

Memo pour l'année

LAURENT Thomas

Master 2 informatique 2018

Contents

1	Fouille de donnée	1
1.1	Pré traitement des données	2
1.1.1	Nettoyage des données	2
1.1.2	Normalisation	2
1.2	Classification	3
1.2.1	Évaluation des classifieurs	3
1.3	Arbre de décision	4
1.3.1	critères de sélection C4.5	4
2	Apprentissage par le pratique	9
2.1	Rappel	10
2.1.1	Matrices et calculs sur les Matrices	10
2.2	Algorithms Learn a Mapping From Input to Output	11
2.2.1	linear ML algorithms	11
2.2.2	Supervised machine learning	11
2.2.3	Unsupervised machine learning	11
2.2.4	semi-supervised machine leaning	11
2.2.5	Overview of bias and variance	12
2.3	Overfitting and Underfitting	13
2.4	Linear Algorithms	14
2.4.1	Régression linéaire	14
2.4.2	Least squares linear regression	15
2.4.3	Gradient Descent	16
2.5	Logistic Regression	17
2.5.1	Logistic function	17
2.5.2	Logistic regression predicts probabilities	17

3	Outils formel	18
3.1	Logique classique des propositions	19
3.1.1	Vocabulaire	19
3.1.2	Propriétés de l'opérateur Models	19
3.1.3	Ensemble de connecteurs fonctionnellement complet . .	20
3.1.4	Preuve par induction structurelle sur un ensemble de connecteurs non fonctionnellement complet	21
3.1.5	Décomposition de Shannon	21
3.1.6	Arbre de Shannon, ROBDD	22
3.1.7	Notion de impliquant premier	23
3.1.8	Système de Hilbertien	24
3.1.9	Forte complétude OU théorème de finitude	24
4	Représentation des connaissances et raisonnement	25
5	Recherche Opérationnel	27
5.1	Rappel	28
5.1.1	Pivot de gauss	28
5.2	Introduction à la PL	28
5.2.1	Modèle linéaire continu à 2 variables	28
6	XML	31

Chapter 1

Fouille de donnée

1.1 Pré traitement des données

1.1.1 Nettoyage des données

Caractéristiques descriptives

Objectifs: Résumer, décrire certains aspects (tendances, variation, dispersion...) des données en utilisant certaines mesures :

Moyenne (espérance) : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

Ecart moyen : $\frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$

Variance : $v = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$

Ecart type : $\sigma_x := \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\frac{1}{n} (\sum_{i=1}^n x_i^2) - \bar{x}^2}$

Médiane : Valeur se trouvant au milieu d'une série de données ordonnées

Mode : Valeur la plus fréquente

Amplitude : min, max

1.1.2 Normalisation

Min-max : $v_n = \frac{v - v_{min}}{v_{max} - v_{min}}$

Min-max dans l'intervalle [A,B] : $v_n = \frac{v - v_{min}}{v_{max} - v_{min}} * (B - A) + A$

Z-Score : $v_n = \frac{v - moyenne}{ecart_{type}}$

Decimal scaling : $v_n = \frac{v}{100^j}$

1.2 Classification

1.2.1 Évaluation des classifieurs

Matrice de confusion

Percent of correct classification :

$$\text{PCC}(\%) := \frac{N_c}{N_t} * 100$$

N_c : nombre d'instances correctement classées

N_t : nombre d'instances testées ($N_t = |D_{test}|$)

Exemple:

$$: \begin{pmatrix} - & c1 & c2 & c3 & c4 \\ c1 & 0 & 1 & 0 & 0 \\ c2 & 1 & 60 & 0 & 1 \\ c3 & 0 & 1 & 23 & 0 \\ c4 & 1 & 0 & 7 & 5 \end{pmatrix}$$

Taux d'erreurs : 100-PCC

$$\text{PCC}(\%) = \frac{0+60+23+5}{100} * 100 = 88\%$$

1.3 Arbre de décision

1.3.1 critères de sélection C4.5

Construction d'un arbre de décision C4.5 La construction d'un arbre de décision avec C4.5 passe par deux phases:

Phase d'expansion : La construction se fait selon l'approche descendante et laisse croître l'arbre jusqu'à sa taille maximale.

