

Quelques notes sur l'apprentissage machine

felipe@iro.umontreal.ca

RALI
Dept. Informatique et Recherche Opérationnelle
Université de Montréal



V0.001

Last compiled: 5 octobre 2018



Plan

Portraits de quelques approches

Perceptron

Une plateforme amusante

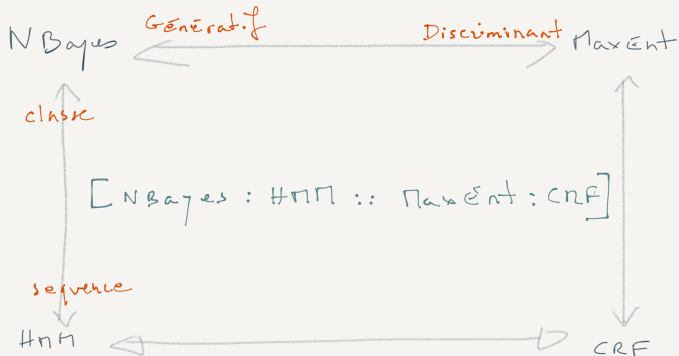
Plan

Portraits de quelques approches

Perceptron

Une plateforme amusante

Analogie quand tu nous tiens



Naive Bayes

in : $D = \{(x_1^n, y)\}$ où $x_i \in \mathcal{X}$ et $y \in \mathcal{Y}$



$$\begin{aligned} \hat{y} &= \operatorname{argmax}_{y \in \mathcal{Y}} p(y|x_1^n) \\ &\propto \operatorname{argmax}_{y \in \mathcal{Y}} p(x_1^n, y) \\ &= \operatorname{argmax}_{y \in \mathcal{Y}} p(y) \times p(x_1^n|y) \\ &= \operatorname{argmax}_{y \in \mathcal{Y}} p(y) \times \prod_i p(x_i|y) \end{aligned}$$

- ▶ si l'input est un document d , on peut considérer que x_i est booléen (présence ou non du i e mot du vocabulaire dans d) ou un scalaire (compte i e mot dans d); la jointe peut alors être représentée par une binomiale ou une multinomiale respectivement
- ▶ on peut sélectionner un sous-ensemble de mots, les lemmatiser ou non, etc. un bon baseline
- ▶ Lire [\[McCallum and Nigam, 1998\]](#) pour une application à la classification de documents



MaxEnt

in : $D = \{(x_1^n, y)\}$ où $x_i \in \mathcal{X}$, $y \in \mathcal{Y}$, et m fonctions booléennes $f_j(x, y)$

► soit $n(x, y) = \exp\left(\sum_{j=1}^m \lambda_j f_j(x, y)\right)$,

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y \in \mathcal{Y}} p_\lambda(y|x) \\ &= \operatorname{argmax}_{y \in \mathcal{Y}} \frac{n(x, y)}{\sum_{y \in \mathcal{Y}} n(x, y)}\end{aligned}$$

- entraînement (λ_j) par une variante de EM ou descente de gradient (une couche softmax sans couche cachée). Intuitivement, c'est le modèle parmi ceux *compatibles avec le corpus d'entraînement* qui a la plus grande entropie croisée : $H(y|x) = \sum_{(x,y)} p(y, x) \log p(y|x)$
- très souple, les fonctions peuvent se recouvrir en terme d'information. Lire [\[Berger et al., 1996\]](#).
- **note** : on peut simuler un naive bayes avec des fonctions $f_{w,c}(x, y)$ retournant vrai si w apparait au moins c fois dans x

HMM

in : $D = \{(x_1^n, y_1^n)\}; x_i \in \mathcal{X}, y_i \in \mathcal{Y}$

- ▶ **décodage** à l'aide de Viterbi en temps $o(E^2 \times n)$; E est le nombre d'états et n la longueur d'une séquence

$$\begin{aligned} \hat{y}_1^n &= \operatorname{argmax}_{y_1^n} p(y_1^n | x_1^n) &= \operatorname{argmax}_{y_1^n} p(y_1^n, x_1^n) \\ &= \operatorname{argmax}_{y_1^n} \prod_{i=1}^n p(y_i | y_{i-1}) \times p(x_i | y_i) \end{aligned}$$

