

MAINTENANCE PRÉDICTIVE

—
08/02/2024



Plan du cours

- Introduction
- Modélisation statistique pour la maintenance
- Capteurs, analyse de données pour la maintenance
- Nettoyage et manipulation des données
- Analyse des défaillances pour la maintenance

Introduction

▪ Rappels des cours précédents

> Cours 1 & 2 :

- Pour optimiser la maintenance, on a besoin (entre autre) d'estimer les lois de vie des équipements
- Les méthodes statistiques comme Weibull modélisent ces lois de vie
- Des méthodes spécifiques traitent les problématiques des covariates

> Cours 3 :

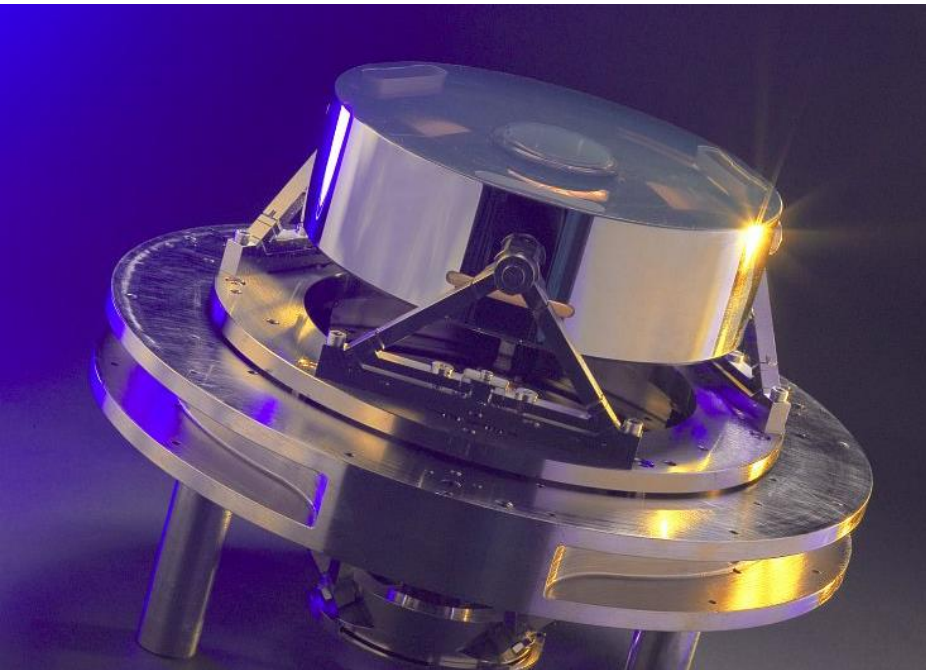
- Désormais, on récolte des données de plus en plus nombreuses, et tout au long de la vie des équipements
- On peut également prédire la durée de vie restante par des méthodes d'apprentissage plus courantes

> Cours 4 :

- Les techniques de traitement de données sur des problématiques classiques :
 - ♦ Le nettoyage des données
 - ♦ La sélection de données
 - ♦ Le traitement des données catégorielles
- De nouvelles méthodes émergent très récemment, pour mixer les approches des cours 2 & 3

> Cours 5 :

- L'analyse des pannes, ou troubleshooting en anglais
 - ♦ L'AMDEC
 - ♦ L'arbre de défaillance
 - ♦ Les réseaux bayésiens



ANALYSE DES MODES DE DÉFAILLANCES ET DES EFFETS CRITIQUES (AMDEC)

08/02/2024



Analyse des Modes de Défaillances et des Effets Critiques (AMDEC)

■ AMDEC:

> Définition :

- Analyse des Modes de Défaillances et des Effets Critiques
- En anglais : FMECA : Failure Mode and Effects and Critically Analysis

> Cette analyse quantitative permet :

- d'approfondir la notion de criticité des défaillances
- Mieux comprendre les processus de propagation des défaillances grâce aux arbres de défaillances