Phase d'élagage : Pour optimiser la taille l'arbre et son pouvoir de généralisation, C4.5 procède à l'élagage (pour supprimer les sous-arbres qui ne minimisent pas le taux d'erreurs)

Approche de construction d'un AD : Partitionner récursivement les données en sous-ensembles plus homogènes ... jusqu'à obtenir des partitions qui contiennent des objets qui appartiennent majoritairement à la même classe.

=> Théorie de l'information pour caractériser le degré de mélange, homogénéité, impureté, incertitude...

Théorie de l'information : Théorie mathématique ayant pour objet l'étude du contenu informationnel d'un message.

Applications en codage, compression, sécurité...

Entropie : Mesure la quantité d'incertitude dans une distribution de probabilités.

Rappel sur les probabilités

Quelques rappels de probabilités : Soient X et Y deux variables aléatoires discrètes prenant leurs valeurs dans $DX=x_1,\dots,x_n$ et $DY=y_1,\dots,y_m$ respectivement.

$$P(x_i) = \frac{|x_i|}{\sum_{j=1}^n |x_j|}$$

$$\sum_{i=1}^n P(x_i) = 1$$

$$P(x_i|y_i) = \frac{P(x_i, y_i)}{p(y_i)}$$

$$P(x_i, y_i) = p(x_i) * p(y_i) \text{ Si X et Y sont indépendantes}$$

Exemple:

$$: \begin{pmatrix} Anne & Sexe & \# & \% \\ M1 & M & 25 & 25/55 \\ M1 & F & 4 & 4/55 \\ M2 & M & 25 & 25/55 \\ M2 & F & 1 & 1/55 \end{pmatrix}$$

$$P(sexe = M) = P(Sexe = MetAnne = M1) + P(Sexe = MetAnne = M2) = 50/55$$

$$P(Anne = M2 | sexe = M) = P(Sexe = MetAnne = M2) / P(Sexe = M) = \frac{25}{55} / \frac{50}{55} = \frac{25}{50} = \frac{1}{2}$$

Entropie

Entropie : Mesure la quantité d'incertitude (manque d'information) dans une distribution de probabilités. Soit X une variable aléatoire discrète prenant ses valeurs dans $DX = x_1, \dots, x_n$. Soit P la distribution de probabilités associée à X .

$$H(X) = - \sum_{i=1}^n p(x_i) * \log_2(p(x_i))$$

Par convention, quand $p(x) = 0, 0 * \log(0) = 0$

Exemple:

X	P(X)
x_1	1/3
x_2	1/3
x_3	1/3

$$H(X) = -p(x_1) * \log_2(p(x_1)) - p(x_2) * \log_2(p(x_2)) - p(x_3) * \log_2(p(x_3))$$

$$H(X) = -3(\frac{1}{3} * \log_2(\frac{1}{3})) = \log_2(3) = 1.58$$

Autre exemples:

$$[\frac{1}{2}, \frac{1}{4}, \frac{1}{4}] : H(X) = 1.5$$

$$[1, 0, 0] : H(X) = 0$$

$$[\frac{1}{2}, \frac{1}{2}] : H(X) = 1$$

Propriétés:

$$H(X) \geq 0$$

$H(X)$ est maximale pour une distribution uniforme (toutes les valeurs sont équiprobables).