- ▶ estimée des paramètres par fréquence relative (supervisé) ou via EM (transition = bigramme, émission = unigramme)
- ▶ cons : ne capture pas les dépendances longues
- ▶ Lire **[Rabiner, 1989]**

CRF (linéaire)

in : $D = \{(x_1^n, y_1^n)\}$; $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, K fonctions valuées $f_k \in \mathbb{R}$

- Inférence via décodage (extension de Viterbi) :

$$\begin{aligned} \hat{y}_1^n &= \operatorname{argmax}_{y \in \mathcal{Y}^n} p(y|x) \\ &= \operatorname{argmax}_{y \in \mathcal{Y}^n} \frac{1}{Z(x)} \underbrace{\exp \left(\sum_i \sum_{k=1}^K \lambda_k f_k(y_i, y_{i-1}, x, i) \right)}_{\phi(x,y)} \end{aligned}$$

où $Z(x) = \sum_{y'} \phi(x, y')$

- De très bon packages (ex : [Wapiti](#)), approche forte.
- Lire [\[Lafferty, 2001\]](#).

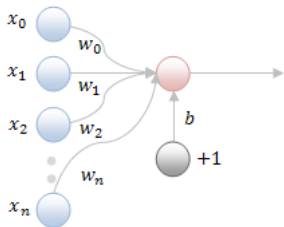
Plan

Portraits de quelques approches

Perceptron

Une plateforme amusante

Perceptron (Rosenblatt, 1957)



- ▶ x_j input (valeur réelle)
- ▶ w_j poids synaptique (valeur réelle)
- ▶ b biais (peut être représenté par un poids supplémentaire et une entrée fixe)
- ▶ $h = \sum_j x_j \times w_j + b$ est le signal arrivant au neurone
- ▶ l'activation du neurone est déterminée par une fonction d'activation $\hat{y} \equiv f(h)$ (ex : $f = \text{sign}$)

Perceptron (Rosenblatt, 1957)

- ▶ soit $(x_i, y_i)_{i \in [1, L]}$ un corpus d'entraînement où y_i (ici à valeur dans $\{-1, +1\}$) est la réponse (supervision) et $x_i \in \mathbb{R}^{n+1}$
- ▶ apprendre les w_j peut se faire **en ligne** en ajustant itérativement les poids une fois chaque observation x_i rencontrée :
$$\forall j \in [0, n], w_j \leftarrow w_j + \eta(y_i - \hat{y}_i) \cdot x_i$$
où η est le **learning rate**

résout des problèmes linéairement séparables



Voted-Perceptron [*Freund and Schapire, 1999*]

- ▶ Entraînement en ligne :

Require: $\{(\mathbf{x}_i, y_i)\}_{i \in [1, L]}$ où $y_i \in \{+1, -1\}$ et $\mathbf{x}_i \in \mathbb{R}^n$

Ensure: a pool of K perceptrons $\{(\mathbf{v}_k, c_k)\}_{k \in [1, K]}$, $\mathbf{v}_k \in \mathbb{R}^n$, $c_k \in \mathbb{N}$

$k, c_1 \leftarrow 0$

$\mathbf{v}_1 \leftarrow [0 \dots 0]^T$

for all epoch **do**

for all $i \in [1, L]$ **do**

$\hat{y} \leftarrow \text{sign}(\mathbf{v}_k \cdot \mathbf{x}_i)$

if $\hat{y} \neq y_i$ **then**

$\mathbf{v}_{k+1} \leftarrow \mathbf{v}_k + y_i \mathbf{x}_i$

$c_{k+1} \leftarrow 1$

$k \leftarrow k + 1$

else

$c_k \leftarrow c_k + 1$

Garanties de convergence (même dans les cas non linéairement séparables)

Voted-perceptron [*Freund and Schapire, 1999*]

- ▶ Test :

$$\hat{y} = \text{sign} \left(\sum_k c_k \cdot \text{sign}(\mathbf{v}_k \cdot \mathbf{x}) \right)$$

- ▶ requiert la sauvegarde de K perceptrons
- ▶ calcul en $O(K \times n)$. On peut également ne retenir qu'un sous-ensemble des perceptrons (à l'extrême un seul, par exemple celui de plus fort c_k).