> Notion de criticité :

- $C = G * F * D$
- F : fréquence
- G : gravité
- D : détection

▪ **AMDEC:**
 ➤ Exemple :

Equipement	Fonction	Modes de défaillances	Effets	Causes	F	D	G	C
Injecteurs	Injecter le carburant	Bouchage	Pas d'injection	Cockéfaction	Rare	Moyenne	Réparation à faire	

- F : fréquence → probabilité du mode de défaillance
- G : gravité → importance des effets
- D : détection → probabilité de détecter avant que le mode apparaisse

■ AMDEC:

➤ Modes de défaillance génériques

1	Défaillance structurelle	6	Fuites	11	Entrées/ sorties erronées
2	Blocage physique	7	Problème de limites	12	Problème d'arrêts / démarrage / switch
3	Vibrations	8	Fonctionnement intempestif	13	Délais
4	Mouvement incontrôlé	9	Fonctionnement intermittent / irrégulier	14	Court-circuits
5	Problème ouverture / fermeture	10	Indication erronée	15	...

- **AMDEC:**

- Exemple de hiérarchisation de la probabilité de la défaillance

F	Proba	Fréquence
1	10^{-9}	1 fois tous les 10 ans
2	10^{-7}	1 fois par an
3	10^{-5}	1 fois par mois
4	10^{-4}	1 fois par semaine

- Dans notre exemple : P = 2

- **AMDEC:**

- Exemple de hiérarchisation de la gravité

G	Pertes matérielles	Gravité humaine
4	Perte du système	Mort
3	Dommages majeurs	Blessé grave
2	Dommages mineurs	Blessé léger
1	Aucun dommage	Aucun blessé

- Dans notre exemple : $G = 1$

- **AMDEC:**

- Exemple de détection

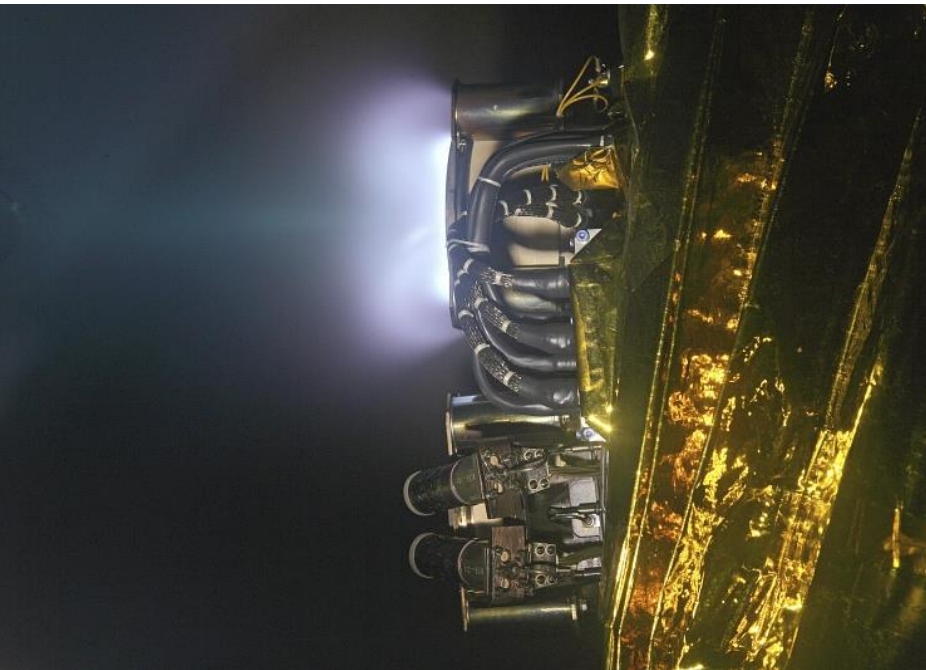
D	Critère de détection
1	10^{-9} de ne pas détecter
2	10^{-7} de ne pas détecter
3	10^{-5} de ne pas détecter
4	10^{-4} de ne pas détecter