Entropie conjointe : L'entropie conjointe de deux variables aléatoires X et Y est l'incertitude relative à ces deux variables conjointement.

$$H(X, Y) = - \sum_{i,j=1}^n p(x_i, y_j) * \log_2(p(x_i, y_j))$$

Exemple : $[0.2, 0.1, 0.3, 0.4] : H(X, Y) = 1.85$

Critère de sélection: Gain d'information:

$$GAIN(T, A) = Info(T) - Info(T|A)$$

Avec $Info(T)$: Entropie au niveau de T (avant de partitionner)

$$Info(T) = - \sum_{c_i} freq(c_i, T) * \log_2(freq(c_i, T))$$

$$\textbf{Avec } freq(c_i, T) = p(c_i) = \frac{|c_i|}{|T|}$$

Avec $Info(T|A)$ l'entropie conditionnelle de T une fois partitionné selon les valeurs de l'attribut A.

$$Info(T|A) = \sum_{a_j \in A} freq(a_j, T) * Info(T|a_j)$$

Critère de sélection: Gain Ration:

Le gain d'information favorise les attributs ayant de larges domaines.

Le ratio de gain utilise le gain d'information avec un facteur pénalisant les attributs ayant des domaines trop larges.

$$GainRatio(T, A) = \frac{Gain(T, A)}{SplitInfo(T, A)}$$

$$\textbf{Avec } SplitInfo(T, A) = - \sum_{a_j \in A} freq(a_j, T) * \log_2(freq(a_j, T)) = EntropiedeA$$

j

Chapter 2

Apprentissage par le pratique

2.1 Rappel

2.1.1 Matrices et calculs sur les Matrices

Addition

$$\begin{pmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1+0 & 3+0 \\ 1+7 & 0+5 \\ 1+2 & 2+1 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 8 & 5 \\ 3 & 3 \end{pmatrix}$$

Multiplication

$$\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$
$$(1 * 5) + (2 * 7) = 19$$

Transposer

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

Inverse

Soit une matrice 2x2 comme : $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Soit Determinant $D = ad - bc$

Si $D \neq 0$ alors il existe une matrice inverse égal à : $\frac{1}{D} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

2.2 Algorithms Learn a Mapping From Input to Output

2.2.1 linear ML algorithms

Simplifier les processus d'apprentissage et réduire la fonction sur ce qu'on connaît

Soit : $B_0 + B_1X_1 + B_2X_2 + B_3X_3 = 0$

Où B_0, B_1, B_2, B_3 sont les coefficients présent sur l'axe des ordonnées.

Et X_1, X_2, X_3 sont les valeurs en Input.

2.2.2 Supervised machine learning

L'apprentissage supervisé peut se diviser en 2 partis

Classification : Quand les variables en sortie sont des Classe (*Vert, Carr, Homme*)

Regression : Quand les variables en sortie sont des valeur numérique (*euro, poids, quantités*)

2.2.3 Unsupervised machine learning

Les problèmes de l'apprentissage non supervisé sont:

Clustering : L'art de faire des paquet d'éléments qui ont des points commun, comme regrouper les clients par paquet de choses qu'ils ont le plus en commun.

Association : Associer des règles d'apprentissage pour décrire une portion du data, comme une personne qui a acheté un item A et qui est aussi tenté par acheter un item B

2.2.4 semi-supervised machine leaning

L'apprentissage semi supervisé c'est avoir un bonne quantité de données en input X, et un peu de data avec le label Y.

2.2.5 Overview of bias and variance

La prédiction des erreurs pour les algorithmes sont regroupé en 3 points:

Bias Error : Simplifier l'hypothèse fait par le modèle pour faire une fonction d'apprentissage plus facile.

Variance Error : Et la quantité estimée par la fonction visée qui changera via un différent ensemble de data utilisé.

