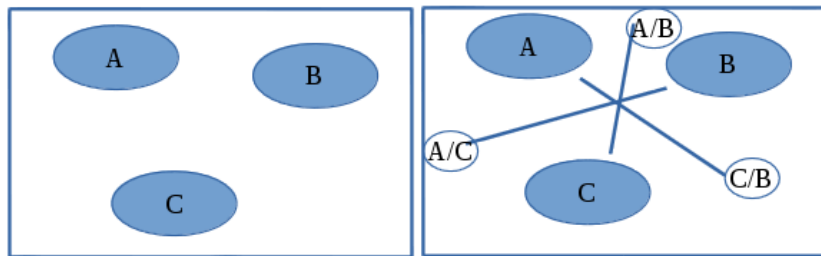


Figure 7: One-versus-one

- ▶ Construire $k(k-1)/2$ (k est le nombre de classes) classificateurs
- ▶ Prédiction : voter et la majorité des classificateurs va être choisie

Méthode MC-SGD - 1-vs-all

- ▶ Problèmes de 1-vs-1 : beaucoup de classificateurs $(k(k-1)/2)$
- ▶ 1-vs-all construit k classificateurs

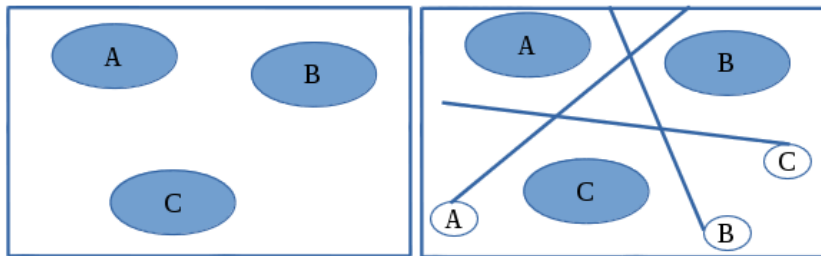


Figure 8: One-versus-all

Méthode MC-SGD - 1-vs-all

- ▶ Problèmes de 1-vs-1 : beaucoup de classificateurs $(k(k-1)/2)$
- ▶ 1-vs-all construit k classificateurs

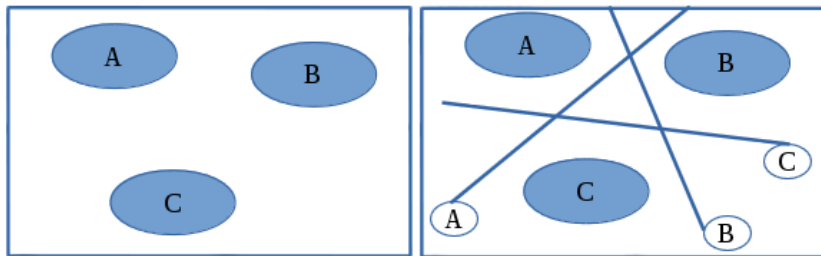


Figure 8: One-versus-all

Prédiction : la classe ayant la distance la plus courte entre son classificateur et l'exemple d'entrée est choisie.

MC-SGD - problème de balance

- ▶ Si le problème de 1000 classes, on a 1 exemple positive et 999 exemples négatives
- ▶ Les exemples positives sont rarement choisis (0.1%)
- ▶ Les exemples négatives sont très souvent choisis (99.9%)

MC-SGD - problème de balance

- ▶ Si le problème de 1000 classes, on a 1 exemple positive et 999 exemples négatives
- ▶ Les exemples positives sont rarement choisis (0.1%)
- ▶ Les exemples négatives sont très souvent choisis (99.9%)
- ▶ Comment balancer l'ensemble d'exemples ?

MC-SGD - problème de balance

- ▶ Si le problème de 1000 classes, on a 1 exemple positive et 999 exemples négatives
- ▶ Les exemples positives sont rarement choisis (0.1%)
- ▶ Les exemples négatives sont très souvent choisis (99.9%)
- ▶ Comment balancer l'ensemble d'exemples ?

Solutions

- ▶ Sous-échantillonnage : augmentation la probabilité sélectionnée de l'exemple positive

MC-SGD - problème de balance

- ▶ Si le problème de 1000 classes, on a 1 exemple positive et 999 exemples négatives
- ▶ Les exemples positives sont rarement choisis (0.1%)
- ▶ Les exemples négatives sont très souvent choisis (99.9%)
- ▶ Comment balancer l'ensemble d'exemples ?

