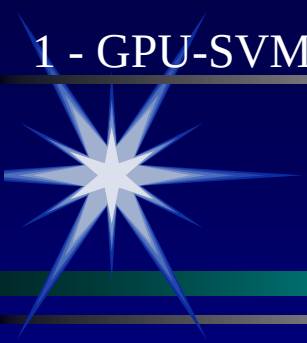
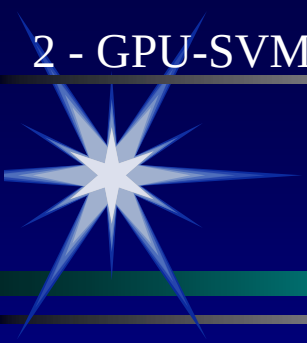


Passage à l'échelle dans la recherche d'information multimédia

Algorithmes parallèles pour la classification de grands ensembles de données

François Poulet



SVM incrémental et parallèle sur GPU

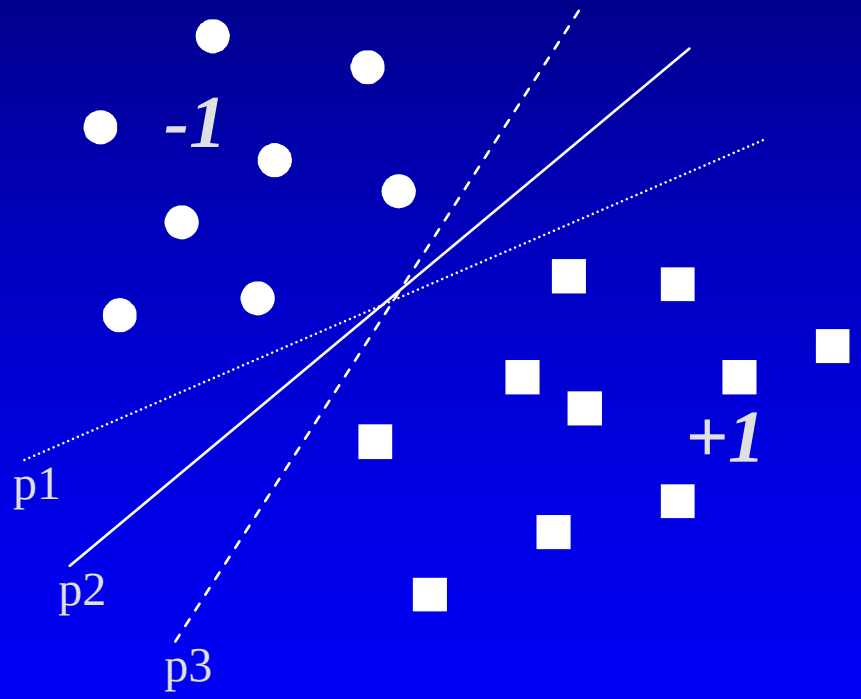
Introduction

- algorithme de SVM (Vapnik, 1995)
  - hyperplan de séparation des données en 2 classes
  - SVM + fonction de noyau = méthode robuste
  - classification, régression, détection d'outlier
  - bons résultats
- programme quadratique : coûteux
- contribution
  - apprentissage incrémental et parallèle LS-SVM
  - fouille de très grands ensembles de données sur GPU