Irreducible Error : Ne peut pas être réduit

2.3 Overfitting and Underfitting

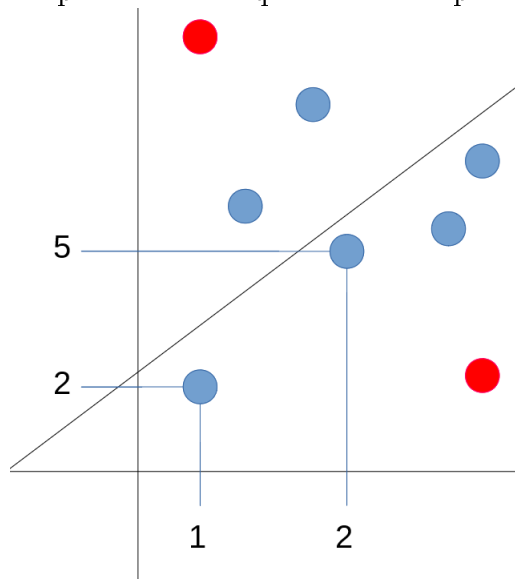
ddddddddd

2.4 Linear Algorithms

Soit X l'ensemble des variables indépendantes sur l'axe des l'abscisse et Y l'ensemble des variable dépendantes sur l'axe des ordonnée.

2.4.1 Régression linéaire

Étant donné un plan à deux dimensions où l'abscisse contient les point d'entrée X et l'ordonnée contient les points de sortie Y, et un nuage de points précédaiant acquitté de tout point éloigné du nuage.



Figureap – linear – regression₁

Avec : $y = \beta_0 + \beta_1 x$

Pour un hyperPlan (3d) : $y = \beta_0 + \beta_1 x_1 + \dots \beta_n x_n$

Exemple:

$$5 = \beta_0 + 2 * \beta_1$$

$$2 = \beta_0 + 1 * \beta_1$$

2.4.2 Least squares linear regression

Calculer la régression linéaire avec la méthode Least squares:

Soit:

$\mathbf{X} = [1, 2, 3, 4, 5]$ les variables indépendantes d'axe abscisse

$\mathbf{Y} = [2, 4, 5, 4, 5]$ les variables dépendantes d'axe ordonnée

Calculons $y = \beta_0 + \beta_1 x$

Calcule de la moyenne de X et Y:

$$\mathbf{Xm} = \sum x_i \in X = 3$$

$$\mathbf{Ym} = \sum y_i \in Y = 4$$

Toutes ligne de régression doivent passer par le point (Xm,Ym).

Calculer tout les écarts des $x_i \in X$ par rapport à Xm (resp Y):

X	Y	$X - Xm$	$Y - Ym$	$(X - Xm)^2$	$(X - Xm)(Y - Ym)$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2

Calculer β_1 :

$$\beta_1 = \frac{\sum (X - Xm)(Y - Ym)}{\sum (X - Xm)^2} = \frac{6}{10} = .6$$

$$\beta_0 : Ym = \beta_0 + \beta_1 * Xm : 4 = \beta_0 + .6 * 3 : 4 = \beta_0 + 1.8 : \beta_0 = 2.2$$

2.4.3 Gradient Descent

Soit:

$$\mathbf{X} = [1, 2, 4, 3, 5]$$

$$\mathbf{Y} = [1, 3, 3, 2, 5]$$

i = une variable qui itère les éléments de X et Y en bouclant à l'infini.

Une initialisation comme:

$$\beta_0 = 0$$

$$\beta_1 = 0$$

α = donnée en énoncé (pour l'exemple égal à 0.01)

Et des fonctions définit tel que:

$$\mathbf{error} = (\beta_0 + \beta_1 * X[i]) - Y[i]$$

$$\beta_{0+1} = \beta_0 - \alpha * error$$

$$\beta_{1+1} = \beta_1 - \alpha * error * X[i]$$

En appliquant l'algorithme des calculs des β_i :

i	$X[i]$	$Y[i]$	$error$	β_0	β_1
0	1	1	-1	0.01	0.01
1	2	3	-2.97	0.06	0.03
2	4	3	-1.77	0.18	0.06
3	3	2	-1.61	0.22	0.08
4	5	5	-4.35	0.44	0.12
0	1	1	-0.42	0.45	0.13
1	2	3	-2.28	0.49	0.49

2.5 Logistic Regression

2.5.1 Logistic function

Soit:

$$\mathbf{t} \in \mathbb{R}[0, 1] \text{ égal à } \beta_0 + \beta_2 * x$$

La fonction de logique de régression, les valeur d'entrée X sont combiné en utilisant les coefficient de valeur pour prédire une sortie Y. Cette sortie sera une valeur binaire.

