

# TP4 Apprentissage

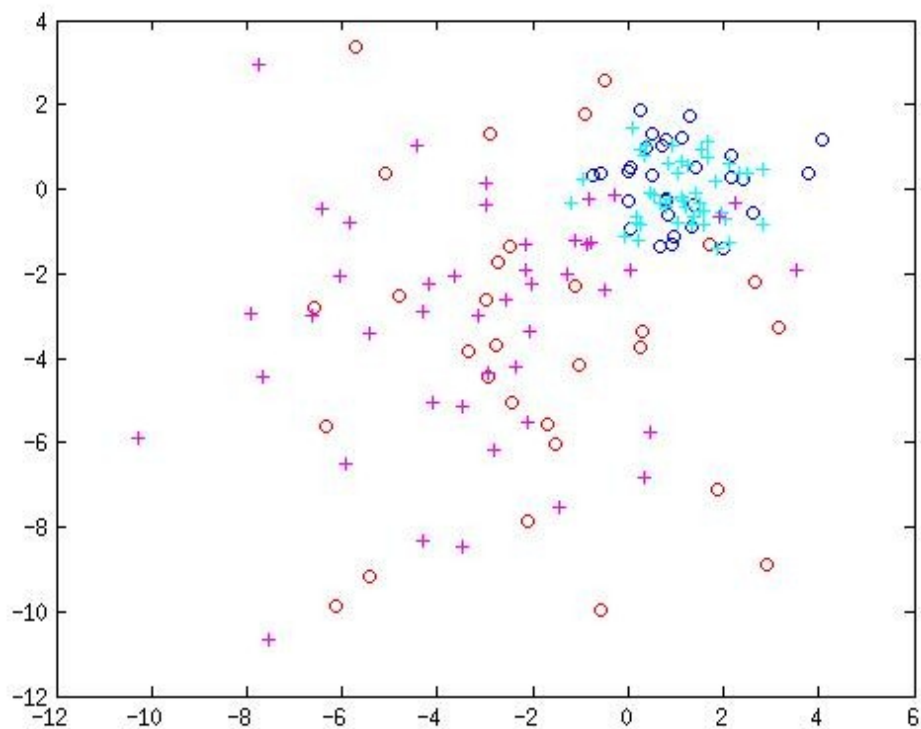
**Étudiant:** Isaías Faria

## **Comparaison PMC/SVM**

**2.1 - La base de données 1 a été généré en utilisant les commandes suivantes:**

```
function [train_pattern_b1 train_labels_b1 test_pattern_b1
test_labels_b1] = generate_base_1(train_size, test_size, Desloc,
variance)
%Database pour le train
%Primeira parte
train_pattern_1 = [ones(train_size,1) zeros(train_size,1)];
train_pattern_1 = train_pattern_1 + randn(train_size,2);
%Segunda parte
train_pattern_2 = -Desloc*[ones(train_size,1) ones(train_size,1)];
train_pattern_2=train_pattern_2 + randn(train_size,2)*variance;
%desloca a distribuicao gaussienne
%concat les 2 ensembles des points.
train_pattern_b1 = [train_pattern_1 ; train_pattern_2];
train_labels_1 = ones(train_size,1);
train_labels_2 = -1*ones(train_size,1);
train_labels_b1 = [train_labels_1 ; train_labels_2];
% base de test 1
test_pattern_1 = [ones(test_size,1) zeros(test_size,1)];
test_pattern_1=test_pattern_1 + randn(test_size,2);
%Generate les patterns 2
test_pattern_2 = -Desloc*[ones(test_size,1) ones(test_size,1)];
test_pattern_2=test_pattern_2 + randn(test_size,2)*variance;
test_pattern_b1 = [test_pattern_1 ; test_pattern_2];
%Generate les labels
test_labels_1 = ones(test_size,1);
test_labels_2 = -1*ones(test_size,1);
test_labels_b1 = [test_labels_1 ; test_labels_2];
end
```

Le centre et la variance de chaque gaussienne est variée. La figure 1 représente les données générées par la base de données 1.



**Figure 1 - Data 1 Plot**

**2.2** - À la base 2, le même code pour la préparation des données du TP précédent a été utilisé.

**3.1** - Comme les données de base 1 est aléatoire, chaque essai a été effectué trois fois, et l'erreur moyenne et la variance sont présentés dans le tableau 1. Il est à noter que, avec un plus grand nombre de couches cachées, l'erreur reste la même. La variation et le déplacement sont plus percutants que le nombre de hidden layers. Le déplacement utilisé était 2 et la variance 3. La taille du train était de 30 points et du test 45.

Nombre de hidden layers	Nombre d'erreurs	Validation croisée
<b>2</b>	<b>10</b>	<b>0,333</b>
<b>20</b>	<b>09</b>	<b>0,320</b>
<b>40</b>	<b>11</b>	<b>0,350</b>

**Tableau 1 - Variation du nombre de hidden layers.**

**3.2 - Binaire-** la première approche en utilisant la base de données USPS était de 1 vs 1. Pour la création de la base de donnée a été sélectionnée tous les images qui sont du type A et B, où A et B sont différents nombres et entre 0 et 9. Chaque vector de train X a un correspondant Y qui admet 2 seulement deux valeurs possibles.

Le tableau 2 il résult de la variation du nombre d'erreurs pour la reconnaissance des chiffres 1 et 7. Il est à noter que, avec un plus grand nombre de couches cachées, l'erreur moyenne diminue.

Nombre de hidden layers	Nombre d'erreurs	Validation croisée
<b>2</b>	<b>63</b>	<b>0,153</b>
<b>20</b>	<b>22</b>	<b>0,053</b>
<b>40</b>	<b>10</b>	<b>0,017</b>

***Tableau 2 - Variation du nombre de hidden layers por usps binaire.***

**3.2 - SVM MultiClasse** - Pour la création de la base de donnée a été sélectionnée tous les images qui sont dans le base de test proposé. Le modèle **1 vs all** a été utilisé.

Dans chaque itération de l'algorithme, un nombre est considéré comme 1 et tous les autres comme -1. Chaque fois un nombre différent est considéré comme le nombre 'actif'.

Après, un élément de X est testé dans chaque PMC **1 vs all** (il y a 10), et le modèle qui donne la plus grand valeur de Y est considéré comme le modèle qui a reconnu l'item X.

Le PMC multiclasse a été testé avec la base de test proposé. Le résultat n'était bonne et le SMV a travaillé meilleur que le PMC multiclasse.

### **Comparaison**

Par rapport à la SVM, le PMC a une plus grande erreur dans tous les tests. Mais il est plus efficace dans les exécutions.