



HAL
open science

Décision de groupe, Aide à la facilitation : Ajustement de procédures de vote selon le contexte de décision

Adama Coulibaly

► To cite this version:

Adama Coulibaly. Décision de groupe, Aide à la facilitation : Ajustement de procédures de vote selon le contexte de décision. Intelligence artificielle [cs.AI]. Université Toulouse 1 Capitole (UT1 Capitole), 2019. Français. NNT: . tel-03726753

HAL Id: tel-03726753

<https://hal.science/tel-03726753>

Submitted on 18 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du **DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

Délivré par l'Université Toulouse 1 Capitole
Cotutelle internationale : Université des sciences, des techniques et des technologies de Bamako (USTTB)

Présentée et soutenue par
Adama COULIBALY

Le 4 juin 2019

**Décision de groupe, Aide à la facilitation : Ajustement de
procédures de vote selon le contexte de décision**

**Ecole doctorale : EDMITT - Ecole Doctorale Mathématiques, Informatique et
Télécommunications de Toulouse**

Spécialité : Informatique et Télécommunications

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Pascale ZARATE et Fana TANGARA

Jury

Mme Juliette Rouchier, Rapporteur
M. Patrice Perny, Rapporteur
M. Laurent Perrussel, Examineur
M. Guy Camillieri, Examineur
Mme Jacqueline Konaté, Examinatrice

M. Fana Tangara, Co-directeur de thèse
Mme Pascale ZARATE, Directrice de thèse

Remerciement

Il est impossible de mentionner le nom de l'ensemble des personnes qui m'ont aidé directement ou indirectement dans la réalisation de ce travail, néanmoins je tiens à remercier quelques-unes en particulier :

ZARATE Pascale, Professeur à l'Université Toulouse I, et membre de l'IRIT (Institut de Recherche en Informatique de Toulouse). Je la remercie du fond de mon cœur avec la plus grande sincérité de m'avoir donné l'opportunité de réaliser cette thèse. Malgré un planning chargé, elle était toujours disponible pour m'écouter et me guider pour le bon déroulement de ce travail. Ses conseils et soutiens moraux m'ont permis de passer des séjours agréables de recherches. Elle m'a aidé à surmonter les difficultés rencontrées et a été compréhensive quand j'accusais du retard dans les travaux. Je la remercie pour sa personnalité toujours souriante, accueillante et compréhensive envers nous, les thésards. Merci chère Professeur.

TANGARA Fana, Professeur à l'Université des Sciences, des Techniques et des Technologie de Bamako (USTTB) et Doyen de la Faculté des Sciences et Techniques. Je le remercie pour avoir accepté de co-diriger cette thèse. Son bureau m'a toujours été ouvert malgré un agenda ultra chargé. Je le remercie pour ses conseils au combien précieux dans la réalisation de ce travail. Merci cher Professeur, de m'avoir fait profiter de votre immense expérience et les efforts que vous faites pour l'enseignement supérieur du Mali.

CAMILLERI Guy, Maître de Conférence à l'Université Toulouse III (Université Paul Sabatier), qui a été mon co-encadrant pour cette thèse. Qu'il trouve ici mes remerciements pour les conseils et assistances qui ont conduit ce travail vers la réussite. Je le remercie d'avoir partagé ses expériences, bonne humeur et motivation avec moi. Vos discussions sur la thèse en général et sur la vie en particulier n'ont été bénéfiques. Merci cher Professeur.

KONATE Jacqueline, Maître Assistant à l'Université des Sciences, des Techniques et des Technologie de Bamako (USTTB). Je la remercie pour avoir accepté de faire partie de mon jury. Elle est à l'origine de cette thèse car c'est grâce à elle, que j'ai fait la connaissance de Pascale, je ne saurais l'oublier. Je la remercie également pour sa disponibilité, conseils importants. Je voudrais également la féliciter pour son dévouement dans son travail et l'encourager à continuer ainsi. Je la remercie pour son engagement pour un Mali meilleur et les efforts qu'elle déploie pour l'enseignement supérieure. Elle est une source d'inspiration pour moi et tous les jeunes enseignants.

ROUCHIER, Juliette, Directeur de recherche CNRS, LAMSADE, je la remercie pour avoir accepté de faire partie de mon jury.

PERNY Patrice, Sorbonne Université-LIP6, je le remercie pour avoir accepté de faire partie de mon jury.

PERRUSSEL, Laurent, Professeur à l'Université Toulouse 1 Capitole, et membre de l'IRIT (Institut de Recherche en Informatique de Toulouse), je le remercie pour avoir accepté de faire partie de mon jury

Je voudrais remercier également l'ensemble des membres de l'IRIT et notamment les membres de l'équipe ADRIA (Argumentation, Décision, Raisonnement, Incertitude et Apprentissage) pour m'avoir bien accueilli et m'offrir un cadre idéal me permettant d'effectuer mes travaux de recherche en toute quiétude.

Je voudrais aussi remercier l'ensemble de mes collègues de la Faculté des Sciences et Techniques de Bamako, qui ont été d'une aide précieuse dans ces travaux de recherche.

Un remerciement particulier à l'ambassade de France au Mali, son Service de coopération et d'action culturelle (SCAC) et Campus France qui aident le Mali à travers le programme d'appui à l'enseignement supérieur par le financement de séjours de recherche (bourses de doctorat).

Je ne saurais terminer sans remercier ma famille. À commencer par ma femme, Madina Dramé, qui m'a toujours soutenu et pris soin des enfants seule pendant mes longs séjours hors du pays. Je remercie mes enfants (Idrissa, Mariam, Fanta et Aminata) qui ont été privé de présence durant cette formation. Je remercie mes parents (Mohamed et Marian Arama) et mes frères et sœurs qui m'ont toujours incité à devenir ce que je suis aujourd'hui, merci pour vos soutiens et amours.

Résumé

La facilitation est un élément central dans une prise de décision de groupe surtout en faisant l'usage des outils de nouvelle technologie. Le facilitateur, pour rendre sa tâche facile, a besoin des solutions de vote pour départager les décideurs afin d'arriver à des conclusions dans une prise de décision.

Une procédure de vote consiste à déterminer à partir d'une méthode le vainqueur ou le gagnant d'un vote. Il y a plusieurs procédures de vote dont certaines sont difficiles à expliquer et qui peuvent élire différents candidats/options/alternatives proposées. Le meilleur choix est celui dont son élection est acceptée facilement par le groupe.

Le vote dans la théorie du choix social est une discipline largement étudiée dont les principes sont souvent complexes et difficiles à expliquer lors d'une réunion de prise de décision. Les systèmes de recommandation sont de plus en plus populaires dans tous les domaines de science. Ils peuvent aider les utilisateurs qui n'ont pas suffisamment d'expérience ou de compétence nécessaires pour évaluer un nombre élevé de procédures de vote existantes. Un système de recommandation peut alléger le travail du facilitateur dans la recherche d'une procédure vote adéquate en fonction du contexte de prise de décisions. Le sujet de ce travail de recherche s'inscrit dans le champ de l'aide à la décision de groupe. La problématique consiste à contribuer au développement d'un système d'aide à la décision de groupe (Group Decision Support System : GDSS). La solution devra s'intégrer dans la plateforme logicielle actuellement développée à l'IRIT GRUS : GRoUp Support.

MOTS-CLÉS : procédures de vote, Systèmes de recommandation, Systèmes d'aide à la Décision, GRECO, GRUS.

Abstract

Facilitation is a central element in decision-making, especially when using new technology tools. The facilitator, to make his task easy, needs voting solutions to decide between decision-makers in order to reach conclusions in a decision-making process.

A voting procedure consists of determining from a method the winner of a vote. There are several voting procedures, some of which are difficult to explain and which may elect different candidate/options/alternatives proposed. The best choice is the one whose election is easily accepted by the group.

Voting in social choice theory is a widely studied discipline whose principles are often complex and difficult to explain at a decision-making meeting. Recommendation systems are becoming more and more popular in all fields of science. They can help users who do not have sufficient experience or competence to evaluate large numbers of existing voting procedures. A recommendation system can lighten the facilitator's workload in finding an appropriate voting procedure based on the decision-making context. The objective of this research work is to design such recommendation system. This work is in the field of group decision support. The issue is to contribute to the development of a Group Decision Support System (GDSS). The solution will have to be integrated into the software platform currently being developed at IRIT GRUS: GRoUp Support.

KEYWORDS: voting procedures, recommendation systems, decision support systems, GRECO, GRUS

Table des matières

Remerciement	1
Résumé	3
Abstract	4
Table des tableaux	9
Table des figures	10
Table des formules	11
Introduction	12
PARTIE I : ETAT D'ART	14
CHAPITRE 1 :	14
1. Différentes procédures de vote	14
1.1 Les procédures sans classement	15
1.1.1 Méthodes de pluralité ou majoritaires	15
1.1.2 Méthode de vote pondérée (Vote par valeurs)	16
1.1.3 Méthode par approbation ou assentiment	16
1.2 Les procédures de votes par classement	17
1.2.1 Les Condorcet Non-Cohérentes	17
1.2.1.1 Méthode de BORDA	18
1.2.1.2 Méthode par élimination	19
1.2.1.3 Méthode de COOMBS	20
1.2.1.4 Méthode de HARE(STV)	22
1.2.1.5 Méthode de Bucklin	24
1.2.1.6 Méthode de Jugement majoritaire	25
1.2.2 Les procédures CONDORCET Cohérentes	28
1.2.2.1 Méthode de CONDORCET	28
1.2.2.2 Méthode de BLACK	30
1.2.2.3 La méthode de NANSON	31
1.2.2.4 Méthode de KEMENY-YOUNG	32
1.2.2.5 Méthode de COPELAND	33
1.2.2.6 Méthode de MINIMAX	35
1.2.2.7 Méthode de DODGSON	35
1.3 Conclusion	36
CHAPITRE 2 :	39

2. Systèmes de recommandation	39
2.1 Introduction aux systèmes de recommandation	40
2.2 Définitions des systèmes de recommandation	40
2.3 Différentes techniques de collecte d'informations	41
2.3.1 Collecte implicite	41
2.3.2 Collecte explicite	41
2.4 Classification des systèmes de recommandation	42
2.4.1 Filtrage basé sur le contenu ou recommandation objet	42
2.4.2 Filtrage Collaboratif ou Recommandation sociale	45
2.4.2.1 Méthodes basées sur la mémoire	45
2.4.2.2 Méthodes basées sur les modèles	46
2.4.2.2.1 Modèle de Cluster	46
2.4.2.2.2 Modèle à base de réseau bayésien	47
2.4.2.3 Avantages et inconvénients du filtrage collaboratif	48
2.4.3 Recommandation hybride	49
2.4.4 Exemple de systèmes de recommandation	52
2.4.5 Autres classifications de systèmes de recommandation	54
2.5 Conclusion	55
CHAPITRE 3 :	57
3. Caractérisations des procédures de vote : paramètres et critères	57
3.1 Introduction	58
3.2 Définitions	58
3.3 Paramètres pertinents d'un vote	59
3.3.1 Les formes des bulletins	59
3.3.2 Nature du vote	59
3.3.3 L'ordre de préférence de chaque électeur parmi les candidats en lice	59
3.3.4 La quantité d'informations que les électeurs ont sur les ordres de préférence des autres électeurs	60
3.3.5 L'ordre dans lequel les électeurs votent s'ils ne sont pas simultanés	60
3.3.6 L'ordre dans lequel les candidats sont votés si les candidats ne sont pas votés simultanément	60
3.3.7 Nombre de candidats	60
3.3.8 Nombre de candidats à élire	61
3.3.9 Gestion des égalités entre candidats	62
3.4 Les théorèmes importants de la théorie du vote	62
3.5 Les caractéristiques / attributs des procédures de vote	66

3.5.1	Les caractéristiques simples	66
3.5.1.1	Caractéristique du vainqueur de Condorcet (Condorcet, 1785 ; Black, 1958).....	66
3.5.1.2	Caractéristique du Perdant de Condorcet (Borda, 1784 ; Black, 1958).....	66
3.5.1.3	Caractéristique de la majorité absolue.....	66
3.5.1.4	Caractéristique du Perdant de la majorité.....	66
3.5.1.5	Caractéristique de Pareto (Fishburn, 1974)	66
3.5.2	Caractéristiques conditionnelles	66
3.5.2.1	Caractéristique de monotonie (Smith, 1973)	67
3.5.2.2	Caractéristique de cohérence (Young, 1974).....	67
3.5.2.3	Caractéristique de troncature (Brams, 1982 ; Fishburn et Brams, 1983).....	67
3.5.2.4	Caractéristique d'abstention (Fishburn et Brams, 1983 ; Ray, 1986 ; Moulin, 1988b ; Holzman, 1988/9 ; Perez, 1995).....	67
3.5.2.5	Caractéristique des jumeaux (Moulin, 1988b).....	67
3.5.2.6	Caractéristique d'indépendance (Fishburn (1974a, b, 1977))	67
3.5.2.7	Caractéristique de symétrie par inversion (Farquharson, 1969 ; Plott, 1973).....	67
3.5.2.8	Caractéristique d'invulnérabilité aux votes tactiques (Gibbard, 1973 ; Satterthwaite, 1975) 67	
3.5.3	Autres Caractéristiques : administratifs et techniques	68
3.5.3.1	La simplicité.....	68
3.5.3.2	Facilité de compréhension de la façon dont le gagnant est sélectionné	68
3.5.3.3	Facilité d'exécution des élections.....	68
3.5.3.4	Le risque de fraude.....	69
3.5.3.5	Discriminabilité.....	69
3.6	Caractéristiques des procédures de vote à l'échelle d'un groupe de travail à taille réduite	69
3.7	Conclusion	71
Partie II : Contribution		73
CHAPITRE 4		73
4.1.	Introduction	74
4.2.	Architecture détaillée	74
4.3.	Technologies utilisées	77
4.3.1.	Django.....	77
4.3.2.	Pandas, numpy et scipy	78
4.3.3.	MySQL	78
4.4.	Algorithme des méthodes de votes codées	79
4.4.2.	Méthode de Borda	80
4.4.3.	Méthode Condorcet.....	81

4.4.4.	Méthode Copeland.....	82
4.4.5.	Méthode de Black.....	83
4.5.	Caractérisation des procédures de vote.....	84
4.6.	Moteur de recommandation de GRECO.....	86
4.7.	GRECO en image : Exemple d'utilisation	89
4.8.	Conclusion.....	94
CHAPITRE 5.....		95
5.1.	Introduction	96
5.2.	Type d'expérimentation.....	96
5.2.1.	Expérimentation hors ligne	96
5.2.2.	Expérimentation basée sur un échantillon d'utilisateurs.....	96
5.2.3.	Expérimentation par évaluation en ligne	97
5.2.4.	Choix du type expérimentation	97
5.3.	Les mesures ou outils disponibles	97
5.3.1.	Mesures de précision prédictive	98
5.3.2.	Mesure de prévision d'utilisation	99
5.3.3.	Mesure d'ordre (ranking mesure)	100
5.3.3.1.	Normalized Distance-based Performance Measure (NDPM).....	101
5.3.3.2.	Mesures de corrélation.....	102
5.3.3.3.	Méthode de K-cross fold	104
5.4.	Expérimentations de GRECO	105
5.4.1.	Test des algorithmes des procédures de vote implémentés.....	105
5.4.2.	Validation par benchmark des exemples classiques de la littérature	109
5.4.3.	Mesures des précisions de la prédiction des votes des procédures de vote implémentées.....	113
5.4.4.	Mesure de prévision d'utilisation (Facilitateur)	115
5.5.	Conclusion.....	116
Conclusion générale		117
Publications de la thèse		119
Annexe.....		120
Bibliographie.....		123

Table des tableaux

Tableau 1-1: Exemple de pluralité	15
Tableau 1-2: Exemple de méthode de vote pondérée.....	16
Tableau 1-3: Exemple de la méthode par approbation.....	17
Tableau 1-4: Exemple de la méthode de Borda	18
Tableau 1-5: Exemple de la méthode par élimination.....	19
Tableau 1-6: Exemple de la méthode par élimination-bis.....	20
Tableau 1-7: Exemple de la méthode de COOMBS	20
Tableau 1-8: Exemple de la méthode de Coombs – itération 2.....	21
Tableau 1-9: Exemple de la méthode de Coombs – itération 3	21
Tableau 1-10: Exemple de la méthode de Coombs -Itération 4	22
Tableau 1-11: Exemple de la Méthode de HARE.....	23
Tableau 1-12: Exemple de la méthode de HARE-itér 1	23
Tableau 1-13: Exemple de la méthode de HARE- itér 2.....	24
Tableau 1-14: Exemple de la méthode de Bucklin.....	25
Tableau 1-15: Bulletin du votant n°1	26
Tableau 1-16: Bulletin du votant n°2	26
Tableau 1-17: Bulletin du votant n°3	26
Tableau 1-18: Bulletin du votant n°4.....	26
Tableau 1-19: Bulletin du votant n°5	26
Tableau 1-20: Profil de mérite des sources d'énergie candidates	26
Tableau 1-21: Départage des mentions	27
Tableau 1-22: Exemple de la méthode de CONDORCET.....	29
Tableau 1-23: Exemple de CONDORCET- les duels	29
Tableau 1-24: Exemple de Paradoxe de Condorcet	30
Tableau 1-25: Exemple de la méthode de Black.....	30
Tableau 1-26: Exemple de Black – résultat des duels.....	31
Tableau 1-27: Exemple de la Méthode Black – Application de Borda.....	31
Tableau 1-28: Exemple de méthode de NANSON.....	32
Tableau 1-29: Exemple de la méthode de KEMENY-YOUNG	33
Tableau 1-30: Exemple de la méthode de KEMENY-YOUNG-Duels.....	33
Tableau 1-31: Exemple de la méthode de Black.....	34
Tableau 1-32: Exemple de la méthode de COPELAND – résultat des duels.....	34
Tableau 1-33: Exemple de la méthode de COPELAND-Scores	35
Tableau 1-34: Exemple de la méthode de MINIMAX.....	35
Tableau 1-35: Exemple de la méthode de Dodgson.....	36
Tableau 1-36: Exemple de Dodgson-Duels.....	36
Tableau 3-1: Exemple de l'ordre des candidats	60
Tableau 3-2: Exemple du Nombre de candidat à élire	61
Tableau 3-3: Listes des Préférences des votants	63
Tableau 3-4: Listes de préférences avec une légère modification.....	65
Tableau 3-5: Critères vs Procédures de vote	70
Tableau 4-1: Matrice de caractérisation des procédures de vote de GRECO (<i>Mp</i>)	84

Tableau 4-2: Liste des notes possibles	85
Tableau 4-3: Matrice des notes des facilitateurs (<i>Mn</i>).....	86
Tableau 4-4: Préférences des votants-GRECO	89
Tableau 5-1: Classification des résultats possibles d'une recommandation d'un élément pour l'utilisateur	99
Tableau 5-2: Exemple de valeur de coefficient de Kendall Tau-b.....	104
Tableau 5-3: Préférence de votants de la situation 1	106
Tableau 5-4: Résultats manuels situation 1	107
Tableau 5-5: Résultats de la situation 1 avec GRECO.....	108
Tableau 5-6: Situation 2- Préférences des votants	109
Tableau 5-7: Résultats des duels de Condorcet.....	109
Tableau 5-8: Préférences des votants (situation 3).....	112
Tableau 5-9: Résultats de GRECO-situation 3.....	113
Tableau 5-10: GRECO -Mesures de prédiction d'utilisation.....	116

Table des figures

Figure 1-1: Jugement majoritaire- Histogramme	27
Figure 2-1: Filtrage basé sur le contenu (JANNACH, et al., 2010)	43
Figure 2-2: Filtrage collaboratif (JANNACH, et al., 2010)	45
Figure 2-3: Filtrage hybride (JANNACH, et al., 2010).....	50
Figure 2-4: Conception d'hybridation monolithique (JANNACH, et al., 2010).....	51
Figure 2-5: Conception d'hybridation parallèle (JANNACH, et al., 2010)	51
Figure 2-6: Conception d'hybridation tubulaire(JANNACH, et al., 2010).	52
Figure 2-7: Exemple de MovieLens	53
Figure 2-8: Système de recommandation du site web Filmaffinity.....	54
Figure 4-1: Digramme de Classes de GRECO	75
Figure 4-2: Architecture interne de Django.....	78
Figure 4-3: Logique interne de GRECO.....	86
Figure 4-4: Ecran de données de bases du vote.....	90
Figure 4-5: Ecran de paramètres de recommandation.....	91
Figure 4-6: Ecran de précision de la procédure recommandée.....	92
Figure 4-7: Ecran de participation au vote	92
Figure 4-8: Ecran du résultat du vote	93
Figure 4-9: Ecran de notation de la procédure de vote.....	93
Figure 5-1: Création du vote de la situation 2 (paramètres).....	110
Figure 5-2: Création du vote de la situation 2 (procédure recommandée)	111
Figure 5-3: Résultat du vote de la situation 2.....	111
Figure 5-4: GRECO- RMSE	114
Figure 5-5: GRECO-MAE	115

Figure 5-6: Précision -Rappel-F-mesure de GRECO 116

Table des formules

$SplCand = \frac{NbVotant(Cand)}{n}$ (F1.1).....	15
$SplCand > i = \frac{1}{n} \sum_{j=1}^i n_j$ (F1.2).....	15
$Sc(Cand) = i = \frac{1}{n} \sum_{j=1}^i n_j$ (F1.3).....	16
$SBordaCand = i = \frac{1}{n} \sum_{j=1}^i [(nbV_j = n - i(Cand)) * i]$ (F1.4).....	18
$Coefdist = 1 - \frac{QNbVotantElu}{n}$ (F1.5).....	22
$xiPxj = 1$, si $Rangxi < Rang(xj)$, 0, si $Rangxj < Rang(xi)$ (F1.6).....	28
$Spref(xi, xj) = i = \frac{1}{n} \sum_{k=1}^i n_k x_{kj}$ (F1.7).....	28
$Duel(xi, xj) = xi$, si $Sprefxi, xj > Spref(xj, xi)$, xj , si $Sprefxi, xj < Spref(xj, xi)$ (F1.8).....	28
$C = \{C_1, \dots, C_k\}$ tels que : $\forall l \in 1, \dots, K, C_l / \emptyset \forall m \in 1, \dots, K_2, C_l \cap C_m = \emptyset$ (F2.1).....	46
$PrC = c, v_1, \dots, v_n = Pr(C = c) = \frac{1}{n} \sum_{i=1}^n Pr(v_i C = c)$ (F2.2).....	47
$RMSE = \frac{1}{ T } \sum_{(u, i) \in T} (r_{ui} - \hat{r}_{ui})^2$ (F5.1).....	98
$MAE = \frac{1}{ T } \sum_{(u, i) \in T} r_{ui} - \hat{r}_{ui} $ (F5.2).....	98
précision = $\frac{\#Vp}{\#Vp + \#Fp}$ (F5.3).....	100
rappel = $\frac{\#Vp}{\#Vp + \#Fn}$ (F5.4).....	100
Taux $Vp = \frac{\#Vp}{\#Vp + \#Fn}$ (F5.5).....	100
F - mesure = $2 * \text{precision} * \text{rappel} / (\text{precision} + \text{rappel})$ (F5.6).....	100
$C^+ = \frac{1}{n} \sum_{i,j} \text{sign}(r_{ui} - r_{uj})$ (F5.7) $C^- = \frac{1}{n} \sum_{i,j} \text{sign}(r_{uj} - r_{ui})$ (F5.8).....	102
$C_u = \frac{1}{n} \sum_{i,j} \text{sign}(r_{ui} - r_{uj})$ (F5.9).....	102
$C_s = \frac{1}{n} \sum_{i,j} \text{sign}(r_{ui} - r_{uj})$ (F5.10).....	102
$C_u^0 = C_u - (C^+ + C^-)$ (F5.11).....	102
$NDPM = C - 0.5 C_u^0 C_u$ (F5.12).....	102
$\rho = 1 - \frac{6i}{n} = \frac{1}{n} \sum_{i=1}^n (2i - 3)$ (F5.13).....	103
$\tau = C^+ - C^- - C_u C_s$ (F5.14).....	103
$\tau_b = C^+ - C^- - \frac{12}{n(n-1)} (U_x)(n-1 - U_y)$ (F5.15).....	103
Avec $U_x = \sum_{i,j} (u_x - 1)$ et $U_y = \sum_{i,j} (u_y - 1)$ (F5.16).....	103

Introduction

Le facilitateur joue un rôle important dans le processus de l'aide à la décision de groupe. Peu de logiciels sont actuellement développés pour aider le facilitateur dans sa tâche d'assistance au processus de prise de décision. Cette assistance peut être définie sur le plan technique mais aussi sur le contenu ou bien dans la conduite du processus de prise de décision. Par exemple un processus de prise de décision d'un groupe peut être composé des étapes suivantes : - Brainstorming - Catégorisation des idées - Etablissement d'un consensus. Dans le cas d'un consensus, il sera facile de conclure. Dans le cas contraire un recours au vote s'impose, donnant ainsi à chaque membre du groupe, la possibilité d'exprimer ses préférences selon les circonstances de travail.