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1*x)}}$$

Note : $p(x)$ peut être interprété comme une fonction de probabilité $P(X) = P[Y = 1|X]$.

$$\beta_0 + \beta_1 * x = \ln\left(\frac{P(x)}{1-P(x)}\right) \text{ aussi appelé odds.}$$

2.5.2 Logistic regression predicts probabilities

Chapter 3

Outils formel

3.1 Logique classique des propositions

3.1.1 Vocabulaire

Déduction $\models \alpha$ ssi $\neg\alpha$ est contradictoire

Absurde ϕ est contradictoire ssi $\neg\phi$ est valide

DAG : Un graphe dirigé acyclique

Taille(Arbre) = $\{\text{tout les symboles} + \text{connecteurs}\}$

Var(Arbre) = $\{\text{Toutes les feuilles}\}$

Sous formules(Arbres) = $\{T + \cup_{i=0}^k \text{SousFormules}(\text{Arbre}_i)\}$

Interprétation : ω de $PROP_{ps}$ est une application de PS dans 0.1

Sémantique : $[[\phi]](\omega)$ d'une formule ϕ de $PROP_{ps}$ dans l'interprétation ω est un élément de 0.1 défini inductivement par:

si $\phi \in PS$ alors $[[\phi]](\omega) = \omega(\phi)$

si $\phi = cX_1...X_n$ alors $[[\phi]](\omega) = C_F([x_1]](\omega)...[x_n]](\omega))$

ω **satisfait** ϕ noté $\omega \models \phi$ ssi $[[\phi]](\omega) = 1$

Lorsque $\omega \models \phi$ on dit que ω est un modèle de ϕ

on note $\eta(\phi)$ l'ensemble des modèles de ϕ

$\omega \in PROP_{ps}$ **est valide** noté $\models \phi$, ssi toute interprétation ω de $PROP_{ps}$ satisfait ϕ

$\phi \equiv \psi$ sont logiquement équivalents ssi $\phi \models \psi$ et $\psi \models \phi$

3.1.2 Propriétés de l'opérateur Models

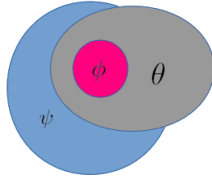
$a \models b \iff M(a) \subseteq M(b)$

Réflexivité : $\phi \models \phi$

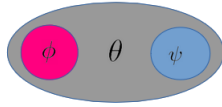
Équivalence à gauche : si $\phi \equiv \theta$ et $\phi \models \psi$ alors $\theta \models \psi$

Affaiblissement à droite (transitivité) : si $\phi \models \psi$ et $\psi \models \theta$ alors $\phi \models \theta$

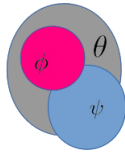
Coupure : si $\phi \wedge \psi \models \theta$ et $\phi \models \psi$ alors $\phi \models \theta$: $== (A \cup B) \subseteq C \text{ si } A \subseteq C \cap B \subseteq C$



Ou : $\phi \vee \psi \models \theta$ si $\phi \models \theta$ et $\psi \models \theta$



Monotonie : si $\phi \models \theta$ alors $\phi \wedge \psi \models \theta$



3.1.3 Ensemble de connecteurs fonctionnellement complet

On dit qu'un ensemble est fonctionnellement complet si avec que les connecteurs de cette ensemble on peut exprimer toutes les formules d'un monde.