- Dans notre exemple : $D = 2,5$

- **AMDEC:**

- Ensuite, on classe les problèmes par criticité :

C	Réaction
$C < 10$	Négligeable
$10 < C < 20$	Moyenne : étudier les cas pour voir s'il faut rehausser la criticité évaluée pour mener des actions, maintenance corrective
$20 < C < 40$	Élevée : trouver les actions à mener pour réduire la criticité, maintenance préventive
$40 < C$	Inacceptable : il faut agir. (on rappellera que l'AMDEC s'utilise aussi <u>pendant</u> la conception)



ARBRES DE DÉFAILLANCE & CHAÎNES DE MARKOV

08/02/2024



Arbres de défaillance

■ Arbres de défaillance

- Méthode déductive pour visualiser et quantifier les différents scénarios de défaillance
- On part d'un événement redouté
 - Panne complète
 - Mais aussi panne spécifique d'un sous-équipement
- On cherche les suites d'événements qui mènent à cet événement
- Puis on remonte récursivement
- Différence avec l'AMDEC :
 - Ici on cherche à quantifier les causes d'un événement particulier, et non la totalité des pannes
 - On peut imaginer effectuer un ou plusieurs arbres de défaillance pour avoir une AMDEC plus précise

Arbres de défaillance

■ Arbres de défaillance

> Fonctions de base :

- Ou : la défaillance d'un composant entraîne la défaillance de l'ensemble
- Et : la défaillance de l'ensemble entraîne la défaillance des composants

> Traitement qualitatif :

- Ecriture de l'équation de l'arbre
- Réduction par algèbre de Boole
- Coupes minimales / chemins critiques : plus petites combinaisons qui entraînent la perte du système / de la mission

> Traitement quantitatif

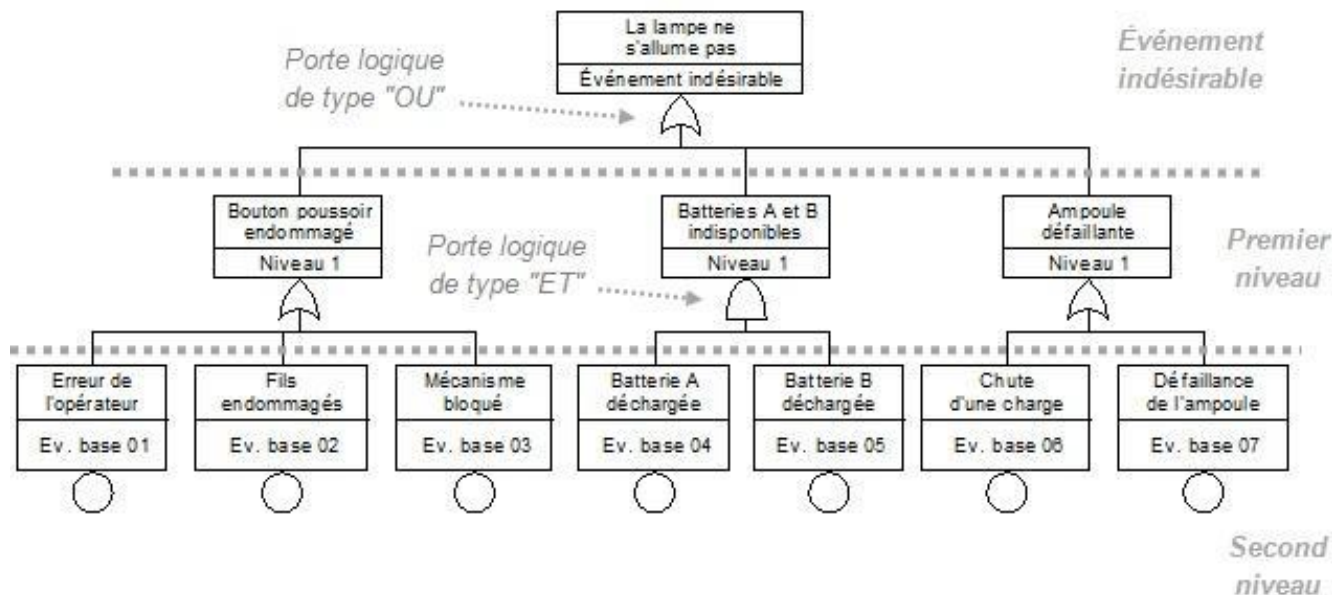
- En entrant les taux de défaillance estimé de chaque composant élémentaire, on peut calculer le taux de défaillance du système
- ET : $\lambda = \lambda_1 * \lambda_2$
- OU : $\lambda = 1 - (1 - \lambda_1) * (1 - \lambda_2)$
- (avec λ les taux de défaillance)
- On notera ici qu'on suppose les λ constants : on a gagné en complexité d'analyse mais on a perdu en complexité de modélisation du composant