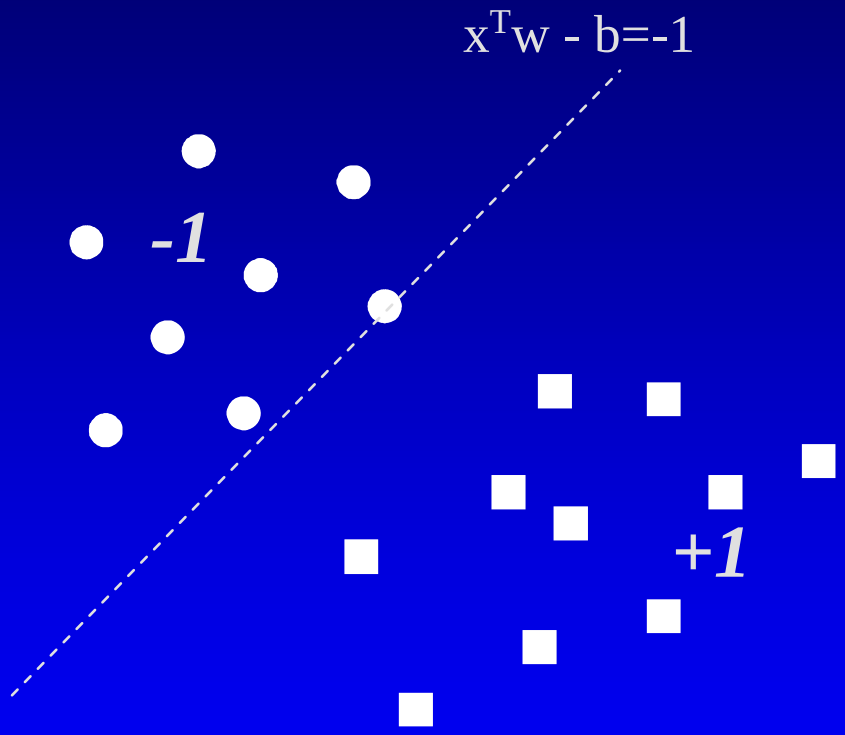
GPU ???

- processeur spécialisé pour accélération graphique
- avantages / CPU pour calcul intensif :
  - meilleur taux de transfert mémoire
  - meilleurs possibilités en calculs flottants
  - centaines d'unités parallèles en SIMD
- alternative intéressante au cluster de CPUs
- GPUs récentes incluent possibilités calcul non graphiques : GP-GPU (General Purpose)