$\{\neg, \wedge\}$ est fonctionnellement complet pour la logique propositionnel classique

Il en va de même pour $\{\neg, \vee\}, \{\text{vrai}, \wedge, \oplus\}, \{\neg, \Rightarrow\}$ ou $\{NAND\}$

Suppression des fils équivalent : Soit un arbre D ayant comme sous arbre plus d'une fois le nœud $\alpha = (\top X \top)$, α peut être remplacé par (\top) tout en concevant les modèles de D.

fusion des nœuds : Soit un arbre D ayant comme sous arbre les nœuds (aBc) et $(a'B'c')$ et $a = a', b = b', c = c'$ alors on peut faire relier les deux branches menant vers ces nœuds vers le même sous arbre.

3.1.4 Preuve par induction structurelle sur un ensemble de connecteurs non fonctionnellement complet

Soit $\forall P \in \{\wedge, \vee\}_{ps}$, vérifier P:

Cas de base $\varphi \in PS$: $1^\rightarrow(\varphi) = 1$ donc 1^\rightarrow constitue un modèle de φ

Étape inductive :

φ s'écrit : $[\alpha \wedge \beta]$ ou $[\alpha \vee \beta]$

Avec $\alpha, \beta \in \{\wedge, \vee\}_{ps}$

Par hypothèse d'induction, α et β vérifient P.

Il ne reste plus qu'à montrer que φ vérifie P.

$$[[\alpha \vee \beta]](1^\rightarrow) = \vee \models ([[\alpha]](1^\rightarrow), [[\beta]](1^\rightarrow)) = \vee \models (1, 1) = 1$$

$$[[\alpha \wedge \beta]](1^\rightarrow) = \wedge \models ([[\alpha]](1^\rightarrow), [[\beta]](1^\rightarrow)) = \wedge \models (1, 1) = 1$$

donc $x \wedge \neg x$ ne vérifie pas P : $[[x \wedge \neg x]](1^\rightarrow) = 0$

3.1.5 Décomposition de Shannon

On note $\phi[x \leftarrow 0]$ la formule obtenue en substituant dans ϕ la constante faux à toutes les occurrences du symbole propositionnel x.

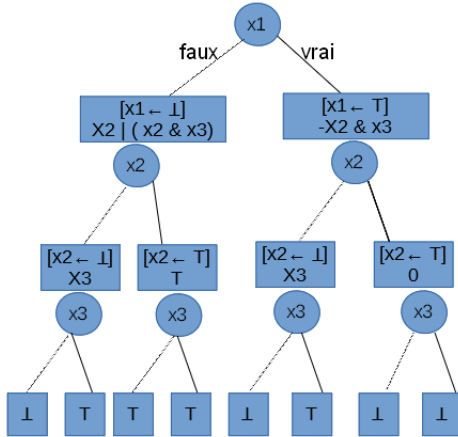
On note $\phi[x \leftarrow 1]$ la formule obtenue en substituant dans ϕ la constante vrai à toutes les occurrences du symbole propositionnel x.

La décomposition de Shannon de ϕ suivant x est la formule:

$$(\neg x \wedge \phi[x \leftarrow 0]) \vee (x \wedge \phi[x \leftarrow 1])$$

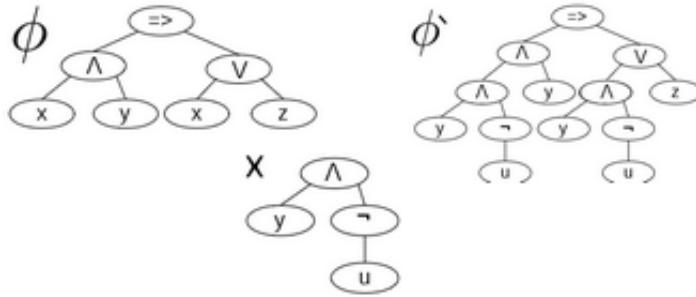
3.1.6 Arbre de Shannon, ROBDD

Étant donnée un ordre strict total $x_1 < x_2 < x_3$ sur $Var(\phi) = \{x_1, \dots, x_n\}$
 Et une formule $\phi = (\neg x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3)$



L'ensemble des modèles de ϕ sont toutes les interprétation où la feuille vaut la valeur T .