Perceptron structuré [Collins, 2002]

```
w0, wa ← 0
repeat
  for all example (x, y) ∈ D do
     $\hat{y} = \operatorname{argmax}_y \mathbf{w}_0^T \Phi(x, y)$ 
    if  $\hat{y} \neq y$  then
       $\mathbf{w}_0 \leftarrow \mathbf{w}_0 + \Phi(x, y) - \Phi(x, \hat{y})$ 
       $\mathbf{w}_a \leftarrow \mathbf{w}_a + c\Phi(x, y) - c\Phi(x, \hat{y})$ 
       $c \leftarrow c + 1$ 
until converged
return  $\mathbf{w}_0 - \mathbf{w}_a/c$ 
```

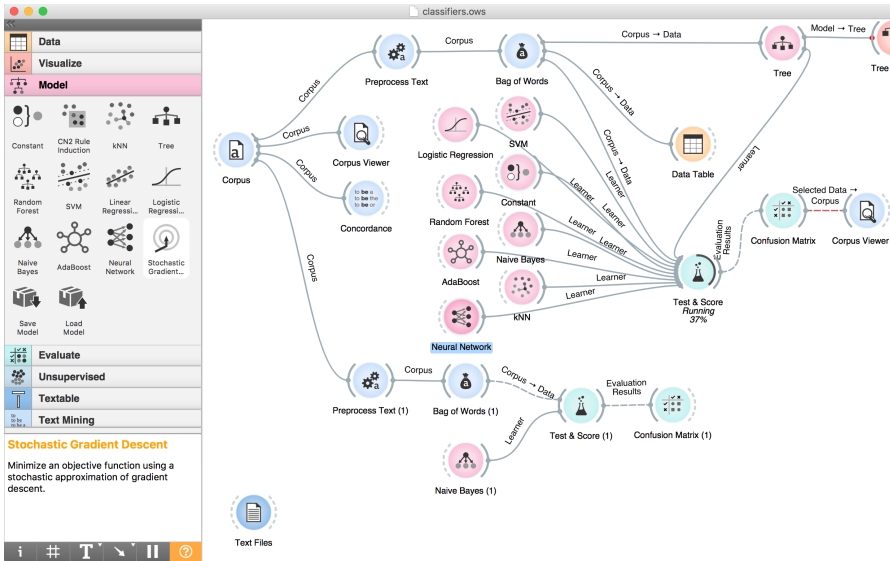
Plan

Portraits de quelques approches

Perceptron

Une plateforme amusante

Orange : une expérience



Orange : visualisation

The screenshot shows the Orange3 software interface. The main window is titled "Classifiers.ows" and displays a workflow with a "Corpus" widget connected to a "Data" widget. A "Corpus Viewer" window is open, showing a list of 100 documents and a detailed view of document 84.

Corpus Viewer Info:

- Documents: 140
- Preprocessed: False
 - Tokens: n/a
 - Types: n/a
- POS tagged: False
- N-grams range: 1-1
- Matching: 140/140

Search features:

- Category
- Text

Display features:

- Category
- Text

Show Tokens & Tags

Auto send is on

RegEx Filter:

84	Document 84
85	Document 85
86	Document 86
87	Document 87
88	Document 88
89	Document 89
90	Document 90
91	Document 91
92	Document 92
93	Document 93
94	Document 94
95	Document 95
96	Document 96
97	Document 97
98	Document 98
99	Document 99
100	Document 100