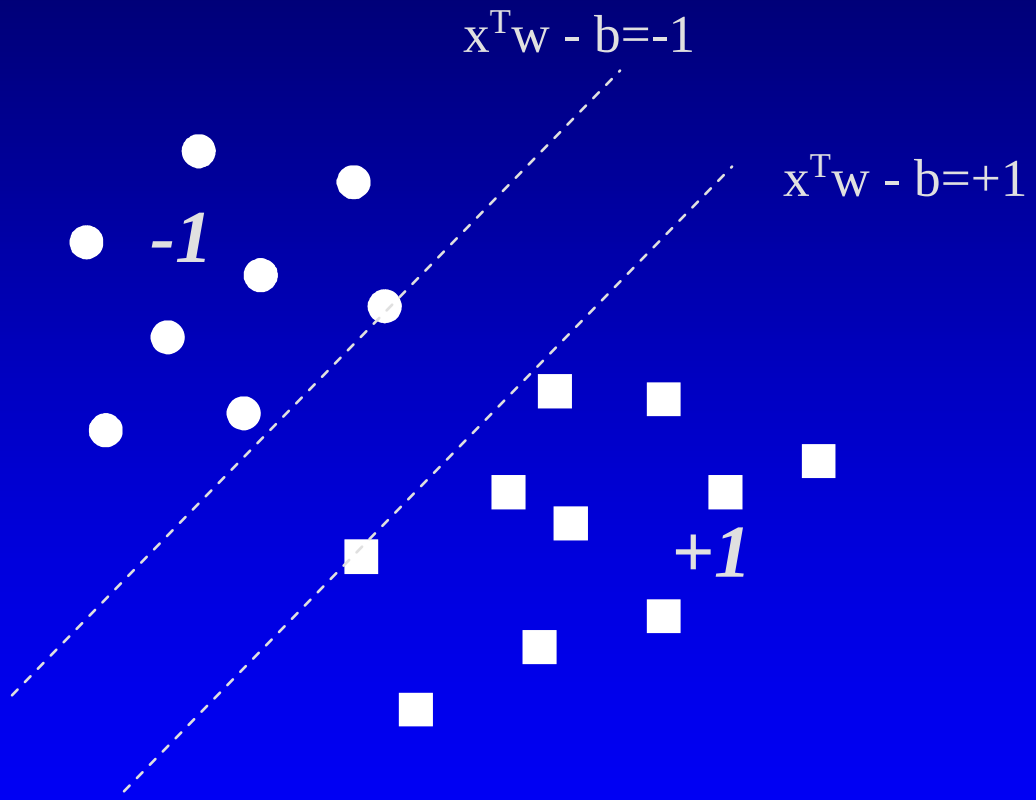
Support vector machine



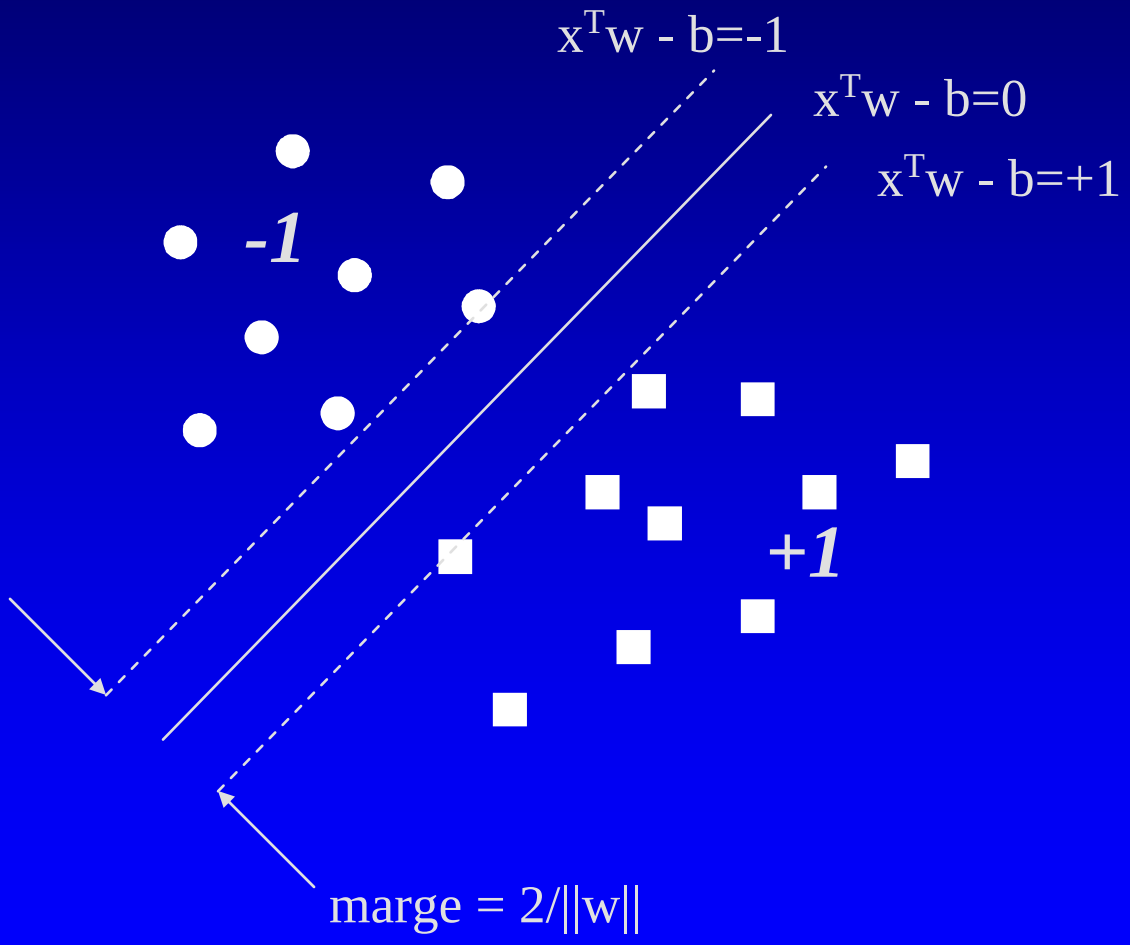
Support vector machine



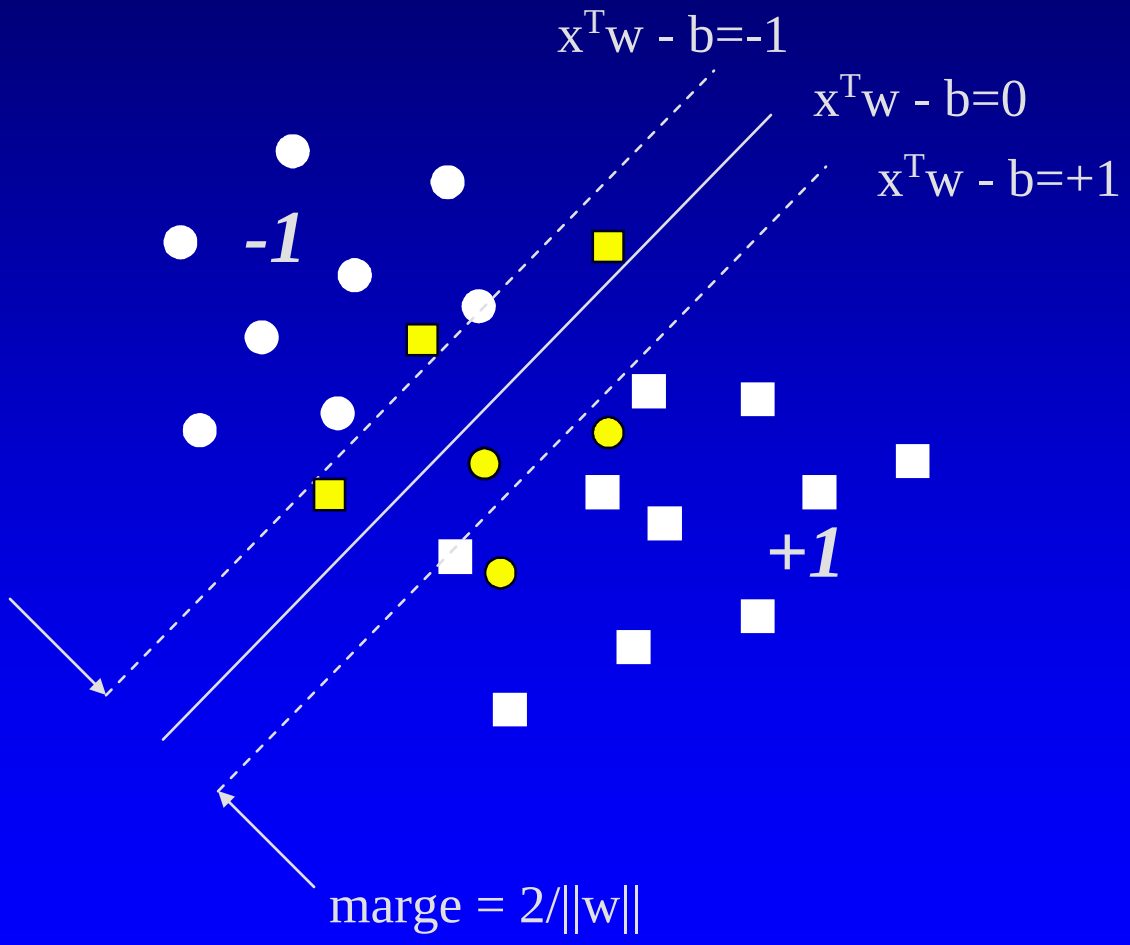
Support vector machine



Support vector machine



Support vector machine



Notations

- A_{mxn} : matrice représentant m ind. en n dim.
- D_{mxm} : matrice diagonale des classes (±1) des m ind.
- e : vecteur colonne de 1
- X^T : transposée de X
- w, b : plan
- ||w|| : norme-2 du vecteur w