Arbres de défaillance

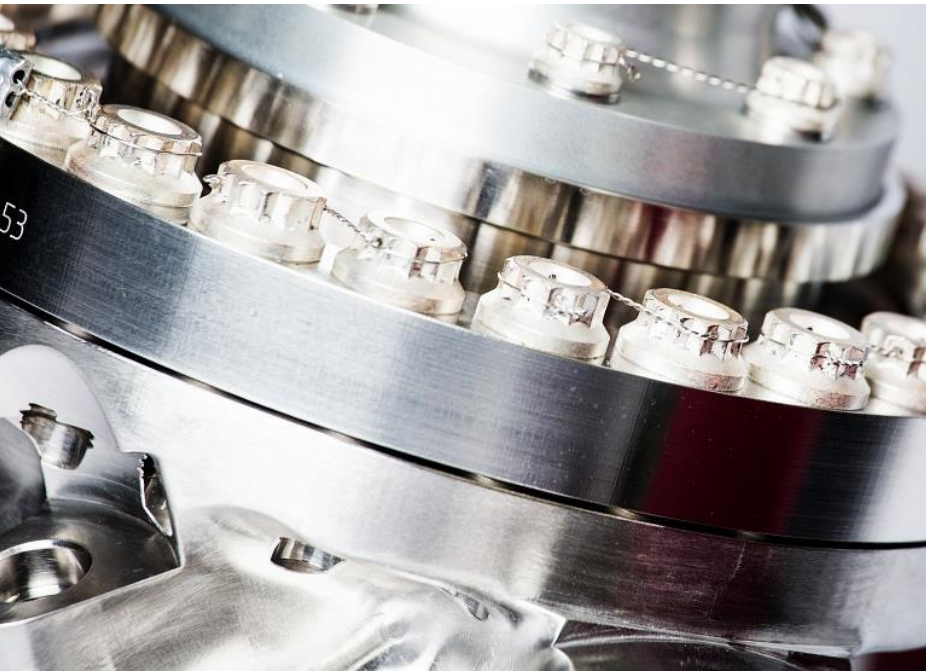
▪ Exemples

- > Événement : le radiateur électrique ne fonctionne plus
- > Causes de 1^{er} niveau :
 - Radiateur défaillant
 - Plus de courant
- > Causes de 2^{ème} niveau :
 - Radiateur défaillant:
 - ♦ Résistance défaillant
 - ♦ Contrôleur défaillant
 - Plus de courant :
 - ♦ Prise non branchée
 - ♦ Courant coupé
- > On associera à chacune de ces causes des probabilités

Arbres de défaillance



Source : wikipedia



RESEAUX BAYESIENS

08/02/2024



Réseaux bayésiens

▪ Rappels :