Solutions

- ▶ Sous-échantillonnage : augmentation la probabilité sélectionnée de l'exemple positive
- ▶ Coût de données : mettre à jours w par des différents cas par rapport des erreurs obtenues

Parallélisation de MC-SGD

- ▶ MC-SGD est linéaire, apprendre des modèles l'un après l'autre
- ▶ Sur des ordinateurs multi-cœurs : pas économiser des ressources

Parallélisation de MC-SGD

- ▶ MC-SGD est linéaire, apprendre des modèles l'un après l'autre
- ▶ Sur des ordinateurs multi-cœurs : pas économiser des ressources
→ Parallélisation MC-SGD est une bonne solution

Parallélisation de MC-SGD

- ▶ MC-SGD est linéaire, apprendre des modèles l'un après l'autre
- ▶ Sur des ordinateurs multi-cœurs : pas économiser des ressources
→ Parallélisation MC-SGD est une bonne solution
- ▶ k processeurs apprennent k modèles en parallèle

Parallélisation de MC-SGD

- ▶ MC-SGD est linéaire, apprendre des modèles l'un après l'autre
- ▶ Sur des ordinateurs multi-cœurs : pas économiser des ressources
→ Parallélisation MC-SGD est une bonne solution
- ▶ k processeurs apprennent k modèles en parallèle
- ▶ Nous choisissons OpenMP pour paralléliser cet algorithme

SGD-Toy de MC-SGD

Interface pour la démonstration de MC-SGD, se base sur SVM-Toy[5]

Données :

- ▶ Chaque point est un exemple
- ▶ x , y sont 2 caractéristiques d'exemples

On applique MC-SGD pour la classification

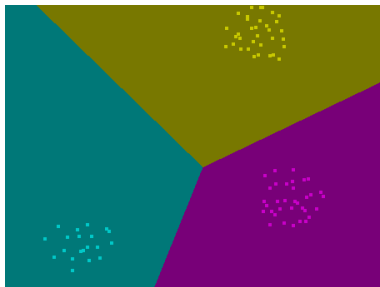


Figure 9: MC-SGD-Toy

Plan de présentation

Introduction

Motivation

Contribution

État de l'art

Processus de la classification d'images

Apprentissage automatique

Algorithme SGD-SVM multi-classes

Algorithme linéaire de SGD-SVM multi-classes

Équilibration de MC-SGD

Parallélisation de MC-SGD

Expérimentation

Jeux de données

Résultat obtenu

Conclusion

Expérimentation

Logiciels références

- ▶ Extraction des caractéristiques : SIFT[low99], K-Means(pour clustering)
- ▶ Classification binaire : SGD (implémentation Pegasos).

Matériels utilisées

- ▶ PC core-i7, 4 double cœurs, 3G RAM
- ▶ Système d'exploitation : GNU-Linux Fedora 20

Expérimentation

Logiciels références

- ▶ Extraction des caractéristiques : SIFT[low99], K-Means(pour clustering)
- ▶ Classification binaire : SGD (implémentation Pegasos).

Matériels utilisées

- ▶ PC core-i7, 4 double cœurs, 3G RAM
- ▶ Système d'exploitation : GNU-Linux Fedora 20

Nos implémentations

Expérimentation

Logiciels références

- ▶ Extraction des caractéristiques : SIFT[low99], K-Means(pour clustering)
- ▶ Classification binaire : SGD (implémentation Pegasos).

Matériels utilisées

- ▶ PC core-i7, 4 double cœurs, 3G RAM
- ▶ Système d'exploitation : GNU-Linux Fedora 20

Nos implémentations

- ▶ Classification multi-classes : MC-SGD et MC-SGD parallèle, implémentés en C/C++ en utilisant l'OpenMP

Expérimentation

Logiciels références

- ▶ Extraction des caractéristiques : SIFT[low99], K-Means(pour clustering)
- ▶ Classification binaire : SGD (implémentation Pegasos).

Matériels utilisées

- ▶ PC core-i7, 4 double cœurs, 3G RAM
- ▶ Système d'exploitation : GNU-Linux Fedora 20

Nos implémentations

- ▶ Classification multi-classes : MC-SGD et MC-SGD parallèle, implémentés en C/C++ en utilisant l'OpenMP
- ▶ Outils pour étudier notre algorithme : MC-SGD-Toy, basé sur SVM-Toy[5]

Jeux de données

Données	#classes	#train	#test	#mots(bow)
Cal 101	101	1515	1515	124000
Cal 7 3D	7	4290	2145	5000
ImgNet 3d	10	4450	2225	5000
ImgNet	10	4450	2225	50000
Protein	3	14895	6621	357
Mnist	10	30000	30000	780

Table 1: Information sur des données

Classification d'images avec SVM et SGD multi-classes

Données	SVM(%)	SGD(%)	SVM(s)	SGD(s)	$\frac{SVM(s)}{SGD(s)}$
Cal 101	61.52	65.12	2873	106.95	26.9
Cal 7 3D	91.52	88.3	113.4	0.81	140
ImgNet 3d	76.54	75.8	144	0.90	160
ImgNet	84.08	86.58	327	1.64	199.4
Protein	68.23	68.41	551	0.20	2755
Mnist	86.92	86.46	2810	0.72	3902.8