Remplacement ou vérifonctionnalité



$\phi \equiv \phi'$ quelque soit la valeur de x (vrai ou faux).

Substitution

Soit un arbre D ayant comme nœud un sous arbre du type infixe $\alpha = (x \Rightarrow y)$
 et un sous arbre de substitution $\beta = (\neg x \Rightarrow \neg y)$
 $(D' = D_{\alpha \leftarrow \beta} \equiv D)$

3.1.7 Notion de impliquant premier

Les impliquant premier sont des sous formules des formules original tel que ces sous formules soit plus petite que la formule d'origine elle conserve les même modèles:

En circuit combinatoire les algo sont appelé Table de Karnaugh ou Quine-McCluskey.

Table de Karnaugh

Appliquer l'algorithme avec la formule $S = \neg ab \neg cd + a \neg b \neg c \neg d + b \neg d$

S	$\neg a \neg b$	$\neg ab$	ab	$a \neg b$
$\neg c \neg d$	X	X	X	X
$\neg cd$		X	X	
cd		X	X	
$c \neg d$	X	X	X	X

les impliquant premier de S sont $b \neg d$

Calcule arithmétique

En logique, les impliquant premier sont calculer que à partir d'une formule en mode CNF transposé en DNF et ensuite détransposé en CNF.

$$\phi = (a \wedge b \wedge c) \vee (\neg b \wedge c)$$

$$\phi = (a \vee \neg b) \wedge (a \vee c) \wedge (b \vee \neg b) \wedge (b \vee c) \wedge (c \vee \neg b) \wedge (c \vee c)$$

$$\phi = (a \vee \neg b) \wedge (a \vee c) \wedge (b \vee c) \wedge (c \vee \neg b) \wedge c$$

$$\phi = (a \vee \neg b) \wedge c$$

$$\phi = (a \wedge c) \vee (\neg b \wedge c) \text{ sont les impliquant premier.}$$

Via une table de Karnaugh:

ϕ	$\neg a \neg b$	$\neg ab$	ab	$a \neg b$
$\neg c$				
c	X		X	X

Égal à $(a \wedge c) \vee (\neg b \wedge c)$.

3.1.8 Système de Hilbertien

gg

3.1.9 Forte complétude OU théorème de finitude

g

Chapter 4

Recherche Opérationnel

4.1 Rappel

4.1.1 Pivot de gauss

$$\begin{aligned} L2 \text{ et } L2 &= \begin{cases} L1 : 160 = 8x + 4y \\ L2 : 120 = 4x + 6y \end{cases} \\ (L2 * (-2)) &= \begin{cases} L1 : 160 = 8x + 4y \\ L2 : -240 = -8x - 12y \end{cases} \\ (L2 = L2 + L1) &= \begin{cases} L1 : 160 = 8x + 4y \\ L2 : -80 = -8y \end{cases} \\ y &= 10 \end{aligned}$$

$$8x + 4 * 10 = 160$$

$$8x + 40 = 160$$

$$8x = 120$$

$$x = 15$$

4.2 Introduction à la PL

Construire une modèle linéaire, c'est donc:

identifier les variables de décision du problème

déterminer : la fonction objectif du modèle

déterminer : les contraintes du modèle

4.2.1 Modèle linéaire continu à 2 variables

Soit le modèle linéaire suivantes:

Déterminer $(x, y) \in \mathfrak{S}^2$

Minimisant $z = 1000x + 1200y$

sous les contraintes :