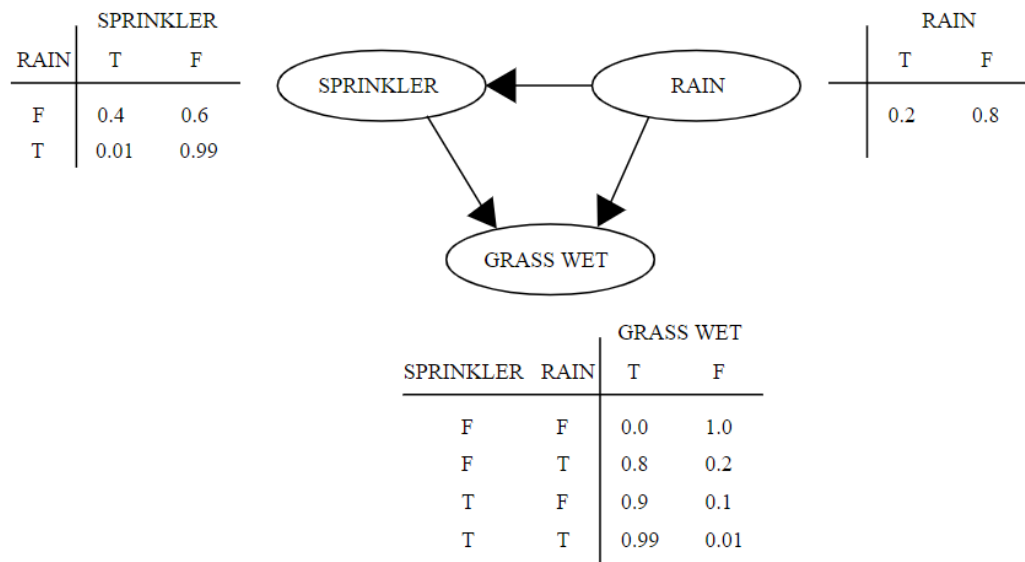
- > $p(a|b) = p(a, b)/p(b)$
- > a et b sont indépendants ssi $p(a, b) = p(a)p(b)$, $p(a|b) = p(a)$, $p(b|a) = p(b)$
- > Indépendance conditionnelle : a et b sont indépendants conditionnellement à c ssi $p(a|b, c) = p(a|c)$
- > Marginalisation : $p(a) = \sum_i p(a, b_i)$ avec les b_i qui forment l'ensemble complet d'événements mutuellement exclusifs
- > Si on connaît les probabilités conditionnelles $p(a|b_i)$ et les probabilités d'occurrence des b_i $p(b_i)$, on a $p(a) = \sum_i p(a|b_i)p(b_i)$
- > Théorème de Bayes : probabilité a posteriori $p(b_i|a) = p(a|b_i)p(b_i)/p(a)$

▪ Le réseau bayésien :

- > Prendre en compte les indépendances conditionnelles entre les variables pour simplifier la loi jointe
- > En français : prendre en compte les relations entre variables pour simplifier la modélisation du système
- > Deux composants :
 - Le graphe orienté
 - Les probabilités conditionnelles
- > Pour la maintenance prédictive → construire des modèles du système et/ou des pannes

Réseaux bayésiens

Exemple graphique

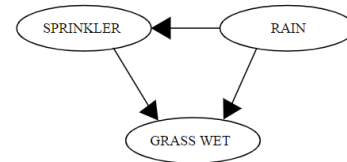


Réseaux bayésiens

Trois types de connexions :

- > Série : $A \rightarrow B \rightarrow C$
- > Divergente : $A \leftarrow B \rightarrow C$
- > Convergente : $A \rightarrow B \leftarrow C$

RAIN	SPRINKLER	
	T	F
F	0.4	0.6
T	0.01	0.99



	RAIN	
	T	F
	0.2	0.8

SPRINKLER	RAIN	GRASS WET	
		T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Réseaux bayésiens