Category: adult

Text:

was beginning to increase that vague feeling of uneasiness which I always have when the Count is near But at the instant I saw that the cut had bled a little and the blood was trickling over my chin I laid down the razor turning as I did so half round to look for some sticking plaster When the Count saw my face his eyes blazed with a sort of demoniac fury and he suddenly made a grab at my throat I drew away and his hand touched the string of beads which held the crucifix It made an instant change in him for the fury passed so quickly that I could hardly believe that it was ever there Take care he said take care how you cut yourself It is more dangerous than you think in this country Then seizing the shaving glass he went on And this is the wretched thing that has done the mischief It is a foul bauble of man's vanity Away with it! And opening the window with one wrench of his terrible hand he flung out the glass which was shattered after a thousand pieces on the stones of the courtyard far below Then he withdrew without a word It is very annoying for I do not see how I am to shave unless in my watch-case or the bottom of the shaving pot which is fortunately of metal When I went into the dining room breakfast was prepared but I could not find the Count anywhere So I breakfasted alone It is strange that as yet I have not seen the Count eat or drink He must be a very peculiar man! After breakfast I did a little exploring in the castle I went out on the stairs and found a room looking towards the South The view was magnificent and from where I stood there was every opportunity of seeing it The castle is on the very edge of a terrific precipice A stone falling from the window would fall a thousand feet without touching anything! As far as the eye can reach is a sea of green tree tops with occasionally a deep rift where there is a chasm Here and there are silver threads where the rivers wind in deep gorges through the forests But I am not in heart to describe beauty for when I had seen the view I explored further Doors doors doors everywhere and all locked and bolted In

Stochastic Gradient Descent

Minimize an objective function using stochastic approximation of gradient descent.



Text Files

Orange : concordancier

Orange Data Mining interface showing the workflow and model selection options.

Data

Visualize

Model

- Constant
- CN2 Rule Induction
- kNN
- Tree
- Random Forest
- SVM
- Linear Regress...
- Logistic Regress...
- Naive Bayes
- AdaBoost
- Neural Network
- Stochastic Gradient...
- Save Model
- Load Model

Evaluate

Unsupervised

Textable

Text Mining

Stochastic Gradient Descent

Minimize an objective function using a stochastic approximation of gradient descent.

Orange workflow diagram showing the process of text preprocessing and classification.

Workflow: Corpus → Preprocess Text → Bag of Words → Concordance → Naive Bayes (1) → Test & Score (1) → Confusion Matrix (1)

Concordance window details:

Info

- Tokens: 133484
- Types: 11744
- Matching: 43/140

Number of words: 10

Query: dear

1	... eyes were closed and his face a horrible colour	Dear	deary me cried my mother what a
2	...that one of the newcomers carried a lantern My	dear	said my mother suddenly take the
3	news : Old Anchor Inn Bristol March 1 17 --	Dear	Livesey -- As I do not know wheth
4	where I had lived since I was born and the	dear	old Admiral Benbow -- since he w
5	...ow -- since he was repainted no longer quite so	dear	One of my last thoughts was of the
6	...g out shod in silver shoes with pointed toes Oh	dear	! Oh dear ! cried Dorothy clasping
7	in silver shoes with pointed toes Oh dear ! Oh	dear	! cried Dorothy clasping her hands
8	surrounds this Land of Oz ! ' m afraid my	dear	you will have to live with us Doroth
9	the words on it asked Is your name Dorothy my	dear	? Yes answered the child looking u
10	and ask him to help you Good - bye my	dear	The three Munchkins bowed low to
11	no idea of looking at anything and still slept on	Dear	me ! exclaimed Polly in consternat
12	...d Polly pointing tragically to the little heap Well	dear	me ! said Jasper Why Polly -- as h
13	... looking up into his face Indeed I am Grandpapa	dear	very hungry Oh to think of it ! Yes
14	the bedclothes I wish you ' d take me Grandpapa	dear	she said holding up her arms So I
15	...r he walked up and down the room There there	dear	! Oh why doesn ' t that Sarah hurry
16	... t mean to be didn ' t mean to Oh	dear	me ! exclaimed old Mr King in dism