Algorithme de SVM

- maximisation marge + minimisation erreurs
  - min $f(z,w,b) = (c/2)\,\|z\| + (1/2)\,\|w\|^2$
  - avec $D(Aw - eb) + z >= e$ (1)
  - où $z >= 0$ variable de ressort et c constante positive
- résolution du programme quadratique (1) : w, b
- classification d'un nouvel individu x : signe($x^T w - b$)

LS-SVM (Suykens et al., 1999)

- Least Squares SVM
  - remplace inégalité par égalité : $D(Aw - eb) + z = e$
  - minimisation des erreurs : min $(1/2) \|z\|^2$

- LS-SVM : système linéaire à (n+1) inconnues (w, b)
  - au lieu du programme quadratique

Apprentissage incrémental

- grands ensembles de données
  - limite : mémoire => apprentissage incrémental
  - petit bloc de données en mémoire à la fois
  - même solution
- nombre individus << nombre dimensions
  - incrémental en colonne
- nombre individus >> nombre dimensions
  - incrémental en ligne

Apprentissage incrémental

- l'équation à résoudre est :
- $(w_1\ w_2 \ldots w_n\ b)^T = (I°/c + E^TE)^{-1}E^TDe$

- complexité spatiale :
  - E^TE et E^TDo : [n+1 x n+1] et [n+1 x 1]
  - E, E^T et D : m x (n+1), (n+1) x m et m x m

  - carré du nombre de dimensions et du nombre d'individus

- impossible traiter grandes quantités données !

Apprentissage incrémental et parallèle

- on découpe en blocs :
  - $E^TE = \Sigma E_i^T E_i$ et $E^TDe = \Sigma E_i^T D_i e_i$
- pour obtenir finalement :

$$[w_1 w_2 w_3 \dots w_n b]^T = \left(\frac{I^o}{c} + \sum_{i=1}^{k} E_i^T E_i \right)^{-1} \sum_{i=1}^{k} E_i^T D_i e_i$$

- complexité spatiale :
  - (n+1) x b_i et b_i x b_i, b_i = taille bloc lignes x nombre dimensions
- complexité en temps :
  - distribution des calculs de blocs sur différents processeurs

Déroulement de l'algorithme

- pour i de 1 à k
  - charge blocs de données Ei et Di en mémoire CPU
  - copie Ei et Di de CPU vers GPU
  - calcule $E^TE=E^TE + E_i^TE_i$ et $E^TDe=E^TDe + E_i^TD_ie_i$ sur GPU
- fin pour

- copie E^TE et E^TDe vers CPU
- résolution équation finale

Environnement de test

- NVidia GeForce 8800 GTX :
  - 16 multiprocesseurs
  - chaque multi-processeur a 8 processeurs (total=128)
  - chaque processeur a un cache L1 et ALU
  - 768 Mo mémoire
  - 330 GFlops
  - 86Go/s bande passante mémoire
  - transfert CPU/GPU : 2Go/s

Environnement de test

- NVidia GTX - 280 :
  - 3 blocs de 10 multiprocesseurs
  - chaque multiprocesseur a 8 processeurs :
  - 240 processeurs (128)
  - 1Go mémoire (768 Mo)
  - bande passante mémoire 142Go/s (86Go/s)
  - 933 GFlops (330 GFlops)
- nVidia CUDA, API en langage C pour GP-GPU
- cublas : implémentation CUDA de BLAS : Basic Linear Algebra Subprograms

Ensembles de données

Dataset	# dimensions	Training set	Test set
Adult	110	32,561	16,281
Forest Cover Types	20	495,141	45,141
KDD Cup 1999	41	4,868,429	311,029
Ringnorm-1M	20	1,000,000	100,000
Ringnorm-10M	20	10,000,000	1,000,000

- attributs nominaux -> binaires
- 2 plus grandes classes Forest Cover Type (pour se comparer aux autres, idem pour le protocole de test)