Trois types de connexions :

> Série : $A \rightarrow B \rightarrow C$

- A & C sont dépendants
- A & C sont indépendants conditionnellement à B
 - ♦ Si je connais B, A est inutile pour déterminer C
 - ♦ $p(C|A,B) = p(C|B)$

> Divergente : $A \leftarrow B \rightarrow C$

- A & C sont dépendants
- A & C sont indépendants conditionnellement à B
 - ♦ Si je connais B, A est inutile pour déterminer C
 - ♦ $p(C|A,B) = p(C|B)$

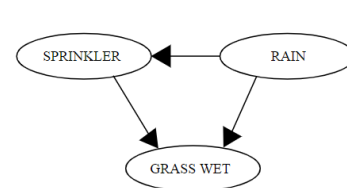
> Convergente : $A \rightarrow B \leftarrow C$

- A & C sont indépendants
- A & C sont dépendants conditionnellement à B
 - ♦ Si je connais B, A apporte de l'information sur C
 - ♦ $p(C|A,B)$ ne se simplifie pas ici

La loi jointe : $p(\text{Nodes}) = p(\text{Node}_1)p(\text{Node}_2|\text{Node}_1)p(\text{Node}_3|\text{Node}_1, \text{Node}_2) \dots$

- > Le réseau bayésien représente de manière compact cette loi
- > Sert à faire de l'échantillonnage (pas en rapport avec la maintenance)
- > Sert à calculer $p(\text{Node}_i|\text{Node}_j = x) \rightarrow$ pour la maintenance, recherche de causes

RAIN	SPRINKLER	
	T	F
F	0.4	0.6
T	0.01	0.99



	RAIN	
	T	F
	0.2	0.8

SPRINKLER	RAIN	GRASS WET	
		T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Réseaux bayésiens

- **calculer** $p(Node_i | Node_j = x)$
 - > Problème NP-complet
 - > Algorithmes exacts
 - Message Passing (Pearl 1988)
 - Junction Tree (Jensen 1990)
 - Shafer-Shenoy (1990)
 - > ➔ problème si graphes trop connectés
- **Comment apprendre un réseau bayésien ?**
 - > Les probabilités à structure fixée
 - > La structure

Réseaux bayésiens

■ Comment apprendre un réseau bayésien ?

> Les probabilités à structure fixée

- Par maximum de vraisemblance :

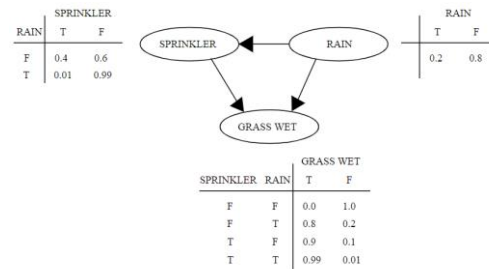
- ♦ $\hat{\theta} = \operatorname{argmax}_{\theta} (P(D|\theta))$
- ♦ En discret $p(event)$ est la fréquence d'apparition de l'événement
- ♦ $\hat{p}(X_i = x_k | Parents(X_i) = x_j) = \hat{\theta}_{i,j,k} = N_{i,j,k} / \sum_k N_{i,j,k}$ avec $N_{i,j,k}$ le nombre d'occurrences de $X_i = x_k$ et $Parents(X_i) = x_j$

- Par approche bayésienne avec maximum à posteriori

- ♦ $\hat{\theta} = \operatorname{argmax}_{\theta} (P(\theta|D)) = \operatorname{argmax}_{\theta} (P(D|\theta)p(\theta))$
- ♦ On a donc une loi « a priori » sur les périmètres $p(\theta)$ cas discret, on prend une Dirichlet : $p(\theta) = \prod_{i,j,k} \theta_{i,j,k}^{\alpha_{i,j,k}}$
- ♦ On obtient alors : $\hat{p}(X_i = x_k | Parents(X_i) = x_j) = \hat{\theta}_{i,j,k} = (N_{i,j,k} + \alpha_{i,j,k} - 1) / \sum_k (N_{i,j,k} + \alpha_{i,j,k} - 1)$
- ♦ Si au lieu de prendre le max on prend l'espérance, la solution est la même sans le précédemment

- Des extensions existent avec les données manquantes...