Il existe différentes procédures de vote applicables. Avec l'évolution de la science informatique, les systèmes de recommandation sont de plus en plus omniprésents dans tous les domaines. L'objectif général de ce travail de recherche s'inscrit dans cadre de la mise en place d'un système d'aide à la Décision de groupe pour le facilitateur, capable de lui recommander une procédure de vote à appliquer selon le contexte.

Notre problématique consiste à mettre en place un système d'assistance qui soit :

- **Evolutif, et interactif**, c'est à dire qu'on puisse ajouter au fur et à mesure d'autres procédures de vote. Le système doit être modulaire capable d'intégrer la palette d'outils du système GRoUp Support (GRUS).
- **Intelligent**, c'est-à-dire avoir un mécanisme de recommandation de procédures de vote adéquate au facilitateur.

Cette thèse se décompose en deux parties principales, qui sont :

Premièrement, l'état de l'art, comportant trois chapitres. Dans le chapitre 1, nous allons décrire les différentes procédures de vote. Cette description est utile et nécessaire à la compréhension pour la mise en place et la codification des procédures de vote. Il est question des procédures de vote par classement et sans classement.

Ensuite nous allons faire une introduction des différents types de systèmes de recommandation dans le chapitre 2. Un système de recommandation est un système d'aide à la décision capable de fournir à l'utilisateur des éléments susceptibles de l'intéresser afin de l'aider dans sa prise de décision Cette introduction va nous permettre de connaître le fonctionnement des différents algorithmes associés.

Nous ferons dans le chapitre 3, une caractérisation des procédures de votes rencontrées et décrites dans le chapitre 1 afin de dresser des critères nécessaires pour notre solution de recommandation.

Les contributions que nous avons réalisées sont décrites dans la partie 2. Nous proposons dans le chapitre 4, les détails de la mise place de Group RECOmmentation (GRECO). Dans ce chapitre présentons l'architecture détaillé que nous avons adapté et les technologies utilisées. Nous expliquons en pseudo code, les algorithmes de recommandation hybride mis en place. En dernier point de ce chapitre, nous proposons des captures d'écrans à travers un exemple d'utilisation.

Enfin nous présentons dans le chapitre 5, les validations expérimentales et benchmark de notre travail. Pour cela nous commençons à décrire les différents types d'expérimentation d'un système de recommandation. Ensuite nous distinguons les catégories de mesures applicables dans les expérimentations d'un système de recommandation qui sont : les mesures de précision prédictive, les mesures de prévision d'utilisation et les mesures d'ordre.

Ensuite nous décrivons les validations expérimentales que nous avons adaptées pour évaluer notre système expliquant nos choix.

Nous terminons ce manuscrit en présentant notre conclusion générale et des perspectives à ces travaux de recherche.

PARTIE I : ETAT D'ART

Cette partie est composée de 3 chapitres. Dans le chapitre 1, nous allons décrire les différentes procédures de vote. Cette description est utile et nécessaire à la compréhension pour la mise en place et la codification des procédures de vote.

Ensuite nous allons faire une introduction des différents types de systèmes de recommandation dans le chapitre 2.

Nous ferons dans le chapitre 3, une caractérisation des procédures de votes rencontrées et décrites dans le chapitre 1 afin de dresser des critères nécessaires pour notre solution de recommandation.

CHAPITRE 1 :

1. Différentes procédures de vote

Le **vote**, terme dérivé de l'anglais *vote*, provenant du latin *totum* signifiant « vœu », désigne une méthode permettant à un groupe une prise de décision commune. Les organisations formelles ou informelles de toute nature (économiques, politiques, associatives, etc.) ont recours à cette pratique, La pratique du vote vise à donner une légitimité à la décision en montrant qu'elle ne vient pas d'un individu isolé. Avant que le vote proprement dit n'ait lieu, il est fréquent qu'un temps de discussion ou de débat soit aménagé pour permettre à chacun des votants d'exposer ou de prendre connaissance des arguments, afin de motiver au mieux sa décision.

Définition 1.1 (procédure de vote) : une procédure de vote consiste à déterminer à partir d'une méthode le vainqueur ou le gagnant du vote. Les procédures de vote sont des outils qui permettent de prendre des décisions dans un contexte de choix social. Elles sont utilisées pour élire un vainqueur parmi des candidats au moment d'une élection.

Définition 1.2 (gagnant de Condorcet) : Le gagnant de Condorcet est le candidat qui peut battre à la majorité tout autre candidat en face-à-face. Un tel candidat peut ne pas exister.

Définition 1.3 (Condorcet cohérente ou consistante) : Une procédure de vote est dite Condorcet-consistante ou cohérente si elle permet d'élire le gagnant Condorcet.

Définition 1.4 (Non Condorcet Cohérent) : Une procédure de vote est dite Non Condorcet Cohérent, si elle peut élire un candidat qui n'est pas forcément le gagnant de Condorcet.

Il existe différentes procédures de vote qui peuvent se regrouper en deux groupes : les Procédures de vote sans classement (scrutin uninominal) et par classement (scrutin de liste). Les procédures par classement se font en soumettant une liste de préférence ordonnée de candidats. Les procédures sans classement n'ont pas besoin d'ordre spécifique dans la liste de préférence que le votant doit soumettre. (BALINSKI, et al., 2012)

1.1 Les procédures sans classement

Dans les procédures sans classement, les votants n'ont pas besoin de dresser une liste ordonnée de préférences pour exprimer leur choix. Parmi lesquelles nous citons les procédures suivantes.

1.1.1 Méthodes de pluralité ou majoritaires

Dans ces méthodes les règles d'agrégation sont basées uniquement sur le nombre de votants qui préfèrent un candidat à autre. Ainsi on aura une majorité **simple** ou une **pluralité** si un candidat obtient un nombre de vote plus grand que les autres, on parle aussi d'une majorité **relative**. La majorité est **absolue** si le nombre de votes d'un candidat est plus grand que la moitié du nombre total de vote. Par exemple, soit 3 candidats A, B et C et soit 7 votants. Chaque votant peut voter pour un seul candidat. Le tableau 1.1 contient le résultat du vote.

Votant →	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇
Candidat	C	C	D	B	A	A	A

Tableau 1-1: Exemple de pluralité

La formule suivante permet de calculer le score des différents candidats dans la méthode de pluralité.

$$S_{pl}(Cand) = Nb_{votant}(Cand). \quad (F1.1)$$

Avec Nb_{votant} : nombre de votants pour le candidat.

Pour avoir une majorité absolue, il faut que

$$S_{pl}(Cand) > \frac{\sum_{i=1}^n V_i}{2}, \quad (F1.2)$$

Où n est le nombre de votants.

Donc $S_{pl}(A) = 3$, $S_{pl}(B) = 1$ et $S_{pl}(C) = 2$.

Le candidat A, avec 3 voix gagne par majorité simple ;

Pas de gagnant par majorité absolue car pour cela il faut 4 voix sur 7 votants.

Il existe d'autres variantes de cette procédure à savoir : la majorité qualifiée ou renforcée et la double majorité, pour plus de détails voir (FISHBURN, 1974).

1.1.2 Méthode de vote pondérée (Vote par valeurs)

Le principe de cette méthode est de donner aux votants un nombre de poids définis que chacun devra répartir sur les différents candidats. Ensuite, collecter l'ensemble des votes afin de classer dans l'ordre décroissant les candidats ainsi évalués (DAUBECHIES, 2002) (BAUJARD, 2014). Cette technique est rapide mais très subjective et peut se mettre en place en trois étapes :

- Choix du barème de notation : Le groupe doit décider du nombre de poids qui servira au vote, chaque poids ayant une valeur différente. C'est ce principe qui va permettre à chacun d'évaluer les candidats.
- Déroulement du vote : Le groupe procède au vote individuel dans le secret le plus total afin d'éviter tout risque d'influence.
- Collecte des résultats, par exemple lors d'une élection, trois (3) candidats A, B et C se présentent. Chaque électeur peut voter pour plus d'un candidat en lui attribuant des points. Le tableau suivant dresse les résultats du vote. On observe que le candidat A, avec un score de 13 points, est le gagnant.

Votant→	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	Score
A	2	4	1	0	3	2	1	13
B	0	5	3	0	1	0	1	10
C	2	1	1	1	2	5	0	12

Tableau 1-2: Exemple de méthode de vote pondérée

Le score d'un candidat se calcule en utilisant la formule suivante :

$$S_c(\mathbf{Cand}) = \sum_{i=1}^n P_i(\mathbf{Cand}), \quad (F1.3)$$

Avec P_i : le point attribué au candidat par le votant i et n le nombre de votants.

1.1.3 Méthode par approbation ou assentiment

Le vote par approbation ou vote par assentiment est un système de vote simple étudié et défendu par des théoriciens depuis les années 1970. (BAUJARD, 2014)

Chaque votant constitue une liste de tous les candidats qu'il souhaite soutenir par son vote. Le candidat qui recueille le plus grand nombre de voix est élu.

Pour mettre en place ce système, le votant range l'ensemble des candidats en deux groupes :

- Les candidats qui reçoivent son approbation ;
- Les candidats qui ne reçoivent pas son approbation.

Ensuite il donne ses voix à chacun des candidats du sous ensemble à approuver. Le candidat qui obtient le plus grand nombre de votes est vainqueur.

Cette méthode est utilisée pour des élections à un seul gagnant mais peut se généraliser pour des élections à plusieurs gagnants. Il existe plusieurs variantes, dont en particulier le vote d'approbation proportionnel qui cherche à optimiser le degré de satisfaction générale.

Par exemple lors d'une élection, trois (3) candidats A, B et C se présentent. Chaque électeur peut voter pour plus d'un candidat s'il le désire et ne pas classer la liste complète.

Voici les résultats du vote.

Nombre de votants→	1	1	2	4	2
	C	B	B	A	C
		C	A		B
			C		A

Tableau 1-3: Exemple de la méthode par approbation

Le score d'un candidat dans la méthode par approbation se calcule avec la formule (F1.1), qui détermine le nombre de votant qui a approuvé ledit candidat.

En analysant les listes d'approbation des votants nous obtenons les scores suivants, $S_{va}(A) = 8$, $S_{va}(B) = 5$, $S_{va}(C) = 6$. Le candidat A qui comptabilise le plus grand score est désigné gagnant du vote.

1.2 Les procédures de votes par classement

Dans les procédures de vote par classement, plutôt que de simplement indiquer leur candidat préféré, les votants classent les candidats dans l'ordre de leur choix.

1.2.1 Les Condorcet Non-Cohérentes

Les procédures de vote classées Condorcet non-cohérente peuvent élire un gagnant qui n'est pas forcément le vainqueur de Condorcet. Dans cette catégorie de procédures nous pouvons citer les suivantes.

1.2.1.1 Méthode de BORDA

C'est une méthode qui a été formalisée en 1770 par Jean-Charles de Borda. Les électeurs classent les candidats selon leurs préférences. Celui qui recueille le plus de points gagne l'élection. Elle est très populaire et peut être utilisée pour élire un ou plusieurs candidats. (DE BORDA, 1781) (BOURSIN, 2004)

Pour mettre en place cette procédure, on attribue des points à chacune de position de la façon suivante : Avec n candidats, on donne :

- n points à la première position,
- $n-1$ points à la deuxième, ainsi de suite,
- 2 points à l'avant-dernière position et
- 1 point à la dernière.

On calcule le nombre de points de chacun des candidats en utilisant la formule [F1.4]. Le gagnant est celui qui obtient le plus grand nombre de points.

$$S_{Borda}(Cand) = \sum_{i=1}^n [(nbV_{j=n-i}(Cand)) * i] , (F1. 4)$$

D'où $nbV_j(Cand)$ désigne le nombre de votants qui ont placé Cand à la position j .

Par exemple, lors d'une élection sept (7) votants veulent choisir un gagnant parmi trois (3) candidats A, B et C. Le tableau présente les préférences exprimées par les votants :

Votant→		v_1	v_2	v_3	v_4	v_5	v_6	v_7	Points
Rang	1 ^{ère}	C	C	B	B	A	A	A	3
	2 ^{ème}	B	B	C	C	B	C	C	2
	3 ^{ème}	A	A	A	A	C	B	B	1

Tableau 1-4: Exemple de la méthode de Borda

Les trois candidats obtiennent les scores suivants :

- $S_{Borda}(A) = (3*3) + (4*1) = 13$ points,
- $S_{Borda}(B) = (2*3) + (3*2) + (2*1) = 14$ points,
- $S_{Borda}(C) = (2*3) + (4*2) + 1 = 15$ points,

Le candidat C avec 15 points, est le gagnant.

1.2.1.2 Méthode par élimination

C'est une combinaison itérative de la méthode de majorité. Pour l'emporter, l'alternative doit obtenir une majorité absolue. Si ce n'est pas le cas on procède de la façon suivante : À la première étape on ne s'intéresse qu'au 1^{er} choix de chaque votant, on élimine le candidat ayant obtenu le moins de votes et on attribue ses votes au candidat qui le suit. À la deuxième étape, on procède à un nouveau comptage pour désigner la première place. Si un candidat obtient alors la majorité des votes, il est vainqueur. Sinon, on recommence la procédure à partir de la première étape. (BOURSIN, 2004)

Par exemple, Bob, Berthe et Robert se sont présentés pour être représentant de classe de 36 élèves. On a demandé aux élèves de placer les candidats en ordre de préférence. Voici les résultats :

Nombre de votants→		9	5	4	12	4	2
Rang	1 ^{er}	Bob	Berthe	Robert	Robert	Berthe	Bob
	2 ^e	Berthe	Bob	Bob	Berthe	Robert	Robert
	3 ^e	Robert	Robert	Berthe	Bob	Bob	Berthe

Tableau 1-5: Exemple de la méthode par élimination

En calculant les voix des candidats nous obtenons :

- $S_{pl}(Bob) = 9 + 2 = 11$
- $S_{pl}(Berthe) = 5 + 4 = 9$
- $S_{pl}(Robert) = 4 + 12 = 16$

Le nombre total de votant est égal à 36, la majorité absolue est estimée à 18 voix par conséquent pas de gagnant. On passe à l'étape suivante en éliminant le candidat **Berthe** qui a obtenu le plus petit nombre de voix. On obtient le **tableau 1-6**.

Nombre de votants→		9	5	4	12	4	2
Rang	1 ^{er}	Bob	Bob	Robert	Robert	Robert	Bob
	2 ^e	Robert	Robert	Bob	Bob	Bob	Robert

Tableau 1-6: Exemple de la méthode par élimination-bis

En calculant de nouveau les scores des candidats on obtient :

- $S_{pl}(Bob) = 9 + 5 + 2 = 16$
- $S_{pl}(Robert) = 4 + 12 + 4 = 20$

Le candidat Robert, obtient une majorité absolue donc est le gagnant.

1.2.1.3 Méthode de COOMBS

C'est une méthode itérative avec élimination, inventée par Clyde Coombs vers 1954, pour sélectionner le meilleur candidat.

Les votants classent les candidats par ordre de préférence. On compte les voix des candidats premiers de liste. Si un candidat obtient la majorité absolue des voix, il est élu. Sinon, on supprime le candidat qui apparait le plus souvent en dernier de liste et on barre son nom dans toutes les listes de préférences, modifiant ainsi le rang des candidats placés après le candidat éliminé. Puis on recalcule les scores. On poursuit le processus jusqu'à ce qu'une majorité absolue se dessine. (BOURSIN, 2004)

Par exemple, quarante-cinq votants classent cinq candidats A, B, C, D et E. Les scores des 45 classements obtenus sont :

Nombre de votants→		5	5	8	3	7	2	7	8
Rang	1 ^{ère}	A	A	B	C	C	C	D	E
	2 ^{ème}	C	D	E	A	A	B	C	B
	3 ^{ème}	B	E	D	B	E	A	E	A
	4 ^{ème}	E	C	A	E	B	D	B	D
	5 ^{ème}	D	B	C	D	D	E	A	C

Tableau 1-7: Exemple de la méthode de COOMBS

Le calcul des scores des candidats se fait en utilisant la formule [F1.1]. Lors de la première itération on obtient :

$$S_{pl}(A) = 10; S_{pl}(B) = 8; S_{pl}(C) = 12; S_{pl}(D) = 7 \text{ et } S_{pl}(E) = 8$$

La majorité absolue implique un score de 23 car on a 45 votants. Le candidat C apparait plus de fois (16 votants) en bas des classements donc sera éliminé. On obtient le tableau suivant pour la seconde itération.

Nombre de votants→		5	5	8	3	7	2	7	8
Rang	1 ^{ère}	A	A	B	A	A	B	D	E
	2 ^{ème}	B	D	E	B	E	A	E	B
	3 ^{ème}	E	E	D	E	B	D	B	A
	4 ^{ème}	D	B	A	D	D	E	A	D

Tableau 1-8: Exemple de la méthode de Coombs – itération 2

Le calcul pour la deuxième itération, donne les scores suivants :

$$S_{pl}(A) = 20; S_{pl}(B) = 10; S_{pl}(D) = 7; \text{ et } S_{pl}(E) = 8$$

Toujours pas de majorité absolue, donc on passe à une troisième itération en éliminant le candidat **B** qui occupe plus de fois le bas des classements. On obtient le tableau I-9 suivant :

Nombre de votants→		5	5	8	3	7	2	7	8
Rang	1 ^{ère}	A	A	B	A	A	B	E	E
	2 ^{ème}	B	E	E	B	E	A	B	B
	3 ^{ème}	E	B	A	E	B	E	A	A

Tableau 1-9: Exemple de la méthode de Coombs – itération 3

Nous obtenons les scores suivants :

$$S_{pl}(A) = 20; S_{pl}(B) = 10; \text{ et } S_{pl}(E) = 15$$

Une quatrième itération, en éliminant le candidat **A**, donne la majorité absolue au candidat **E** qui sera le gagnant de la méthode de Coombs comme le montre le tableau suivant

Nombre de votants→		5	5	8	3	7	2	7	8
Rang	1 ^{ère}	B	E	B	B	E	B	E	E
	2 ^{ème}	E	B	E	E	B	E	B	B

Tableau 1-10: Exemple de la méthode de Coombs -Itération 4

Les candidats restants obtiennent les scores suivants :

$$S_{pl}(B) = 18 \text{ et } S_{pl}(E) = 27$$

1.2.1.4 Méthode de HARE(STV)

Appelée aussi vote unique transférable, a été inventée par Thomas Hare. C'est une méthode de quota et d'élimination qui visent à assurer la représentation proportionnelle pour satisfaire la volonté populaire. Si on a n candidats et v votants, le quota de Hare, pour élire m vainqueurs, se calcule comme suit : $Q = E \left[\frac{v}{m+1} \right] + 1$ où E est la fonction de la partie entière et Q représente le nombre de voix minimale pour être élu. (HYMAN, 2011)

Chaque votant classe les candidats par ordre de préférence pour élire m vainqueurs. Pour proclamer le résultat, on calcule le quota Q qu'on compare aux scores des premiers des listes. Les candidats dont les scores atteignent ou dépassent le quota seront élus.