Résultats

Dataset	Time (s)		ratio		Accuracy (%)
	GPU	CPU	CPU/GPU	SVM/GPU	
Adult	0.007	2.00	285	18150 (libSVM)	85.08
Forest Cover Types	0.06	10.00	166,67	2850 (svm-perf)	76.41
KDD Cup 1999	0.58	55.67	95.98	8189 (CB-svm)	91.96
Ringnorm-1M	0.06	3.33	55.50	N/A	75.07
Ringnorm-10M	0.61	31.67	51.92	N/A	76.68

- en moyenne GPU 130 fois plus rapide que même algorithme en version CPU !!!
- comparaison algorithmes habituels :
  - 10000 fois plus rapide que libSVM, svm-perf, CB-svm

Conclusion

- algorithme de SVM incrémental :
  - résoudre problème de mémoire
- algorithme de SVM parallèle :
  - résoudre problème de temps d'exécution
- moyenne :
  - 130 fois plus rapide que CPU
  - + de 3000 fois plus rapide algorithmes habituels

Perspectives

- utilisé une seule GPU
- *n* GPUs…
  - 10 GPUs : 30000 fois plus rapide que les algorithmes standards
  - 1 an de calcul -> ½ heure de calcul
  - aucune optimisation sur GPU…
- nouvelles applications potentielles en fouille de données…

RFO

- Random Forest (Breiman, 2001)
- construire n arbres de décision sur un sous-ensemble de dimensions / individus
- traiter grands nb dim / ind

- remplace chaque coupe arbre de décision par SVM linéaire, "arbre oblique"
- utilise un cluster de PCs (1 arbre oblique / cœur)

RFO

- comparaison avec :
  - libSVM (noyau RBF, tuning paramètres)
  - Random Forest de C4.5
- 15 ensembles données Kent Ridge Bio Medical Dataset Repository (grand nombre de dimensions)

RFO

Dataset	#pts	#dim	Classes	Test
Colon Tumor	62	2000	tumor, normal	loo
ALL-AML-Leukemia	72	7129	ALL, AML	trn-tst
*MLL-Leukemia	72	12582	MLL, rest	trn-tst
Breast Cancer	97	24481	relapse, non-relapse	trn-tst
Duke Breast Cancer	42	7129	cancer, normal	loo
Prostate Cancer	136	12600	cancer, normal	trn-tst
Lung Cancer	181	12533	cancer, normal	trn-tst
Central Nervous System	60	7129	pos, neg	loo
Translation Initiation Site	13375	927	pos, neg	10-fold
Ovarian Cancer	253	15154	cancer, normal	loo
Diffuse Large B-Cell Lymphoma	47	4026	germinal, activated	loo
*Subtypes of Acute Lymphoblastic (Hyperdip)	327	12558	Hyperdip, rest	trn-tst
*Subtypes of Acute Lymphoblastic (TEL-AML1)	327	12558	TEL-AML1, rest	trn-tst
*Subtypes of Acute Lymphoblastic (TEL-ALL)	327	12558	TEL-ALL, rest	trn-tst
*Subtypes of Acute Lymphoblastic (Others)	327	12558	others, diagnostic groups	trn-tst

RFO

Dataset	precision			recall		
	lib-SVM	RF-C4.5	RF-ODT	lib-SVM	RF-C4.5	RF-ODT
Colon Tumor	68.18	76.19	**82.61**	75.00	72.73	**86.36**
ALL-AML-Leukemia	**100**	95.24	95.24	95.00	**100**	**100**
*MLL-Leukemia	75.00	**100**	**100**	**100**	**100**	**100**
Breast Cancer	69.23	83.33	**84.62**	75.00	83.33	**91.67**
Duke Breast Cancer	85.00	**94.12**	90.00	**94.44**	80.00	90.00
Prostate Cancer	73.53	75.76	**100**	**100**	**100**	96.00
Lung Cancer	88.26	**93.75**	**93.75**	**100**	**100**	**100**
Central Nervous System	47.62	45.46	**61.91**	55.56	23.81	**61.91**
Translation Initiation Site	83.13	**92.58**	90.78	**84.42**	73.83	79.75
Ovarian Cancer	**100**	98.78	**100**	**100**	**100**	**100**
Diffuse Large B-Cell Lymphoma	91.30	**95.65**	92.00	87.50	91.67	**95.83**
*Subtypes (Hyperdip)	95.46	95.24	**100**	**95.46**	90.91	**95.46**
*Subtypes (TEL-AML1)	**100**	**100**	**100**	**100**	96.30	96.30
*Subtypes (TEL-ALL)	**100**	**100**	**100**	**100**	**100**	**100**
*Subtypes (Others)	92.59	**100**	**100**	39.68	29.63	**55.56**