- ♦ Missing Completely At Random
- ♦ Missing at random
- ♦ Not Missing at random



Réseaux bayésiens

■ Comment apprendre un réseau bayésien ?

- Comment apprendre la structure ?
 - Croissance super-exponentielle
 - Ex : $NS(5) = 29281$ $NS(10) = 4,2 \times 10^{18}$

- Dans le cadre de la maintenance, on peut s'en servir pour de la recherche de cause racine

- Deux types de méthodes classiques :
 - Pearl et Verma : IC et IC*
 - Spirtes, Glymour et Scheines : SGS, PC, CI, FCI

- Le principe commun :
 - Construire un graphe non dirigé contenant les relations entre les variables
 - ♦ Par ajout d'arêtes (1^{er} cas)
 - ♦ Par suppression d'arêtes (2^{ème} cas)
 - Détecter les structures convergentes
 - Propager les orientations des arcs quand c'est possible

Réseaux bayésiens

▪ Comment apprendre un réseau bayésien ?

- Comment apprendre la structure ?
 - Recherche d'IC (indépendance conditionnelle)
 - Utilisation de tests statistiques (χ^2 par exemple)
 - Deux problèmes :
 - ♦ Explosion du nombre de tests -> heuristique, faire décroître les p-values avec la taille du réseau
 - ♦ Le test devient de moins en moins fiable avec la quantité de nœuds -> heuristique, commencer par les relations d'ordre 0, puis 1, puis 2, ...
 - Recherche de structures convergentes (V-structure)
 - Se fait grâce aux tests d'indépendance
- Propagation de liens : il ne faut pas introduire de V-structure, puisqu'on doit les avoir trouvés à l'étape d'avant

Réseaux bayésiens

- **Comment apprendre un réseau bayésien ?**

- Comment apprendre la structure ?

- Autre méthode:

- Faire directement de l'optimisation en donnant un score aux graphes (par exemple, de vraisemblance)
 - L'espace est très grand, donc souvent : soit on se restreint aux arbres, soit on se restreint dans la recherche par un algorithme « greedy »
 - On peut utiliser le résultat dans le domaine des arbres pour initialiser un algorithme plus large
 - → rappelez-vous le cours sur la sélection de variables

Réseaux bayésiens

▪ Exemple : imprimante

- > Nœuds d'action
- > Nœuds de question

> 3 pannes :

- Manque de toner
- Distribution toner défectueuse
- Problème de pilote

> Actions :

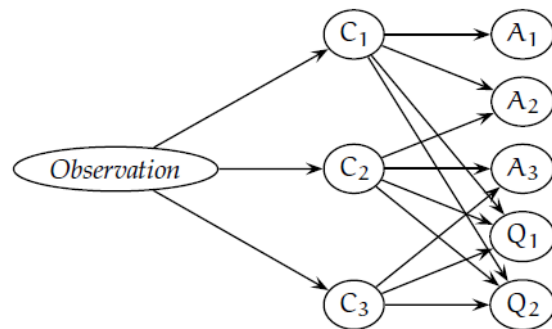
- Changer le toner
- Redémarrer
- Installer un nouveau pilote

> Questions :

- La page test s'imprime-t-elle correctement ?
- Y-a-t-il du papier dans l'imprimante ?