- Si aucun candidat n'atteint le quota, celui qui obtient le moins de voix est éliminé et ses voix sont reportées au deuxième de chaque liste de préférence où il figurait en tête.
- Si un candidat dépasse le quota, l'excédent de voix est distribué équitablement aux seconds de chaque liste où il figurait en tête utilisant la formule **FI.5**.

$$Coef_{dist} = 1 - \frac{q}{Nb_{votantElu}} ; \quad (FI. 5)$$

$Nb_{VotantsElu}$ est le nombre de voix obtenu par le candidat qui atteint le quotas fixé.

On répète la même procédure jusqu'à obtenir les m vainqueurs.

Par exemple, élire deux (2) gagnants avec 45 votants et 5 candidats dont les préférences exprimées sont dans le tableau suivant :

Nombre de votants→		5	5	8	3	7	2	7	8
Rang	1 ^{ère}	A	A	B	C	C	C	D	E
	2 ^{ème}	C	D	E	A	A	B	C	B
	3 ^{ème}	B	E	D	B	E	A	E	A
	4 ^{ème}	E	C	A	E	B	D	B	D
	5 ^{ème}	D	B	C	D	D	E	A	C

Tableau 1-11: Exemple de la Méthode de HARE

Le quota $Q = E \left(\frac{45}{(2+1)} \right) + 1 = 16$ voix pour passer.

Les scores des têtes de liste sont les suivants :

$$S_{pl}(A) = 10 \text{ et } S_{pl}(B) = 8, S_{pl}(C) = 12 \text{ et } S_{pl}(D) = 7 \text{ et } S_{pl}(E) = 8$$

On remarque qu'aucun candidat atteint le quota fixé, 16 voix et le candidat D avec 7 voix, obtient le plus petit score et sera éliminé. En calculant les nouveaux scores des têtes de liste, on obtient :

$$S_{pl}(A) = 10 \text{ et } S_{pl}(B) = 8, S_{pl}(C) = 19 \text{ et } S_{pl}(E) = 8$$

Ainsi le candidat C avec 19 voix, est élu avec un excédent de 3 voix par rapport au quota. On calcule le coefficient de distribution par rapport au 19 voix obtenue $Coef_{dist} = \left(1 - \frac{16}{19} \right) = 0.58$. En distribuant équitablement cet excédent de point entre les listes qui ont placé le candidat élu en tête, on obtient le tableau suivant.

Nombre de votants→		5	5	8	0.47	1.1	0.32	1.1	8
Rang	1 ^{ère}	A	A	B	A	A	B	E	E
	2 ^{ème}	B	E	E	B	E	A	B	B
	3 ^{ème}	E	B	A	E	B	E	A	A

Tableau 1-12: Exemple de la méthode de HARE-itér 1

$$S_{pl}(A) = 11.57, S_{pl}(B) = 8.32, \text{ et } S_{pl}(E) = 9.1$$

Avec ses scores, aucun candidat n'atteint le quota (16) et B est éliminé. Ce qui nous conduit au résultat suivant dans le **tableau 1-13**. Le candidat E avec 17.1 de voix est élu par la méthode de HARE.

Nombre de votants→		5	5	8	0.47	1.1	0.32	1.1	8
Rang	1 ^{ère}	A	A	E	A	A	A	E	E
	2 ^{ème}	E	E	A	E	E	E	A	A

Tableau 1-13: Exemple de la méthode de HARE- itér 2

$$S_{pl}(A) = 11.89, \text{ et } S_{pl}(E) = 17.1$$

1.2.1.5 Méthode de Bucklin

La méthode inventée en 1909 par James Bucklin, un politicien du Colorado est connu sous le nom de système de "Grande Junction" peut être utilisée pour élire un ou plusieurs Gagnants (TIDEMAN, 2007). C'est une méthode qui cumule le nombre de votes qu'un candidat obtient pour chaque position occupée sur les listes de préférence.

Pour la mettre en place, chaque votant doit classer les candidats sur une liste ordonnée de préférence. On commence d'abord à compter, le nombre de voix que chaque candidat occupe en première position sur les bulletins. Si un candidat obtient la majorité absolue il est déclaré vainqueur. Sinon on ajoute aux scores antérieurs des candidats, le score que chaque candidat obtient en occupant la deuxième place sur les listes de préférence. On recompte les voix, ainsi de suite. Le processus s'arrête si un ou des candidats cumulent des scores qui leur donnent la majorité absolue.

En d'autres termes, le dépouillement se déroule par tours. Au tour t , le score provisoire d'un candidat A est le nombre d'électeurs qui placent A entre le rang 1 et le rang t sur leur bulletin. Si au moins un candidat a un score strictement supérieur à la moitié du nombre de votants, alors le candidat possédant le plus haut score est élu. Sinon, on procède au tour de dépouillement $t + 1$.

Par exemple, pour abriter une compétition sportive, 4 villes A, B, C et D se sont portées candidates. Le tableau suivant contient le résultat du vote réalisé entre les villes.

Nombre de votants→		42%	26%	15%	17%
Rang	1 ^{ère}	A	B	C	D
	2 ^{ème}	B	C	D	C
	3 ^{ème}	C	D	B	B
	4 ^{ème}	D	A	A	A

Tableau 1-14: Exemple de la méthode de Bucklin

En appliquant la méthode de Bucklin, lors du premier décompte les villes obtiennent respectivement les scores :

$$S_{Buck1}(A) = 42, S_{Buck1}(B) = 26, S_{Buck1}(C) = 15 \text{ et } S_{Buck1}(D) = 17.$$

Aucune ville n'obtient la majorité absolue. Donc analysons les deuxièmes positions, les villes candidates cumulent les scores suivants :

- $S_c(A) = S_{Buck1}(A) + S_{Buck2}(A) = 42 + 0 = 42,$
- $S_c(B) = S_{Buck1}(B) + S_{Buck2}(B) = 26 + 42 = 68,$
- $S_c(C) = S_{Buck1}(C) + S_{Buck2}(C) = 15 + 26 + 17 = 58,$
- $S_c(D) = S_{Buck1}(D) + S_{Buck2}(D) = 17 + 15 = 32.$

La ville B gagne avec 68% devant la ville C qui obtient 58%.

1.2.1.6 Méthode de Jugement majoritaire

Inventée par deux chercheurs français du CNRS, Michel Balinski et Rida Laraki, est un système de vote par valeur basé sur des appréciations verbales sur une échelle de mention (Très bien, Bien, Assez bien, Passable, Insuffisant et à rejeter). Un électeur peut donner la même mention à plusieurs candidats. Aucune mention n'est obligatoire ; les candidats non évalués reçoivent la mention "A rejeter". (BALINSKI, et al., 2012)

Pour comprendre le fonctionnement de ladite méthode considérons l'exemple suivant. Dans le cadre du respect de l'environnement, une entreprise veut adopter une source d'énergie secondaire. Lors de la réunion de prise de décision, cinq votants ont décidé de faire un choix entre trois alternatives à savoir l'éolien, le solaire ou la biomasse. Chaque votant va soumettre son bulletin de vote comme montrent les tableaux suivants.

Mention→	Très bien	Bien	Assez – Bien	Passable	Insuffisant	À rejeter
Eolien	x					
Solaire	x					
Biomasse						x

Tableau 1-15: Bulletin du votant n°1

Mention→	Très bien	Bien	Assez – Bien	Passable	Insuffisant	À rejeter
Eolien		x				
Solaire	x					
Biomasse				x		

Tableau 1-16: Bulletin du votant n°2

Mention→	Très bien	Bien	Assez – Bien	Passable	Insuffisant	À rejeter
Eolien			x			
Solaire		x				
Biomasse						x

Tableau 1-17: Bulletin du votant n°3

Mention→	Très bien	Bien	Assez – Bien	Passable	Insuffisant	À rejeter
Eolien		x				
Solaire		x				
Biomasse			x			

Tableau 1-18: Bulletin du votant n°4

Mention→	Très bien	Bien	Assez – Bien	Passable	Insuffisant	À rejeter
Eolien	x					
Solaire		x				
Biomasse				x		

Tableau 1-19: Bulletin du votant n°5

Au décompte, on totalise pour chaque candidat les appréciations reçues et on présente la part que chaque appréciation représente dans les votes exprimés. C'est son « profil de mérite » :

Candidat	Très bien	Bien	Assez – Bien	Passable	Insuffisant	À rejeter
Eolien	40%	40%	20%	0%	0%	0%
Solaire	40%	60%	0%	0%	0%	0%
Biomasse	0%	0%	20%	40%	40%	0%

Tableau 1-20: Profil de mérite des sources d'énergie candidates

Cela se présente graphiquement sous la forme d'un histogramme cumulé dont le total correspond à 100 % des voix exprimées :

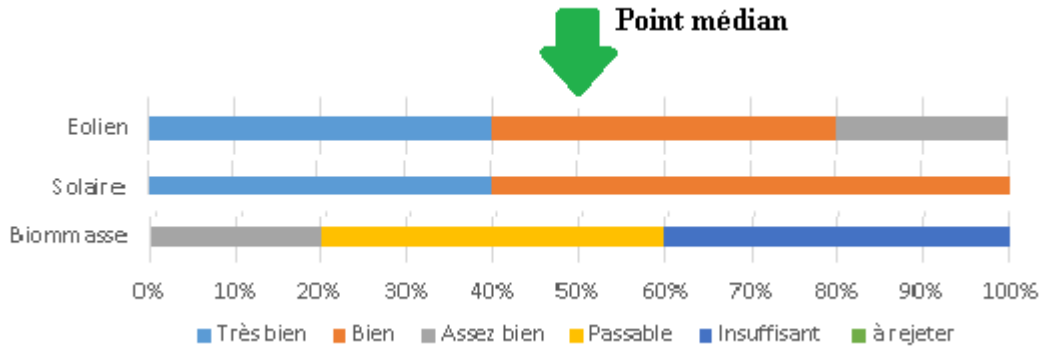


Figure 1-1: Jugement majoritaire- Histogramme

Pour chaque candidat, on détermine la mention médiane, qu'on appelle « mention majoritaire ». Ce mode de sélection signifie en effet que la majorité des électeurs (plus de 50 %) jugent qu'il mérite au moins cette mention, et la majorité jugent qu'il mérite cette mention ou mieux.

Dans l'exemple, la mention majoritaire des candidats *Eolien* et *Solaire* est « *bien* » car dans les deux cas plus de 50 % des électeurs jugent qu'il mérite « *bien* » ou mieux.

Le candidat élu est celui qui obtient la meilleure « mention majoritaire ».

Pour départager deux candidats ayant la même mention majoritaire, on utilise deux principes :

- Critère de victoire : avoir le plus d'électeurs attribuant strictement plus que la mention majoritaire commune
- Critère de défaite : avoir le plus d'électeurs attribuant strictement moins que la mention majoritaire commune

Pour notre exemple on calcule ces deux valeurs pour nos deux candidats à départager.

Candidat	Vote pour les mentions Supérieures	Mention majoritaire	Vote pour les mentions Inférieures
<i>Eolien</i>	40%	<i>Bien</i>	20%
<i>Solaire</i>	40%	<i>Bien</i>	0%

Tableau 1-21: Départage des mentions

La plus grande des 4 valeurs détermine le résultat, afin de satisfaire le plus grand nombre d'électeurs. Ici, c'est le nombre de votes pour des mentions inférieures à la mention majoritaire pour la source *Eolienne*, donc perd et la source *Solaire* gagne. Ce résultat se lit

graphiquement sur l'histogramme : c'est l'extrémité de zone " **Bien**" la plus proche de la médiane.

1.2.2 Les procédures CONDORCET Cohérentes

Ce sont des procédures de vote dont leur principe de base est d'élire le vainqueur de Condorcet s'il existe. En théorie du choix social, un vainqueur de Condorcet est une option préférée majoritairement à toutes les autres options prises une par une selon la règle majoritaire. Généralement elles ont été mises au point pour résoudre le paradoxe de Condorcet. Parmi ces procédures nous pouvons citer celles décrites dans les points suivants

1.2.2.1 Méthode de CONDORCET

C'est une méthode mise au point en 1785 par le marquis de Condorcet, mathématicien et philosophe français. Dans son Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix, Condorcet met en évidence le fait que le vote à la pluralité peut très bien ne pas représenter les désirs des électeurs. (BOURSIN, 2004) (BRAMS, et al., 2002)

Chaque votant classe les candidats par ordre de préférence. Le dépouillement consiste à simuler l'ensemble des duels possibles entre les candidats. Pour chaque paire de candidats, on détermine le nombre de votants ayant voté pour l'un ou l'autre en vérifiant, sur chaque liste de préférence, comment l'un était classé par rapport à l'autre. Si un candidat remporte l'ensemble de ces duels, il est désigné gagnant. Ce principe permet de choisir un candidat bénéficiant généralement d'un haut degré de satisfaction de l'électorat. Les équations suivantes donnent une explication mathématique du principe de Condorcet.

$$x_i P x_j = \begin{cases} 1, & \text{si } Rang(x_i) < Rang(x_j) \\ 0, & \text{si } Rang(x_j) < Rang(x_i) \end{cases}, \quad (F1.6)$$

$$S_{pref}(x_i, x_j) = \sum_{i=1}^n x_i P_n x_j. \quad (F1.7)$$

$$Duel(x_i, x_j) = \begin{cases} x_i, & \text{si } S_{pref}(x_i, x_j) > S_{pref}(x_j, x_i) \\ x_j, & \text{si } S_{pref}(x_i, x_j) < S_{pref}(x_j, x_i) \end{cases}, \quad (F1.8)$$

où n est le nombre de votants, $Rang(x_j)$ détermine le rang du candidat x_j sur la liste de préférence d'un votant ; $S_{pref}(x_i, x_j)$: somme des préférences de x_i , par rapport x_j

$S_{pref}(x_i, x_j)$ détermine la préférence totale du candidat x_i par rapport au candidat x_j de tous les votants et $Duel(x_i, x_j)$: retourne le résultat du duel entre les deux candidats x_i et x_j .

Par exemple, le tableau suivant présente les préférences d'un groupe de 7 informaticiens pour choisir le meilleur antivirus (A : Avast, B : Eset, C : Norton) à installer dans leur laboratoire.

Votant→		v_1	v_2	v_3	v_4	v_5	v_6	v_7
Rang	1 ^{ère}	C	C	B	B	A	A	A
	2 ^{ème}	B	B	C	C	B	C	C
	3 ^{ème}	A	A	A	A	C	B	B

Tableau 1-22: Exemple de la méthode de CONDORCET

En appliquant le principe sur l'exemple précédent et en prenant '>' pour représenter la relation de préférence d'un candidat à l'autre, nous obtenons le tableau des duels :

Duel→	A versus B	A versus C	B versus C
	A>B = 3	A>C = 3	B>C = 3
	B>A = 4	C>A = 4	C>B = 4
Gagnant→	B	C	C

Tableau 1-23: Exemple de CONDORCET- les duels

Par conséquent, l'antivirus Norton sera installé car C remporte tous ces duels donc c'est le gagnant.

Condorcet, va trouver une limite à sa méthode proposée, qui sera appelée paradoxe de Condorcet et va jouer un rôle très important dans la théorie de vote. Ce paradoxe stipule qu'il est possible, « lors d'un vote où l'on demande aux votants de classer trois candidats (A, B et C) par ordre de préférence, qu'une majorité de votants préfère A à B, qu'une autre préfère B à C et qu'une autre préfère C à A. Les décisions prises à une majorité populaire par ce mode de scrutin ne seraient donc pas cohérentes avec celles que prendrait un individu rationnel ».

Considérons par exemple une assemblée de 60 votants ayant le choix entre trois propositions A, B et C. Les préférences se répartissent ainsi dans le tableau suivant :

Nb votants→	23	17	2	10	8
Rang					
1 ^{ère}	A	B	B	C	C
2 ^{ème}	B	C	A	A	B
3 ^{ème}	C	A	C	B	A

Tableau 1-24: Exemple de Paradoxe de Condorcet

En notant $A > B$, le fait que A est préféré à B, dans les comparaisons majoritaires par paire, on obtient les duels entre :

- A et B ($A > B = 33$; $B > A = 27$) donne A est préféré à B
- B et C ($B > C = 42$; $C > B = 18$) donne B est préféré à C
- A et C ($A > C = 25$; $C > A = 35$) donne C est préféré à A

Ce qui conduit à la contradiction interne $A > B > C > A$. Les points suivants expliquent des méthodes qui ont été mis au point pour résoudre ce conflit de circularité.

1.2.2.2 Méthode de BLACK

Elle est utilisée pour lever un conflit généré par le paradoxe de Condorcet. Elle se base fondamentalement sur les méthodes de Condorcet et de Borda. Ce système tire son nom de Duncan Black, qui a été le premier à la proposer. (DUCAN, 1958) (BRAMS, et al., 2002) (MCLEAN, 1996)

Avec n candidats, on applique d'abord le principe de Condorcet. Si un candidat gagne toutes ses confrontations, c'est le gagnant de Condorcet et la méthode Black s'arrête. Si ce n'est pas le cas, on applique la méthode Borda pour trouver un gagnant avec les règles suivantes : on compte le nombre de fois où le candidat X est arrivé en tête, que l'on multiplie par n , puis le nombre de fois où il est arrivé en seconde position que l'on multiplie par $n - 1$, ainsi de suite. Le candidat dont le score est le plus élevé gagne.

Considérons par exemple une assemblée de 60 votants ayant le choix entre trois propositions A, B et C. Les préférences se répartissent ainsi dans le tableau suivant :

Nombre de votants→		23	17	10	2	8
Rang	1 ^{ère}	A	B	C	B	C
	2 ^{ème}	B	C	A	A	B
	3 ^{ème}	C	A	B	C	A

Tableau 1-25: Exemple de la méthode de Black

En appliquant la méthode Condorcet on obtient le **Tableau I-26** avec les résultats des duels.

Duel→	A versus B	A versus C	B versus C
	A>B = 33	A>C = 25	B>C = 42
	B>A = 27	C>A = 35	C>B = 28
Gagnant→	A	C	B

Tableau I-26: Exemple de Black – résultat des duels

En analysant le tableau précédent, aucun candidat gagne tous ces duels. Ce qui contredit le principe de Condorcet. En appliquant le reste du principe de la méthode de Black nous obtenons le **Tableau I-27**.

Candidat→		A	B	C	Points
Rang	1 ^{ère}	23	19	18	3
	2 ^{ème}	12	31	17	2
	3 ^{ème}	25	10	25	1
Score		118	129	113	

Tableau I-27: Exemple de la Méthode Black – Application de Borda

Le candidat B obtient le plus grand score, **129** points donc il est de gagnant de la méthode de Black.

1.2.2.3 La méthode de NANSON

C'est une méthode itérative visant à sélectionner les meilleurs candidats, mise au point en 1907 par Edward John Nanson. Cette méthode conduit à un gagnant qui coïncide à celui de Condorcet. (MCLEAN, 1996)

Initialement c'est le même principe que la méthode de Borda. Ensuite on élimine successivement les candidats ayant le plus mauvais score du compte de Borda. Le candidat éliminé lors de la dernière itération est vainqueur de Nanson.

Par exemple lors d'une élection à 4 candidats, le vote a donné les résultats suivants :

Nombre de votants→		30%	30%	25%	15%
Rang	1 ^{ère}	B	B	A	A
	2 ^{ème}	A	A	C	D
	3 ^{ème}	C	D	D	C
	4 ^{ème}	D	C	B	B

Tableau 1-28: Exemple de méthode de NANSON

En appliquant ce principe de la méthode de Nanson, nous obtenons les scores suivants :

- $S_{Borda}(A) = (40*4) + (60*3) = 340$ points,
- $S_{Borda}(B) = (60*4) + (40*1) = 280$ points,
- $S_{Borda}(C) = (25*3) + (45*2) + (30*1) = 195$ points,
- $S_{Borda}(D) = (15*3) + (55*2) + (30*1) = 185$ points

Le candidat D avec le plus petit score est éliminé. Et un nouveau calcul de score de Borda donne les valeurs suivantes :

- $S_{Borda}(A) = (40*3) + (60*2) = 240$ points,
- $S_{Borda}(B) = (60*3) + (40*1) = 220$ points,
- $S_{Borda}(C) = (40*2) + (60*1) = 140$ points,

Le candidat C obtient le plus petit score et est éliminé. Un dernier calcul des scores donne :

- $S_{Borda}(A) = (40*2) + (60*1) = 140$ points,
- $S_{Borda}(B) = (60*2) + (40*1) = 160$ points,

Le candidat A est éliminé avec le plus petit score par rapport au gagnant B.

1.2.2.4 Méthode de KEMENY-YOUNG

Créée par Kemeny en 1959 puis améliorée par Young en 1995 c'est une extension de la méthode de Condorcet.

Même principe que Condorcet au début, mais pour n candidats on fait toutes les combinaisons de classement possibles (soit $n !$ possibilités). Pour chaque classement on calcule un score en utilisant les résultats des duels. Le classement qui a le plus grand score sera la sélection gagnante. Si on veut choisir un seul gagnant, ça sera le premier dudit

classement qui coïncide avec le vainqueur de Condorcet. (DAVENPORT, et al., 2004) (CONITZER, et al., 2006)

Votant→		v_1	v_2	v_3	v_4	v_5	v_6	v_7
Rang	1 ^{ère}	C	C	B	B	A	A	A
	2 ^{ème}	B	B	C	C	B	C	C
	3 ^{ème}	A	A	A	A	C	B	B

Tableau 1-29: Exemple de la méthode de KEMENY-YOUNG

C'est-à-dire, dans le tableau antérieur, nous avons trois (3) candidats, A, B et C. Les combinaisons de classement sont : ABC, ACB, BAC, BCA, CAB et CBA. (Où ABC signifie qu'on préfère le candidat A aux candidats B et C et aussi que B à C). Le tableau suivant contient le résultat du calcul de score de chaque arrangement.