$$(1) 8x + 4y \leq 160$$

$$(2) 4x + 6y \leq 120$$

$$(3) x \leq 34$$

$$(4) y \leq 14$$

$$(5) 0 \leq x$$

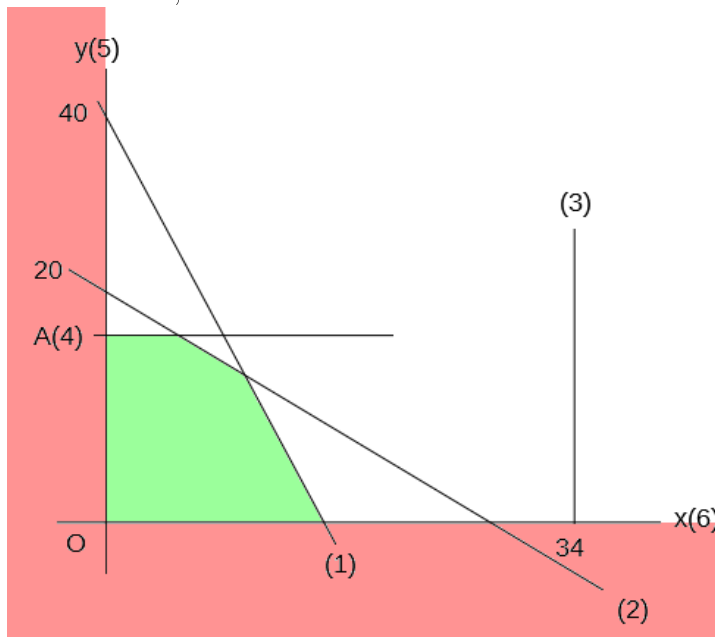
$$(6) 0 \leq y$$

Recherche de solutions

Après avoir tracé graphiquement tout les points:

Pour chaque contrainte, tracer la droite et repérer le demi plan des solution: exemple pour (5) et (6), x et y doivent être supérieurs ou égal à 0, d'où le demi plan des solution sont toutes les valeurs positives.

La partie En vert représente la région admissible, quelque soit le point choisis dans ce vert, aucune contrainte ne sera violé.



recherche de la solution optimal

Changer l'équation z tel que z soit égal à 0

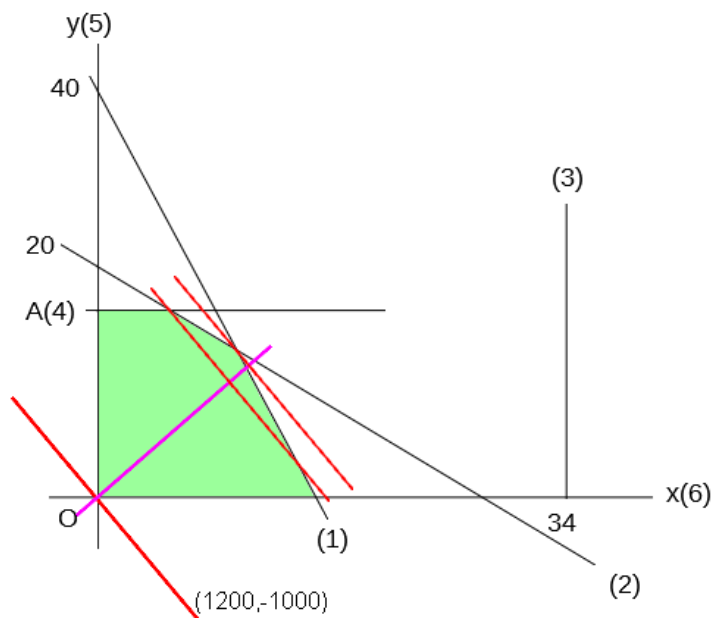
$$z = 1000x + 1200y = 0 = 1000 * (1200) + 1200 * (-1000)$$

Traçons la droite $(0, 0)$, $(1200, -1000)$

Un point extrême : est un point se trouvant sur l'intersection de 2 contraintes et étant dans la zone admissible.

L'altitude : est la droite (rouge) la plus haute touchant un point extrême, ce point sera le vecteur (x, y) le plus optimal pour z .

Les droites rouges doivent être toutes parallèles.



Dans cette exemple le point $(15, 10)$ est le point extrême maximal pour l'équation z .

Chapter 5

Représentation des connaissances et raisonnement

ggggg

Chapter 6

XML

uuuuu