Text Files

Orange : neural network

The screenshot displays the Orange3 software interface with a workflow for text classification. The workflow consists of the following widgets and connections:

- Corpus** (Data source) connects to **Preprocess Text** and **Bag of Words**.
- Preprocess Text** connects to **Bag of Words**.
- Bag of Words** connects to **Neural Network** and **Tree**.
- Neural Network** connects to **Evaluation Results**.
- Tree** connects to **Evaluation Results**.
- Naive Bayes** connects to **Evaluation Results**.
- Evaluation Results** connects to **Confusion Matrix**.
- Text Files** widget is also present at the bottom.

The **Neural Network** widget settings are as follows:

- Name: Neural Network
- Network: Neurons per hidden layer: 100
- Activation: ReLu
- Solver: Adam
- Alpha: 0,00010
- Max iterations: 200
- Apply Automatically:

The **Evaluation Results** widget shows: **Test & Score Running 51%**.

Stochastic Gradient Descent

Minimize an objective function using a stochastic approximation of gradient descent.

Orange : résultats

Test & Score

Sampling

- Cross validation
 - Number of folds: 10
 - Stratified
 - Cross validation by feature
- Random sampling
 - Repeat train/test: 10
 - Training set size: 66 %
 - Stratified
 - Leave one out
 - Test on train data
 - Test on test data

Target Class

(Average over classes)

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
kNN	0.955	0.921	0.921	0.922	0.921
Tree	0.779	0.779	0.778	0.779	0.779
SVM	0.996	0.957	0.957	0.957	0.957
Random Forest	0.902	0.843	0.843	0.843	0.843
Neural Network	0.998	0.979	0.979	0.979	0.979
Naive Bayes	0.879	0.843	0.839	0.880	0.843
Logistic Regression	0.994	0.950	0.950	0.950	0.950
Constant	0.500	0.500	0.495	0.500	0.500
AdaBoost	0.793	0.793	0.793	0.794	0.793

Test & Score

Cross-validation accuracy estimation.

[more...](#)

Text Files

Orange : matrice de confusion

The screenshot displays the Orange3 software interface. On the left, a sidebar contains various widgets categorized into Data, Visualize, Model, Evaluate, Unsupervised, Textable, and Text Mining. The main workspace shows a workflow with the following widgets: Preprocess Text (1), Corpus, Bag of Words (1), Naive Bayes (1), Neural Network, kNN, Corpus → Data, Test & Score (1), Evaluation Results, and Confusion Matrix (1). A 'Confusion Matrix' window is open in the foreground, showing the results of a classifier evaluation.

Confusion Matrix Window:

classifiers.ows

Corpus → Data

Confusion Matrix

Show: Number of instances

		Predicted		Σ
		adult	children	
Actual	adult	59	11	70
	children	11	59	70
Σ		70	70	140

Output

Predictions Probabilities

Send Automatically

Select Correct Select Misclassified Clear Selection

Confusion Matrix

Display a confusion matrix constructed from the results of classifier evaluations.

[more...](#)

Text Files



Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996).

A maximum entropy approach to natural language processing.
Comput. Linguist., 22(1) :39–71.



Collins, M. (2002).

Discriminative training methods for hidden markov models : Theory and experiments with perceptron algorithms.

In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02*, pages 1–8.



Freund, Y. and Schapire, R. E. (1999).

Large margin classification using the perceptron algorithm.
Mach. Learn., 37(3) :277–296.



Lafferty, J. (2001).

Conditional random fields : Probabilistic models for segmenting and labeling sequence data.

In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann.



McCallum, A. and Nigam, K. (1998).

A comparison of event models for naive Bayes text classification.

In *Learning for Text Categorization : Papers from the 1998 AAAI Workshop*, pages 41–48.



Rabiner, L. R. (1989).

A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, chapter 6.

IEEE.