Combinaison	Duels			Score
A B C	A>B=3	A>C=3	B>C=3	9
A C B	A>C=3	A>B=3	C>B=4	10
B A C	B>A=4	B>C=3	A>C=3	10
B C A	B>C=3	B>A=4	C>A=4	11
C A B	C>A=4	C>B=4	A>B=3	11
C B A	C>B=4	C>A=4	B>A=4	12

Tableau 1-30: Exemple de la méthode de KEMENY-YOUNG-Duels

Le gagnant de Kemeny est l'arrangement **C B A** avec un score de 12. En plus si on cherche un seul vainqueur ça sera C qui est le vainqueur de Condorcet.

1.2.2.5 Méthode de COPELAND

C'est une extension de la méthode de Condorcet. Elle est souvent utilisée comme mécanisme pour briser le paradoxe de Condorcet. (HENRIET, 1985) (SALLES, 2006)

C'est le même principe initial que la méthode de Condorcet, mais on attribue +1 à un candidat s'il remporte son duel face à un autre ou -1 s'il perd. Pour chaque candidat on fait la somme de ses scores. Le gagnant de Copeland sera celui qui va totaliser le plus grand score.

Par exemple, quarante-cinq votants classent cinq candidats A, B, C, D et E. Les scores des 45 classements obtenus sont :

Nombre de votants→		5	5	8	3	7	2	7	8
Rang	1 ^{ère}	A	A	B	C	C	C	D	E
	2 ^{ème}	C	D	E	A	A	B	C	B
	3 ^{ème}	B	E	D	B	E	A	E	A
	4 ^{ème}	E	C	A	E	B	D	B	D
	5 ^{ème}	D	B	C	D	D	E	A	C

Tableau 1-31: Exemple de la méthode de Black

En appliquant le principe initial de Copeland, on obtient **Tableau 1-32** avec les duels possibles.

Duel→	A versus B	A versus C	A versus D	A versus E	B versus C
	A>B = 20	A>C = 26	A>D = 30	A>E=22	B>C=16
	B>A = 25	C>A = 19	D>A = 15	E>A=23	C>B=29
Gagnant→	B	A	A	E	C
Duel→	B versus D	B versus E	C versus D	C versus E	D versus E
	B>D=33	B>E=18	C>D=17	C>E=24	D>E=14
	D>B=12	E>B=27	D>C=28	E>C=21	E>D=31
Gagnant→	B	E	D	C	E

Tableau 1-32: Exemple de la méthode de COPELAND – résultat des duels

En analysant le tableau précédent, le candidat **E** a gagné ses duels à l'exception de celui contre le candidat **C**. Ce qui contredit le principe du gagnant de Condorcet.

En remplaçant les résultats des duels par les (+1/-1) comme préconise COPELAND On obtient le **Tableau 1-33**.

	A	B	C	D	E	Score
A		-1	+1	+1	-1	S(A) = 0
B	+1		-1	+1	-1	S(B) = 0

C	-1	+1		-1	+1	S(C) = 0
D	-1	-1	+1		-1	S(D) = -1
E	+1	+1	-1	+1		S(E) = +1

Tableau 1-33: Exemple de la méthode de COPELAND-Scores

En conclusion, le candidat **E** avec un score de +1 est le gagnant de COPELAND.

1.2.2.6 Méthode de MINIMAX

Appelée aussi méthode de KRAMER-SIMPSON une extension de la méthode de Condorcet. Elle est souvent utilisée comme mécanisme pour briser le paradoxe de Condorcet. (BRAMS, et al., 2007)

Le principe de base est celui de Condorcet, mais on associe à chaque candidat A le score suivant : pour chaque autre candidat Y, on calcule N (A, Y) le nombre de préférence du candidat A au candidat Y. Le score de Simpson pour le candidat A est le minimum des N (A, Y). Le candidat élu est celui qui a le plus haut score de Simpson. Autrement dit, on regarde le pire des scores qu'il ferait dans ses différents face-à-face, puis on déclare élu celui des candidats dont le pire des scores est le meilleur.

En appliquant le même principe à l'exemple antérieur. On obtient le tableau des scores qui suit. Le candidat E avec un score de 21 est le gagnant.

	A	B	C	D	E	Score
A		20	26	30	22	S(A) = 20
B	25		16	33	18	S(B) = 16
C	19	29		17	24	S(C) = 17
D	15	12	28		14	S(D) = 12
E	23	27	21	31		S(E) = 21

Tableau 1-34: Exemple de la méthode de MINIMAX

1.2.2.7 Méthode de DODGSON

C'est une extension pour résoudre le paradoxe de la méthode de Condorcet. Elle a été inventée par C. Dodgson appelé aussi Lewis, en 1876.

On part initialement du principe de Condorcet, s'il n'y a pas de vainqueur. On calcule le score de Dodgson $D(x)$ de tout candidat x qui est nombre minimum d'échange de candidats consécutifs dans les préférences des votants pour que x devienne vainqueur de Condorcet. Celui qui a le plus petit score de Dodgson doit être désigné comme le vainqueur. C'est-à-dire si on a $A > C > B$ pour placer B à la tête de la liste, il faut faire d'abord $A > B > C$ puis $B > A > C$ donc, $D(B)=2$. On Calcule le quota m nécessaire pour avoir la majorité absolue à partir du nombre de votants.

On dresse un tableau carré avec les résultats de tous les duels avec une colonne supplémentaire pour y noter le nombre de conversions nécessaires pour devenir le gagnant. (KLAMLER, 2004)

Par exemple, soit un comité de 10 votants qui doit choisir un candidat parmi 4 proposées A, B, C et D. Le tableau suivant contient les préférences exprimées par les votants.

Nb votants→	3	2	1	1	1	1
1 ^{ère}	A	B	D	A	D	C
2 ^{ème}	B	D	C	C	B	D
3 ^{ème}	C	C	A	B	A	B
4 ^{ème}	D	A	B	D	C	A

Tableau 1-35: Exemple de la méthode de Dodgson

Le résultat des duels entre les candidats se présente dans le tableau suivant.

Duel→	A versus B	A versus C	A versus D	B versus C	B versus D	C versus D
	A>B=5	A>C=5	A>D=4	B>C=6	B>D=6	C>D=5
	B>A=4	C>A=4	D>A=5	C>B=3	D>B=3	D>C=4

Tableau 1-36: Exemple de Dodgson-Duels

Le score de Dodgson pour A est $D(A)=2$, car pour battre D, le candidat A doit faire deux échanges pour le dernier votant (entre B et A puis entre D et A).

Les scores de Dodgson pour B, C, et D sont $D(B)=1$; $D(C)=3$ et $D(D)=3$. Par conséquent B est le gagnant de Dodgson.

1.3 Conclusion

Le rôle d'une procédure de vote est de mettre en commun les préférences de chaque électeur, et de les traduire en décisions pour la collectivité.

Dans ce chapitre nous avons décrit des procédures sans classement et les procédures par classement. Les procédures par classement se divisent en deux sous-groupes : les Condorcet Consistantes qui choisissent toujours le gagnant de Condorcet comme gagnant s'il existe et les Non Condorcet Consistantes qui peuvent élire un gagnant outre que celui de Condorcet. Les principales procédures de vote ont été présentées, mais ils existent d'autres procédures de vote qui ne sont pas décrites dans ce document. Une bonne procédure de vote doit être simple et limiter autant que possible les possibilités de manipulation (vote non sincère). De plus, elle doit être capable d'élire s'il existe le gagnant de Condorcet.

Chaque procédure comporte des avantages et des inconvénients. Ainsi les procédures de pluralité et ses dérivées sont simples et peu coûteuses à mettre en place. Par contre elles peuvent aboutir sur l'élection d'un candidat ou d'une option qui déplaît à une grande partie des votants. Par contre, ce sont les meilleures procédures de vote s'il n'y a que deux candidats en lice.

Les méthodes de Borda, de Condorcet et ses variantes, et le vote par élimination permettent de choisir un candidat qui engendre un haut degré de satisfaction des votants.

Leur mise en pratique dans la vie réelle constitue un véritable challenge. L'utilisation de NTIC permet de surmonter une grande partie de ces difficultés.

S'il y a plus de deux candidats, les chercheurs se partagent sur le choix de la procédure idéale. Pour ceux qui sont partisans de l'élection du vainqueur de Condorcet, les deux procédures fortement recommandées sont les méthodes de Copeland et de Kemeny (PALHA, 2017) (FELSENTHAL, et al., 2018). Le choix entre ces deux, dépend de l'importance qu'on accorde aux critères concernant la bonne procédure car les deux exigent un classement de tous les candidats par ordre d'importance pour les votants. Cependant la méthode de Copeland s'avère plus facile à expliquer que celle de Kemeny aux électeurs profanes. La procédure de Kemeny, en revanche, est plus discriminatoire que celle de Copeland lorsque le nombre de candidats est relativement faible, et elle est probablement plus complexe dans la détermination du vainqueur final et moins vulnérable au vote tactique. Cette procédure a aussi l'avantage de dégager une préférence médiane qui peut être considéré comme meilleur compromis pour les votants. Pour plus d'information j'invite le lecteur à consulter (YOUNG, 1995)

Pour ceux qui ne considèrent pas la cohérence de Condorcet comme une propriété essentielle des systèmes de vote et qui se préoccupent du principe de participation, la procédure de Borda semble nettement meilleure aux autres procédures par classement malgré sa vulnérabilité à la manipulation. (FISHBURN, 2015) (SAARI, 1995) (SAARI, 2012)

La procédure qui peut servir de trait d'union entre les deux tendances est celle de Black, car elle profite des avantages des principes de Borda et celle de Condorcet.

La procédure du Jugement majoritaire, introduit un autre concept à la théorie du vote. Selon (BALINSKI, et al., 2012) elle est moins manipulable et résiste à plusieurs paradoxes de la théorie du vote. Mais, elle est difficile à mettre en place et à expliquer aux votants.

CHAPITRE 2 :

2. Systèmes de recommandation

2.1 Introduction aux systèmes de recommandation

Avec l'avènement d'internet pour le grand public et les évolutions technologiques, entre autres, la masse de données à exploiter ou à analyser est devenue très volumineuse. Si bien qu'il est devenu difficile de savoir quelles sont les données à rechercher et où les trouver. Des techniques informatiques ont été développées pour faciliter cette recherche ainsi que l'extraction des informations pertinentes. Les systèmes de recommandation sont pour beaucoup d'entre nous des entités mystérieuses qui semblent deviner nos pensées par exemple Netflix, qui nous suggère des films et Amazon qui nous propose des produits à acheter. Depuis leurs débuts, les outils de recommandation ont été améliorés et raffinés, et ils ne cessent d'enrichir l'expérience utilisateur moyennant des ensembles complexes.

Les systèmes de recommandation sont largement utilisés dans le domaine des réseaux sociaux. Par exemple, YouTube utilise un système de recommandation pour suggérer aux utilisateurs différentes vidéos susceptibles de les intéresser, en fonction de leur historique de navigation dans le système, dans le but d'améliorer l'expérience utilisateur. Facebook possède un système de recommandation basé sur le filtrage collaboratif, qui recommande des amis basés sur les amis communs aux utilisateurs. Depuis quelques années les systèmes de recommandation jouent un rôle de plus en plus important dans plusieurs domaines mais leur utilisation dans les GDSS (Group Decision Support System) est pratiquement inexplorée surtout dans les outils d'aide à la facilitation.

L'un des résultats visés dans cette recherche sur les systèmes de recommandation est de créer un système efficace qui collecte les données nécessaires pour formuler des recommandations sur les différentes procédures de vote totalement transparentes pour l'utilisateur, qui est le facilitateur de la réunion de prise de décision.

2.2 Définitions des systèmes de recommandation

Plusieurs définitions ont été proposées par des auteurs selon leurs objectifs. Les définitions suivantes ont été énoncées :

- "Un système de recommandation est un système dont la tâche principale est de choisir certains objets qui répondent aux besoins des utilisateurs, chacun de ces objets étant stocké dans un système informatique et caractérisé par un ensemble d'attributs." (WANG, 1998)

- « Un système de recommandation est une technologie de filtrage d'informations personnalisée, utilisée pour prédire si un utilisateur particulier aime un élément particulier (problème de précondition) ou identifier un ensemble de N éléments pouvant intéresser certains utilisateurs (problème de recommandation. Top-N.) » (KARYPIS, 2001)
- « Un système de recommandation est un ensemble de techniques de récupération d'informations qui tentent de découvrir l'intérêt des utilisateurs pour certains objets, afin de leur proposer un ensemble d'objets associés, liés à leur profil, qui pourraient les intéresser. (CRESPO, 2011)
- . Dans le cadre de notre travail nous donnons la définition suivante d'un système de recommandation de procédures de vote :

« Un système de recommandation de procédures de vote est un ensemble de techniques de récupération d'informations qui essaient de découvrir la procédure de vote adéquate à appliquer selon le contexte de prise de décision, pour simplifier la tâche du facilitateur »

2.3 Différentes techniques de collecte d'informations

Un système de recommandation devrait fournir un mécanisme permettant de collecter autant d'informations que possible sur le profil des utilisateurs afin de découvrir leurs goûts et leurs intérêts pour certains objets, puis de générer de meilleures recommandations. Ce processus s'appelle "feedback". Comme indiqué dans (RESNICK, et al., 1997), (ADOMAVICIUS, et al., 2005) et (ZIEGLER, et al., 2005), le processus de feedback est classé en deux types : collecte implicite et explicite.

Les deux approches ont des limites qui peuvent être surmontées en les combinant dans un système de recommandation. (JAWAHEER, et al., 2010)

2.3.1 Collecte implicite

Ce processus consiste à évaluer les objets, sans les interventions des utilisateurs. C'est-à-dire cette évaluation est faite à l'insu de l'utilisateur, grâce à la capture des informations obtenues à partir des actions effectuées par les utilisateurs dans l'application. Par exemple, lorsque l'utilisateur accède à une actualité ou lit un article en ligne, en fonction du temps de lecture, le système peut estimer l'intérêt de l'utilisateur sur ce contenu.

2.3.2 Collecte explicite

Cette technique repose sur le fait que l'utilisateur valorise le contenu en attribuant un score pour indiquer ses intérêts. Autrement dit les commentaires explicites donnent aux utilisateurs un mécanisme leur permettant d'exprimer sans équivoque leur intérêt pour les objets.

Par exemple, *Amazon*, *Film affinity*, *Movilens* et d'autres utilisent le système de notation « cinq étoiles », qui permet aux utilisateurs d'évaluer les produits qui les intéressent. D'autre part, les réseaux sociaux tels que Facebook, YouTube et autres utilisent le système de classement « J'aime » pour évaluer le contenu.

2.4 Classification des systèmes de recommandation

Plusieurs facteurs entrent en considération afin de catégoriser les systèmes de recommandation.

- La connaissance de l'utilisateur (c'est à dire son profil en fonction de ses goûts).
- Le positionnement d'un utilisateur par rapport aux autres (la notion de classes ou réseaux d'utilisateurs).
- La connaissance des items ou objets à recommander.
- La connaissance des différentes classes d'items à recommander

De ces facteurs sont produits divers types de recommandations dont les plus utilisés dans la littérature sont le filtrage basé sur le contenu, le filtrage collaboratif et filtrage hybride

2.4.1 Filtrage basé sur le contenu ou recommandation objet

Les systèmes de recommandation basés sur le contenu s'appuient sur des évaluations effectuées par un utilisateur sur un ensemble d'objets ou items. L'objectif est alors de comprendre les motivations l'ayant conduit à juger comme pertinent ou non un item donné. Il est également connu sous le nom de système de recommandation non collaboratif. Afin de savoir si un élément est similaire à un autre, des mots-clés sont recherchés pour l'élément que l'utilisateur a qualifié. La tâche consiste à déterminer quels éléments du catalogue coïncident le mieux avec les préférences de l'utilisateur. Une telle approche ne requiert pas une grande communauté d'utilisateurs ou un gros historique d'utilisation du système comme illustré sur la *Figure 2-1*.

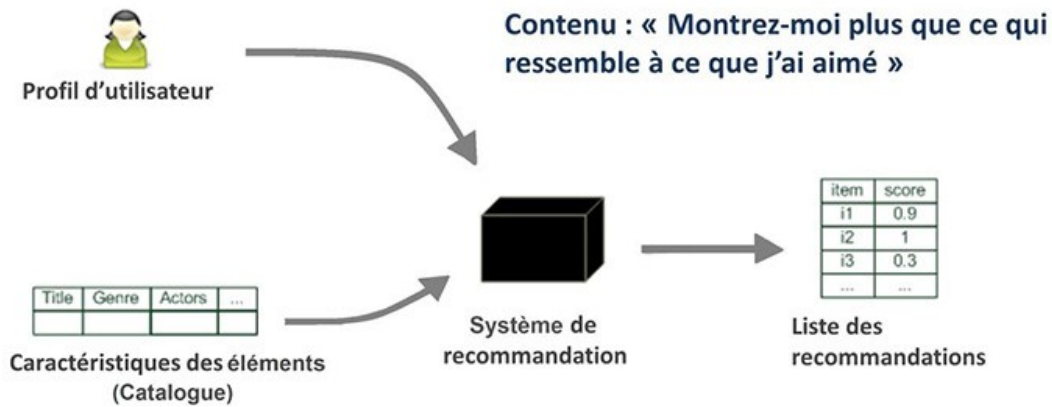


Figure 2-1: Filtrage basé sur le contenu (JANNACH, et al., 2010)

La manière la plus simple de décrire un catalogue d'éléments est d'avoir une liste explicite des caractéristiques de chaque élément (on parle aussi d'attributs, de profil d'élément, etc.). Pour un livre par exemple, on peut utiliser le genre, le nom des auteurs, l'éditeur ou toute autre information relative au livre, puis stocker ces caractéristiques dans une base de données.

Le profil de l'utilisateur est exprimé sous forme d'une liste d'intérêts basée sur les mêmes caractéristiques. La coïncidence entre les caractéristiques des éléments et le profil de l'utilisateur peut être mesurée de différentes manières :

- l'indice de **Dice** ou d'autres mesures de similarité, (BAEZA-YATES, et al., 2011).
- le TF-IDF (Term Frequency-Inverse Document Frequency), (SALTON, et al., 1975)
- les techniques basées sur la similarité des espaces vectoriels (les approches bayésiennes (PAZZANI, et al., 2007), les arbres de décision, etc.) couplées avec des techniques statistiques, lorsqu'il y a trop de mots-clés.

Les approches basées sur le contenu présentent plusieurs avantages et inconvénients. Les points forts des approches basées sur le contenu sont :

- **Autonomie de l'utilisateur** : les techniques de recommandation basées sur le contenu traitent chaque utilisateur de façon indépendante. Ainsi, seules les évaluations de l'utilisateur lui-même sont prises en compte pour construire son profil utilisateur et faire la recommandation, ce qui n'est pas le cas pour les approches utilisant le filtrage collaboratif.
- **Prise en compte immédiate d'un nouvel item** : le filtrage basé sur le contenu peut recommander des items nouvellement introduits dans la base avant même qu'ils

reçoivent une évaluation de la part d'un utilisateur, au contraire des approches collaboratives qui ne peuvent recommander un item que s'il a été préalablement évalué par un groupe d'utilisateurs. Les auteurs (EKSTRAND, et al., 2011) ont démontré qu'il faut qu'un item ait reçu au minimum 20 notes pour que le filtrage collaboratif arrive à le recommander de manière pertinente.

- Il n'y a pas de problème de démarrage à froid lorsqu'un nouvel élément est ajouté au catalogue ou de faible densité puisqu'il s'agit de faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments ;
- Il n'y a pas de problème de parcimonie (sparsity) parce que la modélisation de l'information est présente dans les caractéristiques du document et ne nécessite pas des informations fournies par d'autres utilisateurs ;

Cependant les approches de recommandation basées sur le contenu présentent aussi de nombreux inconvénients :

- **Limite de l'analyse du contenu** : une limite naturelle de la recommandation basée sur le contenu est la nécessité de disposer d'une représentation variée et riche du contenu des items, ce qui n'est pas toujours le cas. La précision des recommandations est liée à la quantité d'informations dont dispose le système pour discriminer les items appréciés de ceux non appréciés par l'utilisateur (LOPS, et al., 2011).
- **Sur-spécialisation (Over-specialization)** : le système ne peut recommander que les items qui sont similaires au profil utilisateur. L'utilisateur ne peut donc recevoir que des recommandations proches des items qu'il a notés ou observés par le passé (ADOMAVICIUS, et al., 2005). Or, la diversité des recommandations est souvent appréciée et s'avère être un critère d'évaluation important des systèmes de recommandation. Idéalement, l'utilisateur doit recevoir des recommandations pertinentes et diversifiées. Par exemple, il n'est pas intéressant de recommander toutes les chansons de Jacques Brel à un utilisateur qui a aimé l'une de ses chansons.
- **Intégration d'un nouvel utilisateur non immédiate** : un utilisateur doit évaluer un certain nombre d'items avant que le système ne puisse interpréter ses préférences et lui fournir des recommandations pertinentes (RICCI, et al., 2011). Ce problème est connu dans la littérature sous le nom du problème de démarrage à froid pour les utilisateurs (user cold start).

2.4.2 Filtrage Collaboratif ou Recommandation sociale

Le filtrage collaboratif est une approche basée sur le partage d'opinions entre les utilisateurs. Il reprend le principe du "bouche à oreille" pratiqué depuis toujours par les humains pour se construire une opinion sur un produit ou un service qu'ils ne connaissent pas (SCHAFER, 2007). L'hypothèse fondamentale de cette méthode est que les opinions des autres utilisateurs peuvent être utilisées pour fournir une prédiction raisonnable de la préférence de l'utilisateur actif sur un item qu'il n'a pas encore noté. Ces méthodes supposent que si des utilisateurs ont les mêmes préférences sur un ensemble d'items, alors ils auront probablement les mêmes préférences sur un autre ensemble d'items qu'ils n'ont pas encore notés.