> On va chercher

- $P(A | C)$ (probabilité de faire l'action de manière efficace, sachant la cause) → choisir les actions
- $P(Q | C)$ (probabilité d'avoir une réponse donnée pour une cause donnée) → éliminer les causes



Les réseaux bayésiens

▪ Outil de manuel de vol auto apprenant

- Le concept : mettre à jour en temps réel les bases de données de trouble-shooting
- En pratique : mettre à jour les probabilités des causes possibles de panne, sachant certaines observations
- Lors de l'utilisation:
 - L'interface présente des questions ou des actions à faire
 - Pour chaque réponse apportée par l'opérateur, la liste des questions se met à jour, et la liste des causes aussi
 - Cette liste peut être classée par coût d'opération, ou bien par efficacité de discrimination
- Une fois l'opération de maintenance totalement terminée, les poids du réseau bayésien sont mis à jour

Les réseaux bayésiens

▪ Exemple de panne dans l'outil :

- > Panne de batterie
- > Questions / actions :
 - Vérifier les contacts
 - Mesurer le voltage
 - Remplacer les contacts
 - Mesurer l'intensité
 - Vérifier les fluides de la batterie
- > Causes:
 - Problème de contacts 90%
 - Problème de fluides 5%
 - Problème d'alternateur 2%
 - ...

Les réseaux bayésiens

▪ Exemple de panne dans l'outil :

> Panne de batterie

> Questions / actions :

- **Vérifier les contacts** ← action choisie, réponse contact OK
- Mesurer le voltage
- Remplacer les contacts
- Mesurer l'intensité
- Vérifier les fluides de la batterie

> Causes:

- Problème de fluides 50%
- Problème d'alternateur 20%
- ...

Les réseaux bayésiens

▪ Exemple de panne dans l'outil :

- > Panne de batterie
- > Questions / actions :
 - Mesurer le voltage
 - Mesurer l'intensité
 - **Vérifier les fluides de la batterie** ← action choisie, résultat non conforme

- > Causes:
 - **Problème de fluides 80%** ← problème trouvé
 - Problème d'alternateur 5%
 - ...



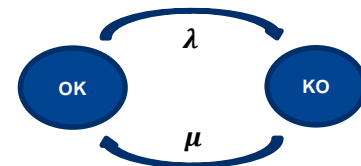
QUELQUES REMARQUES POUR ALLER PLUS LOIN

08/02/2024



Quelques remarques pour aller plus loin

- **Un certain nombre de sujets n'ont pas été abordés dans ce cours, mais son liés à l'analyse de données pour la maintenance prédictive**
- **Les chaînes de Markov**
 - > On peut ajouter des contraintes temporelles sur les taux de réparation et modéliser les vies suivantes
 - > En une petite équation : $P_{ok}(t + dt) = P_{ok}(t) \cdot (1 - \lambda dt) + (1 - P_{ok}(t)) \mu dt$, λ étant le taux de survie, μ le taux de réparation
- **Natural Language Processing :**
 - > Les données de maintenance sont souvent sous forme de langage naturel
 - > Mais avec du vocabulaire technique spécifique, et parfois non uniformisé
 - > NLP est un sujet de recherche très actif
- **Table learning :**
 - > Les données sont structurées en base de données
 - > Il existe des travaux sur l'analyse de table en tant qu'objet en eux-mêmes
 - > De même, l'utilisation d'algorithme sur les tables est un sujet de recherche très actif





CONCLUSIONS

08/02/2024



Conclusions

▪ 3 outils pour analyser des données de maintenance

- Analyse des Modes de Défaillances et des Effets Critiques
 - Outil théoriquement quantitatif mais amenant à des analyses qualitatives
- Arbres de défaillance
 - Outil théorique pour analyser un système complexe
- Réseaux Bayésiens
 - Permet de comprendre les enchaînements d'événements et remonter aux causes racines

▪ La semaine prochaine

- Examen ?
 - Première partie sur table (cours, sans document) ~ 45 minutes
 - Seconde partie TP avec accès à tous les documents
- Date de réception de l'email fait foi.