RFO

Dataset	F1 Measure			accuracy		
	lib-SVM	RF-C4.5	RF-ODT	lib-SVM	RF-C4.5	RF-ODT
Colon Tumor	71.43	74.42	**84.44**	80.65	82.26	**88.71**
ALL-AML-Leukemia	97.44	**97.56**	**97.56**	**97.06**	**97.06**	**97.06**
*MLL-Leukemia	85.71	**100**	**100**	93.33	**100**	**100**
Breast Cancer	72.00	83.33	**88.00**	63.16	78.94	**84.21**
Duke Breast Cancer	89.47	86.49	**90.00**	90.48	88.10	**90.48**
Prostate Cancer	84.75	86.21	**97.96**	73.53	76.47	**97.06**
Lung Cancer	93.75	**96.77**	**96.77**	98.66	**99.33**	**99.33**
Central Nervous System	51.28	31.25	**61.91**	68.33	63.33	**73.33**
Translation Initiation Site	83.77	82.15	**84.91**	92.15	92.30	**93.20**
Ovarian Cancer	**100**	99.39	**100**	**100**	99.21	**100**
Diffuse Large B-Cell Lymphoma	89.36	93.62	**93.88**	89.36	93.62	**93.62**
*Subtypes (Hyperdip)	95.46	93.02	**97.67**	98.21	97.32	**99.11**
*Subtypes (TEL-AML1)	**100**	98.11	98.11	**100**	99.11	99.11
*Subtypes (TEL-ALL)	**100**	**100**	**100**	**100**	**100**	**100**
*Subtypes (Others)	55.56	45.71	**71.43**	64.29	83.93	**89.29**

RFO - Conclusion

- mesure F1 et précision : RFO meilleur partout sauf dans un cas (libSVM)
- plus gros gain (/libSVM) dans le cas de Breast Cancer +20% (le + gd nb de dimensions = 24481)
- coût : 10 000 mots visuels 6500 images => 75 jours cpu (100 cœurs -> - d'1 jour cpu)

Références

- F.Poulet, T-N.Do, V-H.Nguyen : *SVM incrémental et parallèle sur GPU*, in Revue des Nouvelles Technologies de l'Information, Série Extraction et Gestion des Connaissances, RNTI-E-15, Cépaduès, 2009, 103-114.
- T-N. Do, V-H. Nguyen, F. Poulet: *Speed up SVM Algorithm for Massive Classification Tasks* in proc. of ADMA'08, the 4th International Conference on Advanced Data Mining and Applications, 2008, Chengdu, China, in Advanced Data Mining and Applications, Lecture Notes in Computer Science, LNCS 5139, 147-157.
- T-N. Do, V-H. Nguyen, F. Poulet: *A Fast Parallel SVM Algorithm for Massive Classification Tasks* in proc. of MCO'08, Modelling, Computation and Optimization in Information Systems and Management Sciences, 2008, Metz, France, Communications in Computer and Information Science, CCIS-14, Springer-Verlag, 419-428.
- CUDA Programming Guide : http://www.nvidia.com/object/cuda_develop.html
- CUBLAS : http://developer.download.nvidia.com/compute/cuda/1_0/CUBLAS_Library_1.0.pdf
- nvidia 280 GTX : http://www.nvidia.com/object/product_geforce_gtx_280_us.html