La Figure 2-2 illustre ce processus

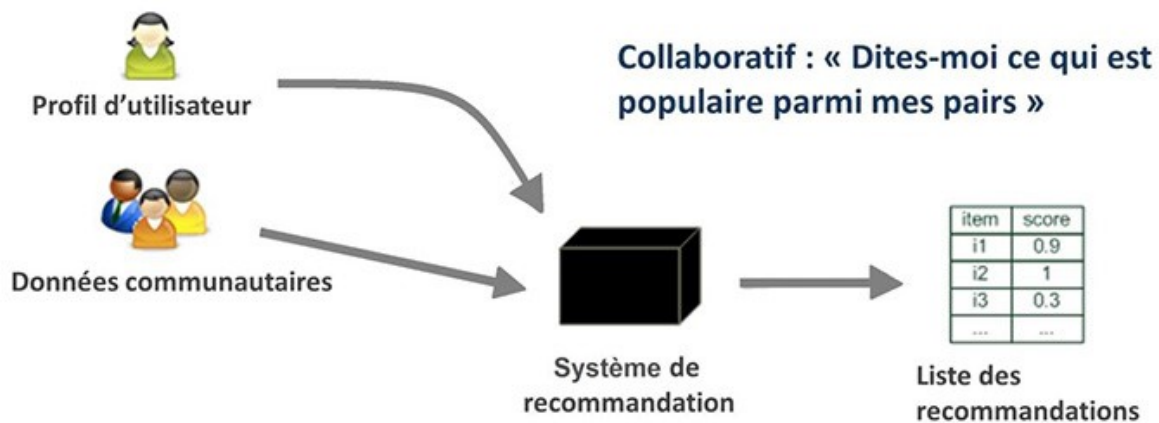


Figure 2-2: Filtrage collaboratif (JANNACH, et al., 2010)

Dans les systèmes de recommandation collaboratifs, il existe deux types d'algorithmes de filtrage :

- **Memory-based, ou basé sur la mémoire** algorithme basé sur la mémoire qui utilise la base de données utilisateur entière pour faire des prédictions,
- **Model-based, ou basé sur le modèle** algorithme basé sur les modèles qui utilisent la base de données utilisateur pour estimer ou apprendre un modèle, qui sera ensuite utilisé pour les prédictions.

2.4.2.1 Méthodes basées sur la mémoire

L'idée de base des systèmes basés sur la mémoire est la suivante :

- Le système maintient un profil utilisateur, c'est-à-dire un enregistrement des intérêts (aussi bien positifs que négatifs) de l'utilisateur pour certains articles.

- Puis il compare ce profil avec les profils des autres utilisateurs, et pèse chaque profil en fonction de son degré de similarité avec le profil de l'utilisateur considéré.
- Enfin, il considère un ensemble des profils les plus similaires (ou les plus opposés), et utilise l'information qu'ils contiennent pour recommander à l'utilisateur (ou mettre en garde contre) des articles qu'il n'a pas encore évalués.

Pour prédire la pertinence d'un article pour un utilisateur, on calcule donc la moyenne des notes données aux articles par les utilisateurs ayant les mêmes goûts (ou des goûts opposés), en utilisant des poids différents selon la mesure de similarité entre utilisateurs comme *Pearson*, *Bayes* et *vote moyen pondéré de Bayes*. Voir (CANDILLIER, 2001) pour plus de détails.

2.4.2.2 Méthodes basées sur les modèles

D'un point de vue probabiliste, la tâche du filtrage collaboratif peut être vue comme le calcul de la valeur la plus probable d'un vote (rating), étant donné ce que nous savons à propos de l'utilisateur, et le modèle que l'on aura construit à partir de la base de données utilisateur. Dans cette technique on peut utiliser deux approches pour déterminer les similarités : le modèle de cluster et le modèle de réseau Bayésien.

2.4.2.2.1 Modèle de Cluster

Un algorithme de clustering cherche à faire émerger la structure d'un ensemble de données en déterminant des groupes d'éléments proches selon des relations de distances préétablies. Le but est de partitionner un ensemble de données $D = \{D_1, \dots, D_n\}$ en K sous-ensembles (ou clusters)

$$C = \{C_1, \dots, C_k\} \text{ tels que : } \begin{cases} \forall l \in \{1, \dots, K\}, C_l \neq \emptyset \\ \forall m \in \{1, \dots, K\}^2, C_i \cap C_m = \emptyset \\ \bigcup_{i=1}^k C_i = D \end{cases} \quad (F2. 1)$$

De très nombreuses méthodes de clustering, dont l'état de l'art est donné dans (BERKHIN, 2006), ont été proposées dans la littérature, chacune présentant des caractéristiques propres qui la rend plus ou moins adaptée à telle ou telle application. Une étude intéressante sur l'utilisation des méthodes de clustering dans le domaine du filtrage de l'information est donnée dans (LAMPRIER, 2008)]. On distingue les méthodes hiérarchiques (JOHNSON, 1967), dont le principe est de créer un arbre dans lequel chaque nœud correspond à un cluster regroupant des clusters de plus bas niveau pour fournir une hiérarchie représentative de la structure du jeu de données concerné, des méthodes non-hiérarchiques (ou Flat Clustering) qui ne produisent qu'un seul niveau de clusters. Le principal avantage des méthodes hiérarchiques est qu'elles ne nécessitent pas la détermination du nombre de clusters a priori (WILLETT, 1988),

contrairement aux méthodes non-hiérarchiques qui requièrent bien souvent l’instanciation d’un grand nombre de paramètres (nombre de clusters, taille des clusters, critères d’optimisation, etc.). Toutefois, ces méthodes présentent elles aussi leurs avantages : les partitions obtenues sont généralement de meilleure qualité que les partitions que l’on peut extraire d’une hiérarchie pour un nombre de groupes donné (STÉPHANE, 2012). Par ailleurs, les méthodes non-hiérarchiques sont généralement bien plus rapides que les méthodes hiérarchiques (ROCCHIO, 1966) (XU, 2005). Néanmoins, une limite bien connue de ces méthodes est que leur partition finale dépend souvent de la partition initiale (SALTON, et al., 1978).

2.4.2.2.2 Modèle à base de réseau bayésien

Du point de vue formel, on s’appuie sur un classifieur bayésien, où la probabilité des évaluations est conditionnellement indépendante sachant l’appartenance à une variable de classe C non observée comportant un ensemble de valeurs discrètes relativement petit. Le modèle de probabilité qui met en relation les probabilités jointes des classes et des évaluations, et un ensemble de distributions conditionnelles et marginales, est la formulation standard « naïve » de Bayes :

$$Pr(C = c, v_1, \dots, v_n) = Pr(C = c) \prod_{i=1}^n Pr(v_i | C = c) \quad (F2. 2)$$

La partie gauche est la probabilité d’observer un individu d’une classe particulière et un ensemble complet de valeurs d’évaluations. Dans ce contexte, on peut calculer directement les expressions de probabilité requises pour l’équation précédente donnant la valeur prédite.

Les paramètres du modèle, les probabilités d’appartenance à une classe $Pr(C = c)$, et les probabilités conditionnelles des évaluations sachant la classe sont estimées à partir d’un ensemble d’exemples d’évaluations d’utilisateurs, appelé la base des évaluations.

Le modèle à base de réseau bayésien associe un nœud à chaque item ou objet du domaine. Les états pour chaque nœud correspondent aux valeurs d’évaluation possibles pour chaque item. On inclut également un état correspondant à l’absence d’évaluation pour les domaines où il n’y a pas d’interprétation naturelle pour les données manquantes.

On peut alors appliquer un algorithme d’apprentissage de réseau bayésien sur la base d’exemples, où les évaluations manquantes sont associées à une valeur « pas d’évaluation ». L’algorithme d’apprentissage cherche sur plusieurs structures de modèle en termes de dépendances pour chaque item. Dans le réseau résultant de l’apprentissage, chaque item a un ensemble d’items « parent » qui sont les meilleurs prédicteurs de ses évaluations. Chaque table

de probabilité conditionnelle est représentée par un arbre de décision qui code les probabilités conditionnelles pour ce nœud. Pour plus d'information sur l'approche de Baye le lecteur peut consulter (JANNACH, et al., 2010) (GHAZANFAR, et al., 2014) (CHIEN, et al., 1999)

2.4.2.3 Avantages et inconvénients du filtrage collaboratif

Les méthodes de filtrage collaboratif présentent plusieurs avantages dont les plus importants sont :

- Pas besoin d'un modèle détaillé de préférences ; nous avons juste besoin d'un vecteur de valorisations d'objet et permet de recommander des contenus difficiles à analyser.
- Effet de surprise (serendipity) : l'effet de surprise que peut recevoir l'utilisateur en recevant une recommandation pertinente qu'il n'aurait pas trouvée seul est souvent souhaitable. Les algorithmes basés sur le filtrage collaboratif permettent généralement de faire des recommandations à effet de surprise. Par exemple, si un utilisateur u est proche d'un utilisateur v du fait qu'il ne regarde que des comédies, et si v apprécie un film d'un autre genre, ce film peut être recommandé à u du fait de sa proximité avec v .
- Non nécessité de la connaissance du domaine : les systèmes de recommandation basés sur le filtrage collaboratif ne requièrent aucune connaissance sur les objets. Ces méthodes peuvent recommander des items sans avoir besoin de comprendre leurs sens ni disposer de leurs attributs. La recommandation est basée uniquement sur les notes données aux items.

Cependant, l'utilisation des techniques de filtrage collaboratif peut entraîner plusieurs problèmes (BENOURET, 2017) :

- Le démarrage à froid : concerne à la fois les nouveaux utilisateurs et les nouveaux items qui sont introduits dans le système. Un nouvel utilisateur qui n'a noté aucun item ne peut pas recevoir de recommandation puisque le système ne connaît pas ses goûts. Ce problème est connu sous le nom de problème du démarrage à froid pour les utilisateurs (user cold start). Une solution à ce problème est de lui demander explicitement de noter un certain nombre d'items. D'autres solutions consistent à recommander au départ les items les plus populaires ou même des recommandations aléatoires. Ce problème du

démarrage à froid se pose aussi lors de l'ajout d'un nouvel item. Celui-ci ne peut pas être recommandé avant d'avoir été noté par un certain nombre d'utilisateurs.

- La parcimonie (sparsity) : Le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous-ensemble des items disponibles. De ce fait, la matrice des notes est une matrice creuse avec un taux de valeurs manquantes pouvant atteindre 95% du total des valeurs (PAPAGELIS, et al., 2005). Les systèmes de filtrage collaboratif ont des difficultés dans ce cas, le nombre de notes à prédire étant largement supérieur aux nombres de notes déjà connues. Le problème de la parcimonie peut être réduit en utilisant les approches par modèles qui réduisent la dimension de la matrice des notes.
- Le problème du mouton gris (gray sheep) : Les utilisateurs qui ont des goûts étranges (qui varient de la norme ou qui sortent du commun) n'auront pas beaucoup d'utilisateurs voisins. Il sera donc difficile de faire des recommandations pertinentes pour ce genre d'utilisateurs (GHAZANFAR, et al., 2014)
- Plus il y a un grand nombre d'éléments et d'utilisateurs, le calcul croît linéairement ; des algorithmes appropriés sont donc nécessaires ;

2.4.3 Recommandation hybride

Les systèmes hybrides exploitent les caractéristiques des systèmes basés sur le contenu et des systèmes collaboratifs, en raison de la nature complémentaire des deux en utilisant ainsi à la fois des connaissances extérieures et les caractéristiques des éléments. Ce qui est recherché est de surmonter les inconvénients des deux systèmes pour obtenir de meilleures recommandations.

Pour créer un système hybride collaboratif basé sur le contenu, les profils utilisateur sont gérés en fonction de l'analyse du contenu des éléments, et ces profils sont comparés directement pour déterminer la similarité entre les utilisateurs pour une recommandation collaborative.

Il est à noter que le terme « hybride » est un artefact de l'évolution historique des systèmes de recommandation où certaines sources de connaissances ont été exploitées en premier lieu, conduisant à des techniques bien établies qui ont ensuite été combinées. L'objectif est alors de s'appuyer sur des sources de connaissances multiples, en choisissant les plus appropriées à une tâche donnée afin de les utiliser le plus efficacement possible. (Voir figure suivante).

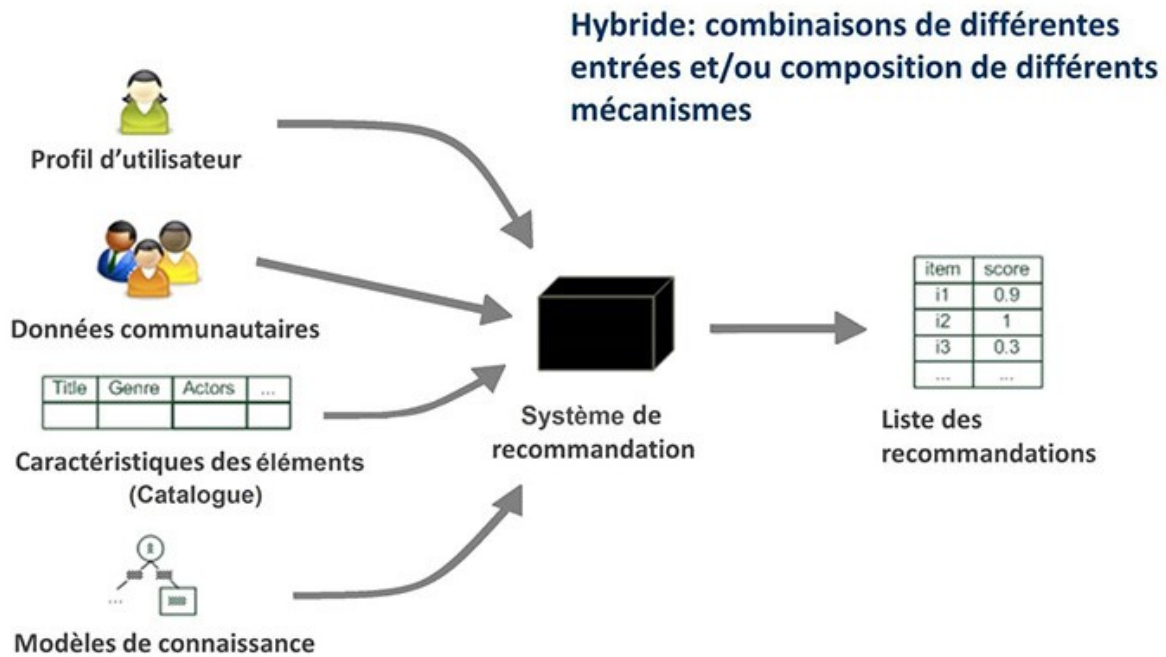


Figure 2-3: Filtrage hybride (JANNACH, et al., 2010)

Il existe trois grandes catégories de combinaisons de systèmes de recommandation pour concevoir un système de recommandation hybride (BURKE, 2002) : la combinaison monolithique (monolithic hybridization design), la combinaison parallèle (parallelized hybridization design) et la combinaison tubulaire (pipelined hybridization design).

« **Monolithique** » décrit une conception d'hybridation qui intègre les aspects de différentes stratégies de recommandation en un seul algorithme. Comme illustré sur la **Figure 2-4**, différents systèmes de recommandation y contribuent puisque l'approche hybride utilise des données d'entrée additionnelles qui sont spécifiques à un autre algorithme de recommandation, ou bien les données d'entrée sont complétées par une technique et exploitées par une autre. Par exemple, un système de recommandation basé sur le contenu qui exploite également des données communautaires pour déterminer des similarités entre éléments relève de cette catégorie.

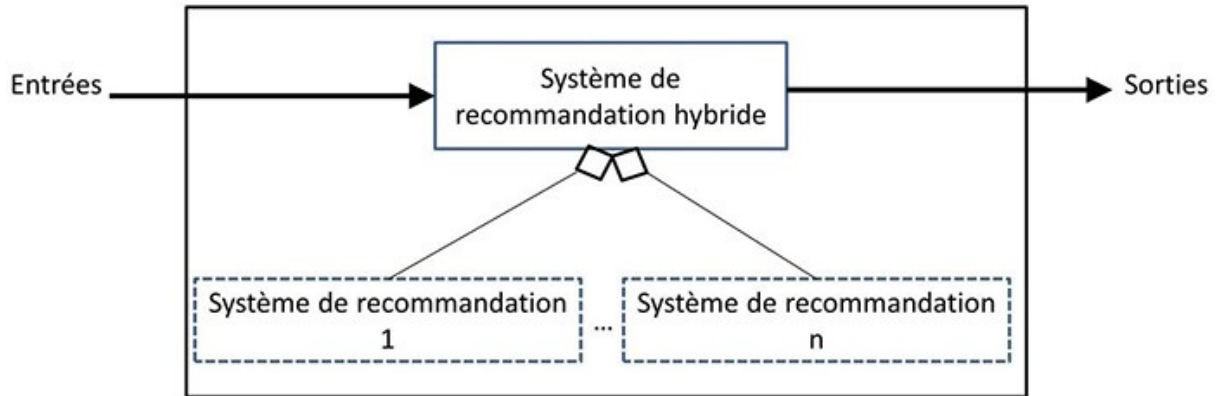


Figure 2-4: Conception d'hybridation monolithique (JANNACH, et al., 2010)

Les deux autres approches hybrides nécessitent au moins deux mises en œuvre de recommandations séparées qui sont combinées en conséquence. Sur la base de leurs données d'entrée, les systèmes hybrides de recommandation parallèles fonctionnent indépendamment l'un de l'autre et produisent des listes de recommandations distinctes, comme illustré sur la **Figure 2-5**. Dans une étape ultérieure d'hybridation, leurs sorties sont combinées en un ensemble final de recommandations.

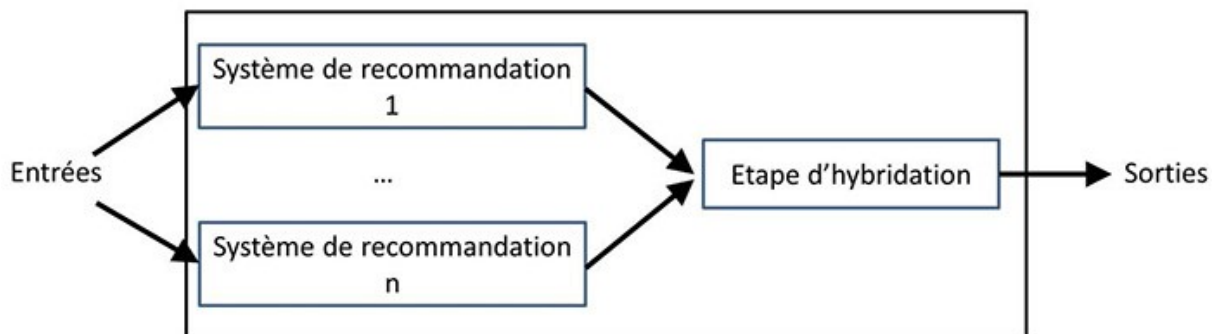


Figure 2-5: Conception d'hybridation parallèle (JANNACH, et al., 2010)

Lorsque plusieurs systèmes de recommandation sont joints dans une architecture tubulaire, comme illustré par la **Figure 2-6**, la sortie de l'un des systèmes de recommandation devient une partie des données d'entrée du système suivant.

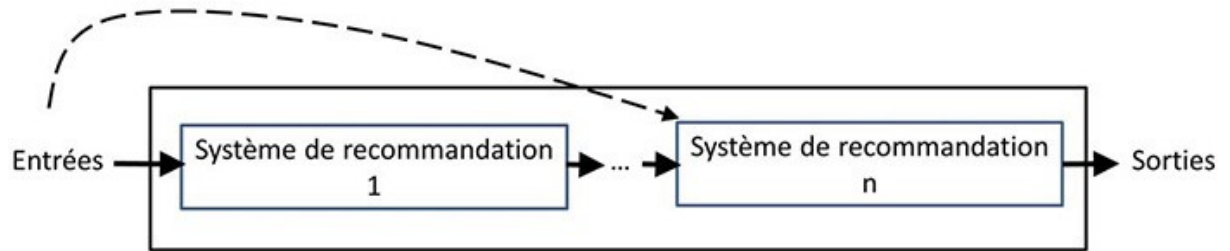


Figure 2-6: Conception d'hybridation tubulaire (JANNACH, et al., 2010).

2.4.4 Exemple de systèmes de recommandation

Il existe actuellement un large éventail de systèmes de recommandation utilisés dans différents domaines, à des fins commerciales, scientifiques ou expérimentales. Parmi lesquels nous pouvons citer :

Google Actualités : C'est un système de recommandation d'actualités personnalisées développé par Google. Selon (DAS, et al., 2007) ce service génère des recommandations avec la méthode hybride combinant trois techniques :

- Le **filtrage collaboratif** avec le clustering *MinHash*, qui est une méthode probabiliste qui attribue une paire d'utilisateurs à un même groupe avec une probabilité proportionnelle à la coïncidence entre l'ensemble des éléments pour lesquels ces utilisateurs ont voté.
- L'Indexation de probabilité sémantique latente (*Probabilistic Latent Semantic Indexing PLSI*). C'est une indexation automatique de document qui se base sur un modèle de classes sémantiques latentes, avec cette technique l'indexation se fait avec l'analyse des mots qui ont des relations sémantiques.
- La Co-visitation, une méthode basée sur l'algorithme Item qui calcule quelles actualités les utilisateurs ont tendance à voir s'ils ont vu un article particulier.

Le service fonctionne pour les utilisateurs enregistrés qui ont explicitement sauvegardé leurs historiques de navigation. Le système crée des profils pour les actualités présentant un intérêt pour l'utilisateur en fonction du nombre de clics qu'il a fait dans ses sessions passées. Toutes les informations relatives à l'historique des clics d'un utilisateur sont sécurisées conformément à la politique de confidentialité de Google.

Movilens : est un système de recommandation de films basé sur le filtrage collaboratif pour générer la recommandation des films. Comme indiqué sur la **Figures 2-7** dans ce système les

utilisateurs notent les films à travers les feedbacks explicites en disant s'ils ont aimé ou non en associant une note. Cette information est utilisée pour générer des recommandations personnalisées d'autres films que l'utilisateur pourrait aimer.