Table 2: Comparaison entre LIBSVM et MC-SGD parallèle

Classification avec SVM et MC-SGD

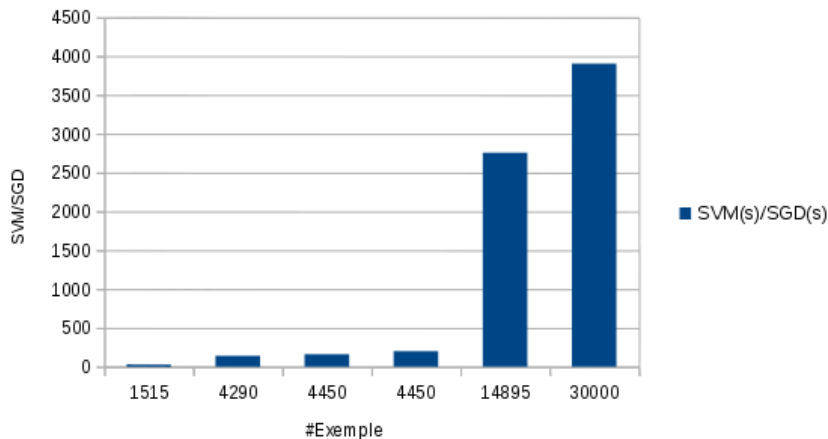


Figure 10: Comparaison de SVM et MC-SGD (SVM/MC-SGD)

Plan de présentation

Introduction

Motivation

Contribution

État de l'art

Processus de la classification d'images

Apprentissage automatique

Algorithme SGD-SVM multi-classes

Algorithme linéaire de SGD-SVM multi-classes

Équilibration de MC-SGD

Parallélisation de MC-SGD

Expérimentation

Jeux de données

Résultat obtenu

Conclusion

Conclusion

- Nous avons étudié le processus pour la classification d'images utilisant la méthode SVM

Conclusion

- ▶ Nous avons étudié le processus pour la classification d'images utilisant la méthode SVM
- ▶ Nous avons implémenté la méthode MC-SGD de classification multi-classes

Conclusion

- ▶ Nous avons étudié le processus pour la classification d'images utilisant la méthode SVM
- ▶ Nous avons implémenté la méthode MC-SGD de classification multi-classes
- ▶ Résoudre le problème de balance quand la base de données est de multi-classes

Conclusion

- ▶ Nous avons étudié le processus pour la classification d'images utilisant la méthode SVM
- ▶ Nous avons implémenté la méthode MC-SGD de classification multi-classes
- ▶ Résoudre le problème de balance quand la base de données est de multi-classes
- ▶ Paralléliser MC-SGD

Conclusion

- ▶ Nous avons étudié le processus pour la classification d'images utilisant la méthode SVM
- ▶ Nous avons implémenté la méthode MC-SGD de classification multi-classes
- ▶ Résoudre le problème de balance quand la base de données est de multi-classes
- ▶ Paralléliser MC-SGD
- ▶ Implémenter MC-SGD-Toy pour étudier l'algorithme MC-SGD

Conclusion

- ▶ Nous avons étudié le processus pour la classification d'images utilisant la méthode SVM
- ▶ Nous avons implémenté la méthode MC-SGD de classification multi-classes
- ▶ Résoudre le problème de balance quand la base de données est de multi-classes
- ▶ Paralléliser MC-SGD
- ▶ Implémenter MC-SGD-Toy pour étudier l'algorithme MC-SGD
- ▶ MC-SGD est plus vite que SVM, correspond bien pour la classification d'image

Perspective

- ▶ SGD est difficile de choisir des paramètres entrées

Perspective

- ▶ SGD est difficile de choisir des paramètres entrées
- ▶ Nous étudierons pour chercher des paramètres optimales de la méthode






Perspective

- ▶ SGD est difficile de choisir des paramètres entrées
- ▶ Nous étudierons pour chercher des paramètres optimales de la méthode
- ▶ Nous testerons aussi des bases d'images plus grandes tel que ImageNet

Perspective

- ▶ SGD est difficile de choisir des paramètres entrées
- ▶ Nous étudierons pour chercher des paramètres optimales de la méthode
- ▶ Nous testerons aussi des bases d'images plus grandes tel que ImageNet
- ▶ Nous développerons pour que SGD s'adapte à la classification de vidéos

Références

-  S.Schwartz, Y.Singer and N.Srebro *Pegasos: Primal estimated sub-gradient solver for svm.*, Proceedings of the Twenty-Fourth International Conference Machine Learning, pp. 807-814. ACM (2007)
-  David G. Lowe, *Object Recognition from Local Scale-Invariant Features*, In proceeding of the 7th International conference of computer vision, pages 1150 - 1157, Corfou, Grèce, 1999.
-  V.Vapnik, *The Nature of Statistical Learning Theory.*, Springer, New York (1995).
-  U.Krebel, *Pairwise classification and support vector machines*, Support Vector Learning, Advances in Kernel Methods. pp.255–268. (1999).
-  C.Chang and C.Lin, *LIBSVM – a library for support vector machines*, <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (2001).

Merci pour votre attention!!!!