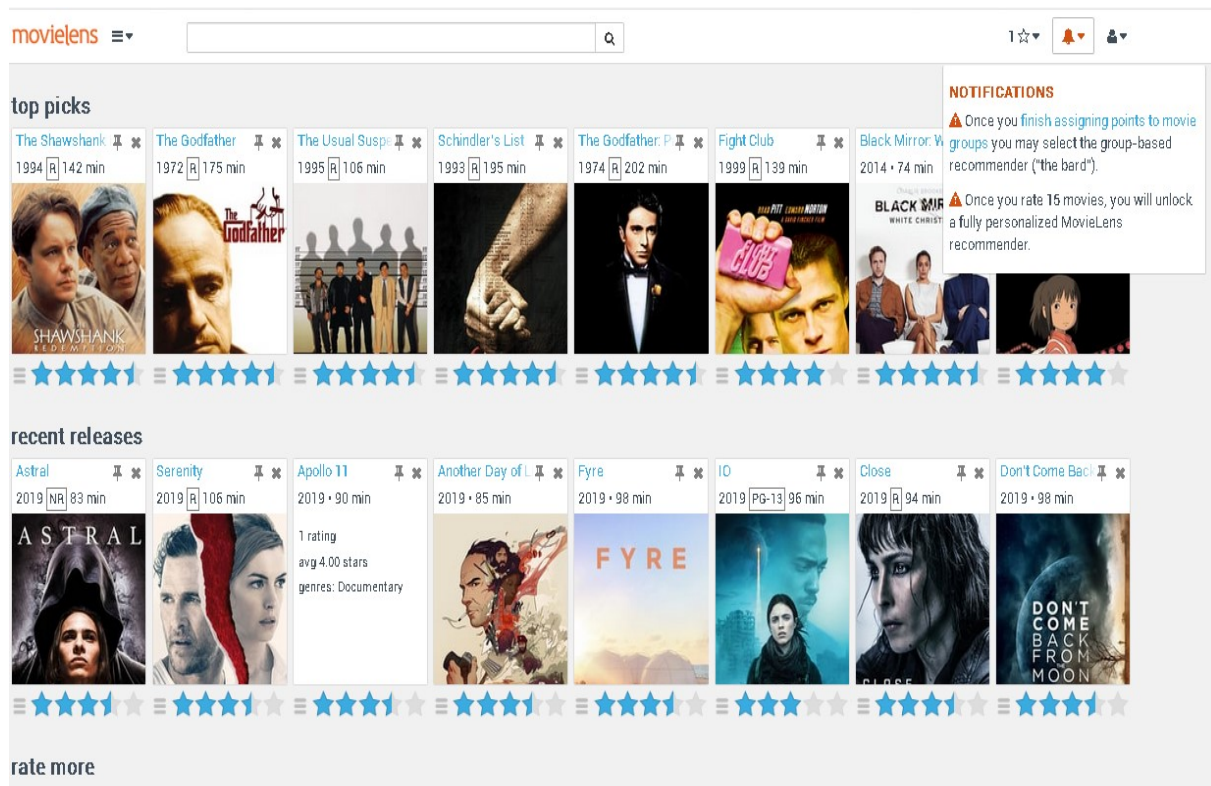


Figure 2-7: Exemple de MovieLens

Chaque membre du système est en voisinage avec d'autres utilisateurs qui ont des goûts similaires. Les évaluations de ces voisins sont utilisées pour créer des recommandations personnalisées pour un utilisateur donné.

Filmaffinity : est un système de recommandation de films (documentaires, court-métrage, séries télévisées, etc.) qui utilise le filtrage collaboratif avec plusieurs milliers d'utilisateurs. Comme montre la **Figure 2-8**, une fois enregistré, pour recevoir des recommandations, un utilisateur doit évaluer certains contenus en utilisant un barème de notes comprises entre 1 et 10. Le système crée des "âmes sœurs" qui sont les utilisateurs dont les notes sont proches. Le système dispose des filtres pour sélectionner les films par genres, année de sortie, pays, etc.

PERNY, Patrice et ZUCKER, Jean-Daniel. Preference-based search and machine learning for collaborative filtering: the Film-Conseil movie recommender system. Information, Interaction, Intelligence, 2001, vol. 1, no 1, p. 9-48. (PERNY, 2001)

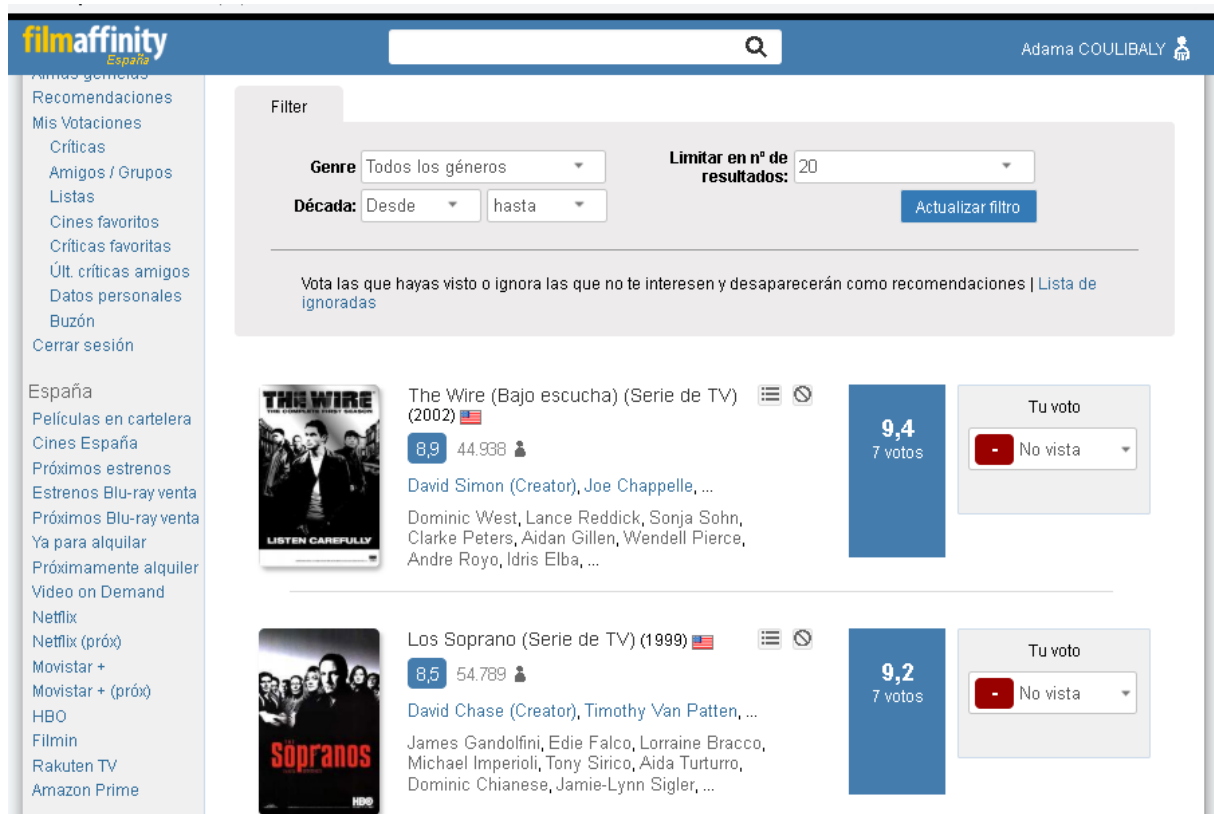


Figure 2-8: Système de recommandation du site web FilmAffinity

2.4.5 Autres classifications de systèmes de recommandation

Dans la littérature (BURKE, 2002), ils existent d'autres types de systèmes de recommandation, bien que ceux-ci soient liés d'une manière ou d'une autre, aux types de systèmes de recommandation mentionnés ci-dessus :

- **Les approches démographiques :**

Dans une approche démographique, le système de recommandation demande à l'utilisateur de renseigner un certain nombre de paramètres (âge, sexe, ville, emploi, intérêt, etc.). Ces paramètres permettent de définir des classes démographiques dans lesquelles les utilisateurs sont catégorisés. Par la suite, les recommandations de chaque utilisateur proviennent de la classe qui lui a été attribuée.

- **Les approches basées sur la connaissance et les approches basées sur l'utilité :**

Les approches basées sur la connaissance s'appuient sur un ensemble de règles définissant le domaine. Le mode opératoire dans ce genre de système est conversationnel et l'utilisateur est amené à définir à chaque fois un certain nombre de critères pouvant mieux répondre à ses attentes. Il ne s'agit donc pas de construire un

profil utilisateur à long-terme mais de demander systématiquement à l'utilisateur de déterminer ses besoins explicitement. Le système cherche donc les items remplissant au mieux les conditions définies par l'utilisateur pour les lui présenter. Ceci peut être effectué grâce à un ensemble de règles définies par l'expert qui filtrent les items selon les critères définis par l'utilisateur.

Les approches basées sur l'utilité peuvent être considérées comme un cas particulier des approches basées sur la connaissance. En effet, les approches basées sur l'utilité se basent également sur un ensemble de règles et, en plus de définir ses critères, l'utilisateur est amené à définir sa propre fonction d'utilité selon certaines propriétés du domaine. Cette fonction est donc utilisée afin de générer des recommandations, tandis que dans une approche basée sur la connaissance, seules les règles définies par l'expert sont utilisées pour la recommandation (BURKE, 2002).

Par ailleurs on trouve des systèmes de recommandation sémantiques qui utilisent le web sémantique pour améliorer et enrichir la représentation de l'information. Ses systèmes peuvent se classer en trois types (PEIS, et al., 2008) :

- **Systèmes basés sur l'ontologie** : L'ontologie (schéma conceptuel de l'information d'un domaine donné) est utilisée pour représenter les informations dans la modélisation des profils d'utilisateurs.
- **Systèmes adaptables au contexte** : Ces systèmes prennent en compte différents facteurs tels que temps, l'emplacement, niveau d'expérience, l'appareil qui est utilisé au moment de la réception de la recommandation, etc. pour déduire le contexte dans lequel l'utilisateur se trouve et adapter les recommandations à ces circonstances.
- **Systèmes basés sur les réseaux des confiances** : Ces systèmes ajoutent des filtres d'information additionnels aux systèmes adaptables au contexte. Ces systèmes visent à garantir la fiabilité et la précision des recommandations en créant des réseaux de confiance entre les différentes parties composant le système. La confiance des utilisateurs, en plus de donner de la crédibilité aux résultats des recommandations, implique une augmentation de la qualité et de la convivialité du système.

2.5 Conclusion

Ce chapitre a présenté les systèmes de recommandations qui sont devenus omniprésents ces dernières années dans de nombreux domaines.

Ces systèmes sont conçus pour aider les utilisateurs à trouver des ressources qui les intéressent

et qui sont adaptées à leurs préférences, parmi le nombre important de choix qui s'offrent à eux. Ils peuvent donc être considérés comme des systèmes d'aide à la décision.

Les systèmes de recommandation sont adaptables à plusieurs domaines de la science particulièrement dans la prise de décision. Dans ce domaine nous pouvons citer les travaux de (MARTIN, 2012) (FOMBA, 2018)). Le premier a mis au point un système capable d'apporter des aides à l'activité de l'utilisateur en fonction de son profil. Tandis que le second a proposé un outil d'aide dans le choix d'un opérateur d'agrégation dans des contextes multicritères.

Pour plus de détail sur les systèmes de recommandation, nous invitons le lecteur à lire (PERNY, 2001).

Plusieurs définitions des systèmes de recommandation ont été proposées ainsi que leurs classifications en plusieurs approches dont les principales sont : la recommandation Objet ou Contenu, la Recommandation sociale ou Collaborative et la Recommandation hybride.

Les deux méthodes de recommandation objet et collaborative présentent des caractéristiques complémentaires. Les nombreux travaux s'intéressant à la recommandation hybride combinent plusieurs approches dans le but de bénéficier de leurs avantages, tout en réduisant les inconvénients de chaque méthode lorsqu'elle est utilisée séparément.

Nous avons vu aussi d'autres approches de classement qui sont les approches démographiques, les approches basées sur la connaissance et les approches basées sur l'utilité. Pour améliorer les recommandations d'autres approches proposent d'utiliser le web sémantique.

CHAPITRE 3 :

**3. Caractérisations des procédures
de vote : paramètres et critères**

3.1 Introduction

Il existe plusieurs opinions des chercheurs à l'égard des procédures de vote. Cela peut rendre difficile la comparaison objective des procédures de vote. On peut utiliser plusieurs approches pour comparer des procédures de vote. Une approche consiste à définir mathématiquement et logiquement des exigences théoriques ou des propriétés souhaitables auxquels sont soumis les procédures de votes. Cela donne des résultats parfaitement objectifs, mais leur pertinence pratique reste discutable.

Après avoir décrit certaines procédures de vote dans le **chapitre I**, il est convenable de les caractériser afin d'avoir des jugements objectifs pour choisir la procédure dans un contexte donné. Lors du choix d'une procédure de vote, il est important de déterminer les avantages et les inconvénients de cette dernière. C'est la raison pour laquelle ont été développés des exigences ou propriétés inspirés des théorèmes et des paradoxes de la théorie du choix social, permettant d'évaluer et de comparer les procédures de vote.

Dans ce chapitre, nous allons définir les principaux termes qui sont utilisés pour caractériser les procédures de vote. Nous allons dégager la liste des paramètres pertinents qui ont des influences sur les résultats d'un vote. Ensuite nous énoncerons les différents théorèmes qui ont permis de déterminer les critères d'évaluations des différentes procédures de vote. Enfin, les critères de choix seront décrits et utilisés pour faire une comparaison des procédures. Avant de conclure nous proposerons une adaptation des critères aux cas d'une prise décision en groupe de taille réduite.

3.2 Définitions

Définition 3.1 (paradoxe) [encarta 2009] : un paradoxe se définit comme :

1. un fait ou une chose singulière et qui heurte le bon sens en contenant apparemment une contradiction ;
2. une opinion contraire à la logique ou à ce qui est communément admis ;
3. une affirmation à la fois vraie et fausse.

Dans notre champ d'étude, un paradoxe de vote se définit comme un résultat indésirable qu'une procédure de vote peut produire et qui peut être considéré à première vue, du moins par certaines personnes, comme surprenant ou contre-intuitif. Donc le paradoxe, désigne des résultats étonnants lors d'un vote. Les paradoxes sont décrits et analysés par (FISHBURN, 1974) (HUDRY, 2003) (DURAND, 2000) (FELSENTHAL, 2010).

3.3 Paramètres pertinents d'un vote

(FELSENTHAL, et al., 2018) proposent les paramètres suivants comme pertinents :

- Forme des bulletins,
- Nature du vote,
- Nombre de votants,
- Nombre de candidats,
- Nombre de candidats à élire,
- L'ordre de préférence de chaque électeur parmi les candidats en lice,
- La quantité d'informations que les électeurs ont sur les ordres de préférence des autres électeurs,
- L'ordre dans lequel les électeurs votent s'ils ne sont pas simultanés, les premiers votants peuvent influencer les suivants
- L'ordre dans lequel les candidats sont votés si les candidats ne sont pas votés simultanément, et
- La gestion d'égalités entre les candidats (ex aequo).

3.3.1 Les formes des bulletins

Selon le type de procédure de vote choisi, la forme des bulletins de vote peut varier entre un seul élément, un sous-ensemble d'éléments ou une liste ordonnée. Cette variation peut amener des résultats différents dans les calculs de voix obtenues par les candidats.

3.3.2 Nature du vote

Dans la pratique on peut décider de voter à bulletin ouvert ou secret selon les situations. Cela peut avoir un impact considérable sur les résultats car le choix d'un votant peut influencer l'autre. Souvent les votes à bulletin ouvert peuvent encourager les votes stratégiques ou non-sincères.

3.3.3 L'ordre de préférence de chaque électeur parmi les candidats en lice

Dans un vote préférentiel, ce sont les ordres exprimés par les électeurs qui déterminent le résultat final car les calculs des voix se font en se basant sur ces ordres. Donc c'est l'élément incontournable.

3.3.4 La quantité d'informations que les électeurs ont sur les ordres de préférence des autres électeurs

C'est un élément difficile à mesurer et est fortement lié à la nature du vote, il peut favoriser la manipulation de la procédure.

3.3.5 L'ordre dans lequel les électeurs votent s'ils ne sont pas simultanés

C'est aussi un paramètre difficile à mesurer et peut aussi favoriser la manipulation de la procédure de vote en cours.

3.3.6 L'ordre dans lequel les candidats sont votés si les candidats ne sont pas votés simultanément

C'est un paramètre qui peut facilement influencer le résultat final surtout avec les procédures de votes de type séquentiel et élimination ou des compétitions en duel.

Considérons la situation suivante avec trois votants et quatre candidats (A, B, C et D) avec les préférences montées dans le tableau suivant.

Nb candidata →		1	1	1
Rang	1 ^{ier}	C	B	A
	2 ^{ème}	B	A	D
	3 ^{ème}	A	D	C
	4 ^{ème}	D	C	B

Tableau 3-1: Exemple de l'ordre des candidats

Supposons que les confrontations majoritaires soient organisées en suivant l'ordre alphabétique : A contre B, le vainqueur contre C, etc. Les résultats des duels majoritaires sont alors : B bat A, C bat B et D bat C. Le candidat D est donc l'unique vainqueur alors qu'il est facile de vérifier que tous les votants lui préfèrent le candidat A.

3.3.7 Nombre de candidats

On peut dire que c'est l'élément clé de la théorie du choix social. Les théorèmes de Condorcet, d'impossibilité d'Arrow, de May et de Sen qui ont fait évoluer la théorie du vote, ont été énoncés sur une base de nombre de candidats supérieur ou égal trois.

Si le choix doit se faire entre deux candidats seulement, toutes les procédures de vote étudiées convergent vers un même résultat. Par contre au-delà de deux candidats tous les paradoxes rentrent en jeu et peuvent conduire à des situations difficiles à élucider.

Des travaux (HUDRY, 2003) (DURAND, 2015) ont démontré l'influence de la parité du nombre de candidats sur les résultats des votes. Selon HUDRY, la probabilité d'apparition du paradoxe de Condorcet pour trois candidats varie entre 5.56% (pour trois votants) et 8.77% (pour un nombre de votants qui tend vers l'infini). Elle croît avec le nombre de candidats, et, à nombre de candidats fixé, elle croît quand le nombre de votants augmente. Par exemple, d'après (GEHRLEIN, 1997), elle vaut environ 52,5 % pour 25 candidats et 3 votants, pour atteindre environ 73 % pour 25 candidats et une infinité de votants.

De plus, un nombre élevé de candidats peut, entrainer des difficultés dans l'exécution d'un vote, amenant une dispersion de voix dans les comptages et aussi encourager la manipulation stratégique.

3.3.8 Nombre de candidats à élire

Presque toutes les procédures de votes étudiées dans ce manuscrit, permettent d'élire un ou plusieurs candidats. Si le nombre de candidats à élire est considérable alors le problème de départage des égalités peut être évité. Vu de cet angle, ce paramètre est important dans le résultat d'un vote.

Si par exemple nous avons A, B, C et D, 4 candidats et 6 votants qui ont exprimé les préférences suivantes (voir le *tableau 3-2.*)

Nb candidata→		3	2	1
Rang	1 ^{ier}	C	B	B
	2 ^{ème}	B	C	C
	3 ^{ème}	A	D	A
	4 ^{ème}	D	A	D

Tableau 3-2: Exemple du Nombre de candidat à élire

En appliquant par exemple la méthode de pluralité (ou Condorcet ou Borda), pour déterminer le vainqueur de ce vote nous obtenons des scores comme $S(B)=S(C)>S(A)>S(D)$.

Si on doit désigner un seul vainqueur, le résultat de ce vote va poser des conflits entre les candidats B et C. Par contre, si on a besoins de deux candidats pour entrer dans un conseil

d'administration, B et C seront déclarés vainqueurs, d'où l'influence du nombre de candidats à élire.

3.3.9 Gestion des égalités entre candidats

Une procédure de vote est dite résolue si, pour chaque profil, elle renvoie un candidat unique appelé vainqueur. Tandis qu'elle est dite irrésolue ou non-résolue si, pour certains profils, elle renvoie un ensemble de candidats ex-aequo qu'elle n'arrive pas à départager, appelé "Co-vainqueurs".

Il y a deux manières de fixer les règles de départage des égalités soit par :

- Le déterminisme (par exemple selon une priorité prédéfinie entre les candidats, comme la priorité au plus âgé, ou entre les votants, comme la priorité à la voix du chef),
- Un tirage au sort.

L'unique vainqueur sera choisi parmi les Co-vainqueurs au moyen de la règle de départage ainsi fixée.

C'est un paramètre influant dans un vote et il doit être fixé en avance afin que les résultats obtenus soient facilement acceptés.

3.4 Les théorèmes importants de la théorie du vote

Un des théorèmes importants de la théorie du vote, est celui de Condorcet qui a été énoncé sous forme de paradoxe dans le chapitre I (paradoxe de Condorcet). Selon ce paradoxe, il n'existe pas de processus de choix social indiscutable, qui permette d'exprimer une hiérarchie des préférences cohérente pour une collectivité à partir de l'agrégation des préférences individuelles exprimées par chacun des membres de cette même collectivité..

Un autre théorème important est celui d'Arrow qui sera utilisé dans le reste de ce manuscrit pour caractériser et/ou comparer les différentes procédures de vote étudiées.

Théorème 3.1 : Théorème d'impossibilité d'Arrow

Pour au moins trois options de choix et deux votants, il n'existe pas de fonction de choix social satisfaisant les propriétés suivantes :

- **Universalité**, ou domaine non restreint : la fonction de choix social doit être définie dans tous les cas de figure, c'est-à-dire déterminer toujours un choix collectif, quel que soit le profil de préférences de chaque individu.

- **Non-dictature** : il n'existe aucun individu pour lequel la liste de ses choix personnels coïncide avec la fonction de choix social, indépendamment des préférences des autres ;
- **Unanimité** (optimum de Pareto) : lorsque tous les individus préfèrent une certaine option à une certaine autre, la fonction de choix social doit associer cette même préférence à la société.
- **Indépendance des options non pertinentes** : le classement relatif de deux options ne doit dépendre que de leur position relative pour les individus et non du classement d'options tierces ; si l'on ne considère qu'un sous-ensemble d'options, la fonction ne doit pas aboutir à un autre classement de ce sous-ensemble.

Dans une autre version du théorème, l'unanimité peut être remplacée par les deux hypothèses suivantes :

- **Monotonie** : un individu ne doit pas pouvoir faire diminuer le classement global d'une option en la classant plus haut.
- **Souveraineté** : aucun choix social ne doit être impossible a priori.

Illustration du théorème de d'impossibilité d'Arrow

Soit A, B, C et D quatre candidats à une élection. Considérons trois votants dont les préférences sont les suivantes :

		Nombre de votants →	
		2	1
Rang	1 ^{ier}	C	A
	2 ^{ème}	A	B
	3 ^{ème}	B	D
	4 ^{ème}	D	C

Tableau 3-3: Listes des Préférences des votants

En appliquant la méthode de Borda, nous obtenons les scores suivants :

$$S(A)_{Borda} = 10 ; S(B)_{Borda} = 7 ; S(C)_{Borda} = 9 \text{ et } S(D)_{Borda} = 4.$$

Donc le classement final du groupe conduit à : A, C, B et D.

Considérons le scrutin de même type avec les préférences du **Tableau 3-4**.

		Nombre de votants→	
		2	1
Rang	1 ^{er}	C	A
	2 ^{ème}	A	C
	3 ^{ème}	B	B
	4 ^{ème}	D	D

Tableau 3-4: Listes de préférences avec une légère modification

Les nouveaux scores avec la méthode Borda sont :

$$S(A)_{Borda} = 10 ; S(B)_{Borda} = 6 ; S(C)_{Borda} = 11 \text{ et } S(D)_{Borda} = 3.$$

Par conséquent le classement final du groupe devient C, A, B et D ce qui contredit le principe de l'Indépendance.

Notons que dans chacune des listes la position respective de A et de C est restée identique d'un scrutin à l'autre : A est préféré à C par un votant tandis que deux votants ont la préférence inverse.

La condition d'indépendance voudrait alors que la position relative de A et de C soit identique à l'issue des deux scrutins ce qui n'est pas le cas avec la méthode de Borda. Cette dernière utilise le fait que l'écart entre A et C semble plus grand dans la liste A>B>D>C que dans la liste A>C>B>D, puisque B et D viennent s'intercaler entre A et C dans le premier cas. Cette dépendance de la position respective de A et de C par rapport à B et à D est exclue par la condition d'indépendance.

Elle exclut de même toute méthode qui utiliserait en plus des listes une information qualifiant la plus ou moins grande intensité de chaque préférence à l'intérieur des listes.

Théorème 3.2 : Théorème d'impossibilité de Gibbard-Sattthertwaite

Toute méthode de vote non dictatoriale est manipulable (i.e. il est préférable pour au moins un des individus de mentir sur ses préférences).

Théorème 3.3: A. K. Sen [Sen, 1978]¹.

On ne peut satisfaire à la fois les propriétés d'unanimité et d'universalité et la propriété de liberté individuelle.

¹ <https://www.cairn.info/revue-cahiers-d-economie-politique-2004-2-page-53.htm>

Ces théorèmes et les paradoxes de Condorcet ont servi de repère pour constituer les différents attributs ou caractéristiques de base pour évaluer ou caractériser une procédure de vote. Les travaux (GEANAKOPOLOS, 2005), (LENGAIGNE, et al., 2004) et (SALLES, 2006) contiennent des exemples illustrant les théorèmes énoncés plus haut.

3.5 Les caractéristiques / attributs des procédures de vote

D'après plusieurs auteurs (FELSENTHAL, et al., 2018), ces critères se classent en deux groupes : les simples et les conditionnels. De plus il peut exister des critères additionnels appelés administratif-techniques.

3.5.1 Les caractéristiques simples

Ce sont des critères qui stipulent que, si à partir de l'ensemble des bulletins de vote on peut déduire un classement, un certain candidat doit ou ne doit pas gagner. Parmi cette catégorie on peut citer :

3.5.1.1 Caractéristique du vainqueur de Condorcet (Condorcet, 1785 ; Black, 1958)

Dans des bulletins avec classement des candidats, s'il existe un candidat x qui, confronté à tout autre candidat, est toujours le gagnant alors le candidat x doit être élu.

3.5.1.2 Caractéristique du Perdant de Condorcet (Borda, 1784 ; Black, 1958)

Si un candidat x comparé en duel à d'autres candidats, perd, alors il ne doit pas être élu.

3.5.1.3 Caractéristique de la majorité absolue

Si un candidat x est placé en premier dans plus de la moitié des bulletins des votants, il doit être élu.

3.5.1.4 Caractéristique du Perdant de la majorité

Si un candidat x est placé derrière un candidat Y dans la majorité des bulletins, alors il ne doit pas être élu.

3.5.1.5 Caractéristique de Pareto (Fishburn, 1974)

Si le candidat x est placé, dans tous les bulletins, derrière le candidat y , alors x ne peut pas gagner devant y .

3.5.2 Caractéristiques conditionnelles

Ce sont des caractéristiques qui stipulent que, si un certain candidat gagne dans une circonstance particulière, le même candidat doit (ou ne doit pas) gagner dans une autre circonstance. Les parties suivantes expliquent les principaux critères conditionnels.

3.5.2.1 Caractéristique de monotonie (Smith, 1973)

Si un candidat n'est pas gagnant et si on le rétrograde dans certains bulletins sans modifier l'ordre relatif des autres candidats alors il ne doit pas pouvoir gagner.

3.5.2.2 Caractéristique de cohérence (Young, 1974)

Si les bulletins sont partagés en deux groupes et si un candidat est le gagnant dans chaque groupe, il doit être le gagnant des élections.

3.5.2.3 Caractéristique de troncature (Brams, 1982 ; Fishburn et Brams, 1983)

Si un votant peut obtenir un meilleur résultat en classant dans sa liste seulement une partie de sa préférence sincère sur les candidats concurrents que s'il alignait dans sa liste sa préférence sincère complète. (FISHBURN, et al., 1984)

3.5.2.4 Caractéristique d'abstention (Fishburn et Brams, 1983 ; Ray, 1986 ; Moulin, 1988b ; Holzman, 1988/9 ; Perez, 1995)

Ce critère stipule que l'ajout d'un bulletin de vote, où le candidat A est strictement préféré au candidat B, à un décompte des voix existant ne devrait pas changer le gagnant du candidat A au candidat B. Les procédures de vote qui ne satisfont pas ce critère, permettent le vote tactique ou stratégique (s'abstenir d'une élection peut aider l'électeur à faire son choix préféré)

3.5.2.5 Caractéristique des jumeaux (Moulin, 1988b)

Deux électeurs ayant la même préférence peuvent obtenir un meilleur résultat si l'un d'eux décide de ne pas participer à l'élection tandis que l'autre vote sincèrement (C'est un cas spécial du critère d'abstention).

3.5.2.6 Caractéristique d'indépendance (Fishburn (1974a, b, 1977))

L'introduction d'un candidat supplémentaire ne doit pas modifier l'ordre relatif existant entre les autres candidats dans chaque bulletin.

3.5.2.7 Caractéristique de symétrie par inversion (Farquharson, 1969 ; Plott, 1973)

Si un candidat est gagnant et si on range, dans chaque bulletin, les candidats dans l'ordre inverse, ce même candidat doit perdre.

3.5.2.8 Caractéristique d'invulnérabilité aux votes tactiques (Gibbard, 1973 ; Satterthwaite, 1975)

- **Caractéristique de non-compromission.** Un électeur ne doit pas pouvoir placer un candidat au-dessus d'un candidat qu'il préfère en réalité.
- **Caractéristique de non-enterrement.** Un électeur ne doit pas pouvoir enterrer un candidat qu'il aurait normalement placé en position médiane.
- **Caractéristique de non-surévaluation.** Un électeur ne doit pas pouvoir surévaluer un candidat de manière à réserver à son candidat préféré un duel facile à vaincre.

3.5.3 Autres Caractéristiques : administratifs et techniques

En plus des critères décrits ci-dessus, les procédures de vote sont jugées également à l'aide de critères qui ne sont pas mathématiquement précis mais qui sont tout de même importants. Parmi ces critères on peut citer :

3.5.3.1 La simplicité

Certaines procédures de vote rendent plus difficile la participation de l'électeur à une élection en l'obligeant à classer tous les candidats en lice, tandis que d'autres procédures facilitent la tâche de l'électeur en lui demandant de voter pour un seul candidat ou pour tout candidat qu'il approuve.

3.5.3.2 Facilité de compréhension de la façon dont le gagnant est sélectionné

Afin d'encourager la participation à une élection, une procédure de vote doit être transparente, c'est-à-dire que les électeurs doivent comprendre comment leurs votes (préférences) sont regroupés dans un choix social. Ainsi, une procédure de vote où le gagnant est le candidat qui a obtenu la pluralité des voix est plus facile à expliquer - et considérée comme plus transparente - qu'une procédure qui peut impliquer des calculs mathématiques considérables (par exemple, celle de Kemeny) afin de déterminer le gagnant.

3.5.3.3 Facilité d'exécution des élections

Les procédures de vote ne nécessitant qu'un seul tour de scrutin (ou dépouillement) sont plus faciles à exécuter que les procédures électorales qui peuvent exiger plus d'un tour de scrutin (ou dépouillement). De même, il est plus facile de mener des procédures de vote n'exigeant que le décompte du nombre de voix reçues par chaque candidat que des procédures exigeant la conduite des duels entre tous les m candidats, ou celles exigeant

l'examen d'un maximum de $m!$ ordres de préférence sociale possibles afin de déterminer le gagnant.

3.5.3.4 Le risque de fraude

Il va mieux minimiser la tentation de voter de façon malhonnête, bien que toutes les procédures de vote soient vulnérables à la manipulation c'est à dire au phénomène selon lequel certains votants peuvent bénéficier d'un vote malhonnête, certaines procédures de vote (par exemple la méthode de Borda) y sont beaucoup plus vulnérables que d'autres.

3.5.3.5 Discriminabilité

On devrait préférer une procédure de vote plus discriminatoire, c'est-à-dire qu'elle est plus susceptible de sélectionner (de façon déterministe) un gagnant unique que de produire un ensemble de candidats à égalité. Ainsi, par exemple, lorsque l'ordre des préférences sociales est cyclique, les procédures de Schwartz et de Copeland sont nettement moins discriminantes que les autres procédures Condorcet-cohérents étudiées dans le présent document.

Il peut exister des conflits entre certains de ces critères technico-administratifs. Par exemple, une procédure comme celle de Kemeny qui, d'une part, est plus difficile à mettre en pratique et à expliquer aux électeurs potentiels (et donc moins transparente), est, d'autre part, plus discriminatoire et moins vulnérable aux comportements insincères.

3.6 Caractéristiques des procédures de vote à l'échelle d'un groupe de travail à taille réduite

Il existe des relations entre certains caractéristiques des points 3.5.1 et 3.5.2. Certains sont des cas spéciaux d'autres. Par exemple la caractéristique de la majorité absolue est un cas particulier de celle du gagnant de Condorcet. De même la caractéristique du perdant de Condorcet est un cas particulier du perdant de la majorité et sera désigné « critère du perdant ». Le critère des jumeaux est une version particulière du critère d'abstention, les deux seront regroupées en « critère de participation ».

L'effet de certaines caractéristiques évoquées dans les points 3.5 n'est pas aussi remarquable pour les votes en groupe de taille réduite qu'en un vote à grande échelle. L'un des buts de ce travail étant de concevoir un outil d'aide pour un facilitateur dont l'environnement de travail est numérique. La caractéristique de la participation n'interviendra pas ou la participation des décideurs désignés pour voter afin de soutenir leur point de vue dans la prise

décision est pratiquement évident. Ces faits peuvent réduire l'impact de certaines caractéristiques et paramètres évoqués antérieurement.


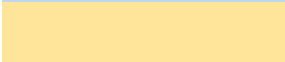
En tenant compte des considérations mentionnées ci-dessus nous retiendrons les le caractéristiques ou attributs suivantes pour caractériser les procédures de vote dans un contexte de prise de décision en groupe de taille réduite : C_1 : Caractéristique du gagnant de Condorcet, C_2 : Caractéristique de majorité absolue, C_3 : Caractéristique de Pareto, C_4 : Caractéristique du perdant, C_5 : : Caractéristique de participation, C_6 : Caractéristique de monotonie, C_7 : Caractéristique de cohérence.

Les études (SUITT, et al., 2014), (NURMI, 2012), (DURAND, 2000) (KONCZAK, et al., 2005) (FELSENTHAL, et al., 2018) ont permis de montrer l'impact des critères retenus sur les différentes procédures de vote. En considérant l'échelle de notation suivante, 0 : le critère n'affecte pas la procédure de vote, 1 : le critère affecte la procédure de vote et 2 : le critère a un fort impact sur la procédure nous proposons le tableau suivant inspiré des travaux (NURMI, 2010) (FELSENTHAL, 2010) (FELSENTHAL, et al., 2018).

Critères→	C_1	C_2	C_3	C_4	C_5	C_6	C_7	Nb	Nb	Nb
Procédure ↓								2	1	2+1
PLURALITE	1	2	0	2	0	0	0	2	1	3
APPROBATION	1	2	2	2	0	0	0	3	1	4
BORDA	1	2	0	0	0	0	0	1	1	2
COOMBS	1	0	0	0	1	2	1	1	3	4
HARE	1	0	0	0	1	2	1	1	3	4
BUCKLIN	1	2	0	2	1	0	1	2	3	5
JUGEMENT MAJ	1	2	0	2	1	0	1	2	3	5
CONDORCET	1	0	0	0	1	0	1	0	3	3
BLACK	0	0	0	0	1	0	1	0	2	2
NASON	0	0	0	0	1	2	1	1	2	3
KEMENY-YOUNG	0	0	0	2	1	0	1	1	2	3
COPELAND	0	0	0	0	1	0	1	0	2	2
MINIMAX	0	0	0	2	1	0	1	1	2	3
DODGSON	0	0	0	2	1	2	1	2	2	4

Tableau 3-5: Critères vs Procédures de vote

Le code couleur est

	Procédure sans classement
	Procédure par classement non Condorcet consistant
	Procédure par classement Condorcet consistant

3.7 Conclusion

Nous avons présenté les paramètres de base qui peuvent influencer les résultats d'un vote. Parmi ces paramètres certains sont quantifiables telles que le nombre de candidats, le nombre de votants, le nombre de candidats à élire et la nature du vote qui peut être secrète ou ouverte.

Dans un contexte de deux candidats seulement, les procédures de vote convergent à l'élection du candidat majoritaire et l'emploi de la méthode de pluralité est conseillée vu sa simplicité. Par contre, les principales controverses de la théorie du vote se multiplient quand le nombre de candidats dépasse deux. Il est possible qu'un vote conduise à la victoire de différents candidats selon la procédure de vote choisie pour déterminer le gagnant.

Des travaux de description des procédures de vote ont été initié par (PALHA, 2017) en utilisant certaines méthodes multicritères. Les études se sont portées sur l'analyses multicritère (ELECTRE III et PROMETHEE I) pour comparer certaines procédures de vote pour leur utilisation dans le contexte de prise de décision. Ils ont abouti en deux tendances : 1) l'élection d'un candidat majoritairement aimé, 2) candidat moyennement aimé.

Pour l'élection d'un candidat majoritairement aimé, les procédures Condorcet - cohérentes sont conseillées. Parmi ce type de procédures de vote, deux se démarquent par rapport aux autres : la méthode proposée par COPELAND et celle de KEMENY-YOUNG. Voir les travaux de (FELSENTHAL, et al., 2018) (PALHA, 2017) pour plus de détails. Si la transparence dans l'exécution est une caractéristique souhaitée, dans un contexte de prise de décision, alors c'est la méthode de COPELAND qui est adaptée. Et si c'est la manipulation qui doit être considérée comme déterminante, alors c'est la méthode de KEMENY-YOUNG qui est conseillée car elle résiste plus à ce critère.

Dans le cas où la participation est à encourager, les procédures de positionnement sont attrayantes particulièrement la méthode BORDA.

Les expériences menées par (BALINSKI, et al., 2012) montre aussi l'attractivité de la méthode du jugement majoritaire. Selon les résultats de ces expériences, le jugement majoritaire élit un candidat jugé médian, facilement accepté par une majorité des votants. Par contre la mise en place de cette méthode et le calcul associé à la détermination des voix rendent son implémentation un peu compliquée et peut jouer en sa défaveur.

La description des procédures de vote doit aussi prendre en compte d'autres attributs ou caractéristiques qui ne sont pas mathématiquement définissable c'est-à-dire de critère administratif-techniques.

Partie II : Contribution

Les contributions que nous avons réalisées sont décrites dans cette partie 2. Nous proposons dans le chapitre 4, les détails de la mise place de GRECO (Group vote RECOmmendation). Dans ce chapitre nous présentons l'architecture détaillé que nous avons adapté et les technologies utilisées. Nous expliquons en pseudo code, les algorithmes de recommandation hybride mise en place. En dernier point de ce chapitre, nous proposons des captures d'écrans à travers un exemple d'utilisation.

Enfin nous présentons dans le chapitre 5, les validations expérimentales et benchmark de notre travail.

CHAPITRE 4

4. GRECO

Group vote RECOmmendation

4.1. Introduction

GRECO (Group vote RECOmmendation) est un système de recommandation de procédures de vote pour un facilitateur. Le système peut automatiquement proposer au facilitateur, une procédure de vote adaptée au contexte de la prise de décision. Le facilitateur a aussi la possibilité de choisir une procédure de vote qu'il juge appropriée. Le but de GRECO est d'offrir un outil d'aide à la décision pour le facilitateur, dans lequel sont implémentées différentes méthodes de vote notamment : la pluralité, Borda, Condorcet, Black et Copeland. Pour choisir une procédure de vote, GRECO utilise une recommandation hybride basée sur la similarité entre les différents contextes de décision. Les procédures de vote sont regroupées en deux grandes catégories à savoir les procédures par classement et procédure sans classement. Les procédures par classement se divisent en deux sous-groupes, les Condorcet cohérents ou consistents et les Condorcet non cohérents non consistents.

Dans ce chapitre, nous présenterons l'architecture de GRECO, les technologies utilisées et l'algorithme détaillé des procédures de vote implémentées. Ensuite nous allons proposer la description interne de notre moteur de recommandation et quelques captures d'écran du système GRECO à l'aide d'un exemple concret de vote.

4.2. Architecture détaillée

Le diagramme suivant contient les entités nécessaires pour la mise en place de notre système. Nous allons décrire les différentes entités pour une meilleure compréhension de notre architecture.

GRECO est une application web basée sur une architecture 3 tiers : MVT (Modèle, Vue Template). Cette architecture est détaillée dans le point 4.3.1(Django).

Le facilitateur a la possibilité de créer les votes dans une réunion de prise de décision. Pour la création d'un vote, les différentes entités utilisées sont : les candidats, les votants.

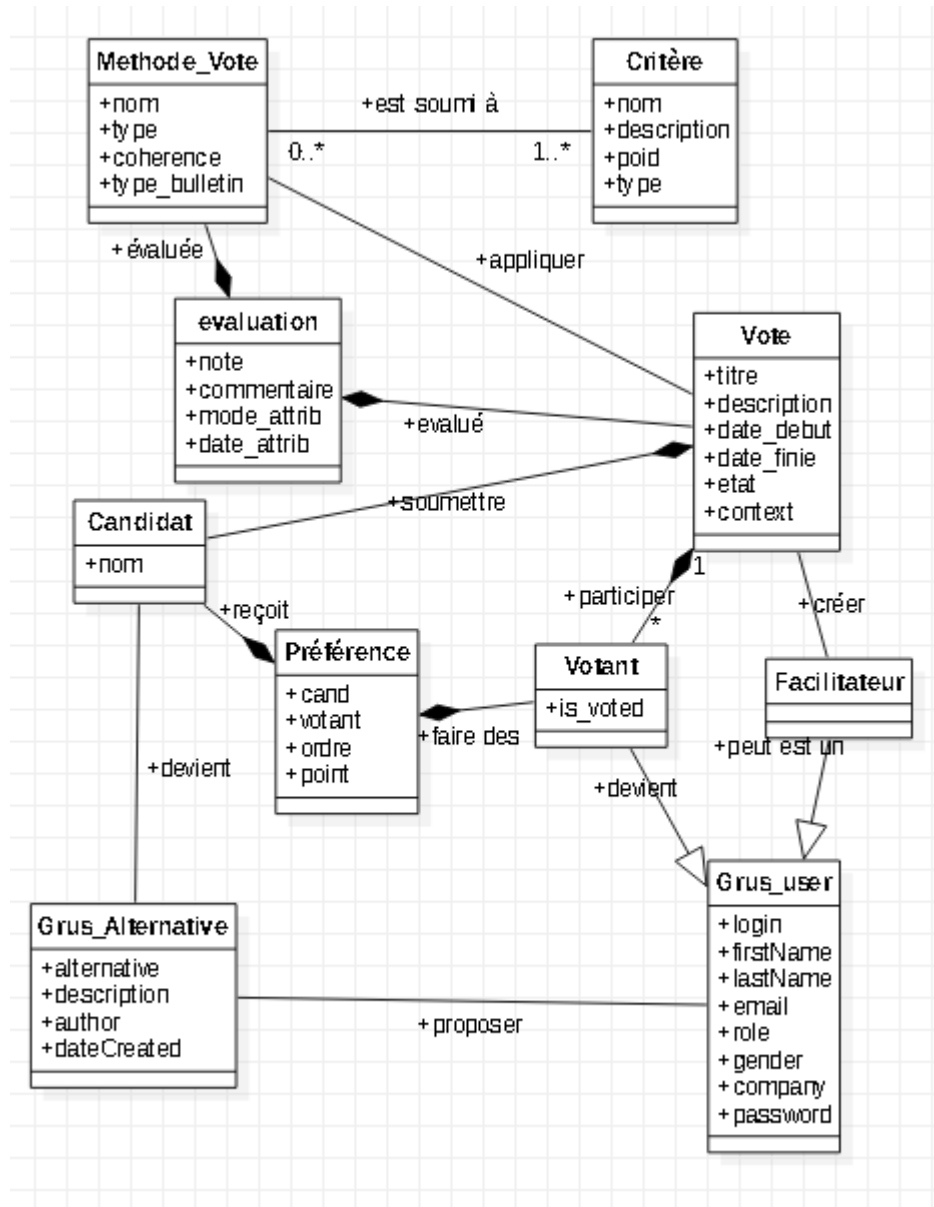


Figure 4-1: Diagramme de Classes de GRECO

Entité Vote

Dans GRECO, un vote est caractérisé par :

- Un titre, une description : ses deux attributs servent pour nommer et décrire le vote en instance de création
- Une date de début et de clôture : pour préciser la période de disponibilité du vote,
- Un contexte,
- Un état, pour savoir si le vote est terminé, en cour ou suspendu,
- Une liste de candidats, pour lesquels le vote est organisé,

- Une liste de votants, qui vont participer pour déterminer le ou les vainqueurs du vote.

Entité Candidat

Le vote est organisé pour choisir entre des candidats proposés par les participants de la réunion. Dans les prises de décision, les participants peuvent soumettre des alternatives ou des ensembles d'alternatives appelé Clusters. Les candidats du vote peuvent être ses alternatives ou ses clusters. Un candidat est caractérisé par

- Le nom, qui décrit le candidat
- Le type (simple ou cluster) qui désigne si le candidat est une alternative pour un groupe d'alternative proposé dans la réunion de prise de décision.

Entité Votant

Les votants forment l'élément central dans un vote. C'est la contribution des votants qui permet de déterminer le résultat d'un vote. Dans GRECO, ce sont les différents participants à la réunion qui peuvent devenir des votants. Ainsi un votant est une spécialisation de *Grus_User*. Un votant est caractérisé par :

- L'identifiant, qui reconnaît un votant dans le système.
- L'attribut *is_voted* de type booléen permettant de savoir s'il a voté ou non.

Entité Préférence

La participation d'un votant à un vote se matérialise par des préférences émises par ce dernier. Une préférence est caractérisée par :

- Le candidat,
- Le votant,
- L'ordre
- Le score obtenu (dans le code appelé point).

Entité Méthode de vote

Pour faciliter la mise en place du moteur de recommandation, une procédure de vote est caractérisée par :

- Un nom, décrit pour l'appellation de la procédure (par exemple : Borda, Condorcet, etc)

- Le type de vote, qui peut être par classement et sans classement
- Le type de bulletin, liste ou un seul élément.
- Cohérence, si c'est une procédure qui est compatible avec la méthode de Condorcet ou non.

Entité Critère

C'est une entité qui permet de caractériser les différents critères qui sont utilisés pour évaluer les procédures de vote (voir le point 3.6 du chapitre 3)

Un critère est caractérisé par :

- Le nom, pour décrire le critère
- Une description permet d'expliquer brièvement le critère
- Le poids, pour savoir si l'importance du critère
- Le type si c'est un critère simple ou conditionnel

Entité évaluation

C'est un élément principal d'un système de recommandation. Elle va nous permettre de recueillir les informations des évaluations effectuées par les facilitateurs sur les procédures de vote lors des réunions de prise de décision, elle est composée de :

- La valeur de la note suivant le **Tableau 4-2**, (Tableau de notation du moteur de recommandation)
- Le commentaire, permet au facilitateur de rédiger ses impressions verbales sur le résultat du vote
- L'identifiant de la procédure de vote
- Le mode de notation, si c'est automatique ou manuel et
- La date d'attribution.

4.3. Technologies utilisées

4.3.1. Django

Django est un Framework web gratuit et libre écrit en Python. Un Framework web est un ensemble de composants qui permet de développer des sites et des applications web plus rapidement et plus facilement. Il met l'accent sur la réutilisation des composants, le développement rapide et le principe de la non répétition.

Il suit l'architecture MVT axée autour de trois pôles : le modèle, la vue et le Template (gabarit). Le modèle interagit avec la base de données. Sa mission est de chercher dans la base de données les tuples correspondant à une requête et de renvoyer une réponse facilement exploitable par le programme en s'appuyant sur le ORM (Object Relational Mapping) ou Mappage relationnel des objets.

Le gabarit, est fichier HTML qui peut recevoir des objets Python et qui est lié à une vue. Il interprète et affiche les variables. La vue joue un rôle central dans le l'architecture MVT, elle reçoit les requêtes HTTP et y répond via le navigateur.

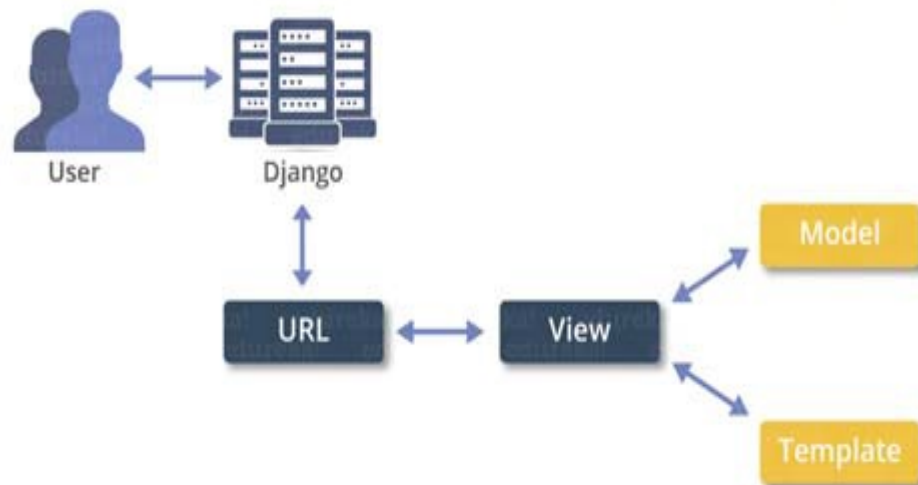


Figure 4-2: Architecture interne de Django

4.3.2. Pandas, numpy et scipy

Pandas et **numpy** sont des bibliothèques python qui permettent de manipuler facilement des données à analyser. Numpy est optimisé pour l'exploitation des matrices contenant des données de même type. Tandis que Panda avec ses *DataFrames* permet des opérations de jointure des tables contenant différents types de données (des nombres, des chaînes de caractères, des booléens, des dates, etc.)

Scipy propose un ensemble de bibliothèques mathématiques regroupées par domaines (analyse numérique, algèbre linéaire, statistiques, traitement du signal, traitement d'images, ...)

4.3.3. MySQL

Le système de gestion de Base de données (SGBD) utilisé pour la persistance des entités est MySQL. C'est un serveur de base de données relationnelle Open Source qui utilise le langage SQL (acronyme de *Structured Query Language*) universellement reconnu qui permet d'interroger et de modifier les contenus des différentes tables.

4.4. Algorithme des méthodes de votes codées

Dans le système GRECO, avant l'application de chaque procédure de vote on considère que le vote est créé avec toutes les données nécessaires à savoir les candidats, les votants. Chaque votant doit d'abord participer au vote en soumettant sa liste de préférence. Les sections suivantes décrivent brièvement l'implémentation de quelques procédures de vote présentées dans le chapitre I qui sont implémentées dans GRECO.

4.4.1. Méthode de pluralité

Pour coder cette procédure de vote on se base sur les prérequis suivantes : Avoir une liste contenant les premiers des listes de préférence des votants

- Calculer le nombre de voix d'un candidat x utilisant la formule **FI** du **chapitre 1**, qui est égale au nombre de fois qu'il apparaît sur la liste des premiers
- Retourner une liste contenant le candidat et son nombre de voix

Données

Lcand : liste des candidats

Lvotants : liste des votants

vote_id : identifiant d'un vote donné

Début

Lcand \leftarrow AvoirListeCandidat(vote_id)

Lvotants \leftarrow AvoirListevotant(vote_id)

cantV \leftarrow Lvotants.count

Pour i allant de 1 à cantV

tempV \leftarrow Lvotants[i]

Lpref ←AvoirVotantPreference(tempV)

Lpremiers.ajouter() ←Lpref[0]

Fin pour

candC ←Lcand.count

Pour j allant de 1 à candC

tempC ←Lcand[j]

nbVoix ←Lpremiers.compter(tempC)

eltRes ← (tempC,nbVoix)

LresultatPlur.ajouter() ←eltRes

Fin pour

Retourner LresultatPlur

Fin algorithme pluralité

4.4.2. Méthode de Borda

- Avoir la liste des candidats et la liste des votants
- Pour chaque candidat, calculer son score sur chaque liste de préférence de chaque votant en appliquant la formule *FI.4* (voir chapitre 1)
- Garder sur une liste le score et le nom du candidat

Données

Lcands : liste des candidats

Lvotants : liste des votants

Vote_id : identifiant d'un vote existant dans le système

Début

Lcand ←AvoirListeCandidat(vote_id)

Lvotants ←AvoirListevotant(vote_id)

cantV ←Lvotants.count

candC ←Lcand.count

Pour i allant de 1 à candC

```
tempC ← Lcands[i]
scoreCand ← 0
Pour j allant de 1 à cantV
    tempV ← Lvotants[j]
    scoreCand+ ← calculScoreBorda(tempC, tempV)
Fin pour
eltRes ←-[tempC, scoreCand]
ListResulBorda.ajouter() ←-eltRes
Fin pour
```

Fin Algorithme Borda

4.4.3. Méthode Condorcet

- Avoir la liste des candidats et la liste des votants
- Déterminer la liste de tous les duels possibles entre les candidats
- Calculer les scores des duels en faisant des comparaisons sur les listes de préférence de chaque votant et garder le nom du gagnant sur une liste, voir les formules **F1.6**, **F1.7** et **F1.8** (voir chapitre 1)
- Déterminer le nombre de duels gagnés par chaque Candidat
- Avec n candidats, si le nombre de duels gagnés par un candidat x est $n-1$ alors il est le gagnant de Condorcet

Données

Lcands : liste des candidats

Lvotants : liste des votants

Vote_id : identifiant d'un vote existant dans le système

Début

Lcand ← AvoirListeCandidat(vote_id)

Lvotants ← AvoirListevotant(vote_id)

cantV ← Lvotants.count

candC ← Lcand.count

Lduels ←determineDuels(Lcand)

LResDuel ←[]

nbDuels ←Lduels.count

Pour i allant 1 à nbDuels

tempDuel ←LResDuel[i]

dgagnant ←confrontation(tempDuel)

LResDuel.ajouter() ←dgagnant

Fin pour

Pour i allant i à Lcand.count

tempC ←Lcand[i]

nbVict ←LResDuel.compter(tempC)

eltResCond ← [tempC, nbVict]

ResultatCondorcet.ajouter ←eltResCond

Fin pour

Fin Algorithme Condorcet

4.4.4. Méthode Copeland

- Avoir la liste des candidats et la liste des votants
- Déterminer la liste de tous les duels possibles entre les candidats
- Déterminer les scores de Copeland à partir d'un duel en utilisant la formule **F1.8** du chapitre 1, avec une modification qui est égale à 1 pour le gagnant du duel, -1 pour le perdant et vaut 0.5 en cas d'égalité.
- Déterminer le score total de chaque Candidat
- Celui qui a le plus grand score est le gagnant de Copeland

Données

Lcands : liste des candidats

Lvotants : liste des votants

Début

```
Lcand ←AvoirListeCandidat(vote_id)
Lvotants ←AvoirListevotant(vote_id)
cantV ←Lvotants.count
candC ←Lcand.count
Lduels ←determineDuels(Lcand)
LResDuel ← []
nbDuels ←Lduels.count
Pour i allant 1 à nbDuels
    tempDuel ←LResDuel[i]
    dgagnant ←confrontationCopeland(tempDuel)
    LResDuel.ajouter() ←dgagnant
Fin pour
Pour i allant i à Lcand.count
    tempC ←Lcand[i]
    nbVict ←LResDuel.compter(tempC)
    eltResCond ←[tempC, nbVict]
    ResultatCopeland.ajouter ←eltResCond
```

Fin pour

Fin Algorithme Copeland

4.4.5. Méthode de Black

- Avoir la liste des candidats et la liste des votants
- Appliquer la méthode de Condorcet
- S'il y a un gagnant c'est le gagnant de Black et la méthode s'arrête
- Si non appliquer la méthode de Borda et le gagnant est le gagnant de Black

Données

Lcands : liste des candidats

Lvotants : liste des votants

Début

appliquer_condorcet

Bgagnant ← ResultatCondorcet

Si Bgagnant alors

resBlack ← Bgagnant

Si non

appliquer_Borda

resBlack ← ResultatBorda

Fin si non

Retourner resBlack

Fin Algorithme

4.5. Caractérisation des procédures de vote

En se basant sur les travaux (SUITT, et al., 2014), (NURMI, 2012), (DURAND, 2000) (KONCZAK, et al., 2005) (FELSENTHAL, et al., 2018) et tenant compte que GRECO est un outil d'aide à la décision en groupe de taille réduite, nous avons établi dans le chapitre 3 le **tableau 3-5**. Ce tableau est une caractérisation des procédures de vote étudiées dans ce manuscrit. Les critères dégagés sont C_1 : Critère du gagnant de Condorcet, C_2 : critère de majorité absolue, C_3 : Critère de Pareto, C_4 : critère du perdant, C_5 : critère de participation, C_6 : Critère de monotonie et C_7 : Critère de cohérence.

Nous obtenons M_p une matrice de caractérisations des procédures implémentées comme montre le **Tableau 4-1**.

Caractéristiques → Procédures	Type	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Pluralité	NRV	1	2	0	2	0	0	0
Borda	RNC	1	2	0	0	0	0	0
Condorcet	RCC	1	0	0	0	1	0	1
Black	RCC	0	0	0	0	1	0	1
Copeland	RCC	0	0	0	0	1	0	1

Tableau 4-1: Matrice de caractérisation des procédures de vote de GRECO (M_p)

En considérant, 0 : le critère n'affecte pas la procédure de vote, 1 : le critère affecte la procédure de vote et 2 : le critère a un fort impact sur la procédure. NRV : procédure sans classement,

RCC : procédure par classement Condorcet cohérent et RNC : procédure par classement non Condorcet cohérent.

Le système GRECO donne la possibilité aux facilitateurs de faire des appréciations sur les procédures de votes qu'ils vont utiliser dans les prises de décision. Le facilitateur peut utiliser l'échelle de notion suivante :

Appréciation	Mauvais	Insuffisant	Passable	Satisfaisant	Bien	Excellent
Note	0	1	2	3	4	5

Tableau 4-2: Liste des notes possibles

Nous utilisons les situations suivantes pour la notation des différentes procédures de vote :

Situation 1 :

Le facilitateur peut décider d'appliquer une procédure donnée, donc une sélection manuelle. Ce choix implique que la procédure est connue et appréciée par le facilitateur. Si le facilitateur confirme son choix, le système attribue à la procédure choisie une note de '**Bien**'. Ce mécanisme permet au système de ne pas avoir le problème de démarrage dans l'émission des recommandations connue sous le nom de '**Cold- Start**'.

Situation 2 :

Le système peut proposer automatiquement au facilitateur, une liste de procédures de vote à appliquer. Si le facilitateur confirme son choix, le système attribue la note '**Bien**' à la procédure choisie.

Situation 3 :

Après l'application d'une procédure de vote dans un contexte donné, les participants à la prise de décision peuvent adresser le degré de satisfaction globale du groupe au facilitateur. Cela permet d'attribuer une note à la procédure utilisée. Cette note est très importante car elle provient du groupe de décideurs.

Les différentes notes des facilitateurs, permettent de dresser une matrice M_n , contenant les notes sur les procédures de vote. Cette matrice peut être creuse, car certaines procédures de vote peuvent ne pas être encore notée par certains facilitateurs, comme montrée sur le **Tableau 4-3**.

Procédure→ Facilitateur	<i>Borda</i>	<i>Condorce</i>	<i>Black</i>	<i>Pluralité</i>	<i>Copeland</i>
<i>Fac₁</i>	5	5		3	

Fac_2			3		5
Fac_3		4			
.....					
Fac_n	3				4

Tableau 4-3: Matrice des notes des facilitateurs (M_n)

4.6. Moteur de recommandation de GRECO

Nous avons utilisé le Framework Django pour développer GRECO en suivant la Conception d'hybridation parallèle comme montre le **figure 4-3**. Ce Framework est basé sur Python qui nous offre des librairies comme Pandas, Numpy, Scipy, etc. qui facilitent l'implémentation des fonctionnalités souhaitées.

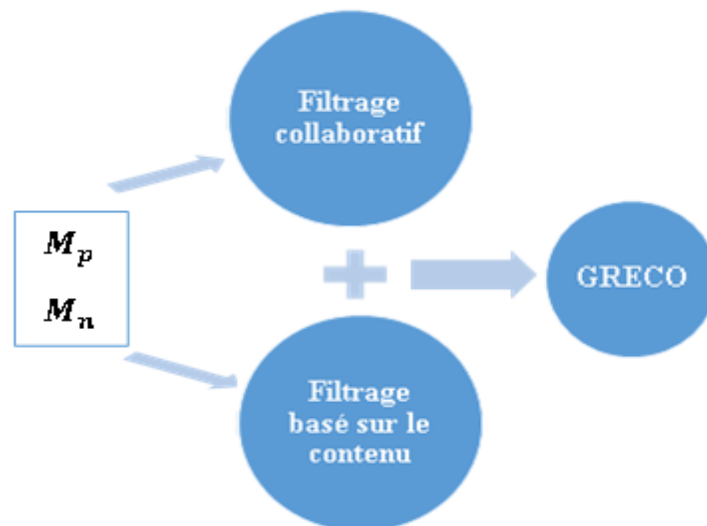


Figure 4-3: Logique interne de GRECO

Les algorithmes qui suivent expliquent nos différents mécanismes de recommandation.

Algorithme1 ReccommantionObjet_GRECO

/*Pour le filtrage basé sur le contenu dans notre système. Elle utilise les caractéristiques des procédures pour justifier les recommandations proposées aux facilitateurs. */

Données

Entrées

M_p : Matrice de caractérisation des procédures de vote

M_n : Matrice des notes des procédures (Notation des facilitateurs)

Facc : Facilitateur

Sortie :

Liste_recom_ob : Liste de procédures de votes à proposer au facilitateur connecté par recommandation objet.

Début

1- Construire un profil utilisateur à partir des procédures de vote déjà utilisées dans les réunions de prise de décision utilisant les Matrices M_p et M_n

1.1-Centrage des notes des procédures émises par les facilitateurs, nous aurons une matrice centrée $M_c \leftarrow M_n$

1.2- Calcul des coordonnées pour chaque caractéristique du profil utilisateur

2-Recherche des profils des procédures similaires

2.1-Indexer chaque procédure de vote par ses caractéristiques de la matrice M_n

2.2-Chercher les k profils des procédures de vote les plus similaires au profil d'utilisateur en utilisant le modèle vectoriel similarité cosinus.

Liste_recom \leftarrow ProcedureVotesimilarité (Facc, M_n)

3- retourner **Liste_recom**

Fin Algorithme

Algorithme2 RecommantionCollab_GRECO

/*Pour le filtrage collaboratif dans notre système. Elle utilise les avis des facilitateurs qui ont déjà utilisé le système dans des occasions passées pour justifier les recommandations proposées */

Données

Entrées

M_p : Matrice de caractérisation des procédures de vote

M_n : Matrice des notes des procédure (Notation des facilitateurs)

Facc : Facilitateur

Sortie :

Liste_recom_fc : liste de procédures de votes à proposer au facilitateur connecté par recommandation collaborative.

Début

- 1- Obtenir la liste des identifiants des procédures de vote évaluées par le facilitateur,
Liste_prod_eval ← **Get_procedure(Facc)**
- 2- Obtenir le cluster auquel le facilitateur appartient
user_cluster_name ← **Get_cluster_name(Facc)**
- 3- Obtenir les autres facilitateurs qui appartiennent au cluster du facilitateur connecté
user_cluster_other_members ← **Get_cluster_member(user_cluster_name)**
- 4- Obtenir les noms de procédures de vote déjà évaluées par ses facilitateurs
users_evals ← **Get_procedures_evaluation(user_cluster_other_members)**
- 5- Ordonner la liste des procédures en fonction des notes attribuées et retourner la liste.
Liste_recom_fc ← **Sorted_list(users_evals)**
- 6- Retourner **Liste_recom_fc**

Fin algorithme

Dans notre recommandation hybride, nous utilisons les résultats des deux algorithmiques *RecommantionObjet_GRECO* et *RecommantionCollab_GRECO* pour proposer une liste de procédures de vote susceptibles correspondre aux attentes du facilitateur qui est en train d'utiliser GRECO. (Voir Figure 4-3). Les deux algorithmes retournent des listes contenant les références des procédures de vote pouvant servir le facilitateur. Donc nous faisons dans le code, une concaténation des deux listes tout en éliminant des doublons et donnant une priorité à la procédure dont les références se répètent sur la liste concaténée.

4.7. GRECO en image : Exemple d'utilisation

Nous allons utiliser un exemple pour présenter le système GRECO. Soit un comité de décideurs de 5 membres qui veut choisir un lieu de rencontre pour célébrer leur réunion annuelle de conseil d'administration. Trois hôtels (Azalai, Grand Micasa et Radison Blu) sont en lice. Dans la réunion, ils ont utilisé GRECO pour déterminer l'hôtel qui va abriter la réunion suivante les préférences montrées dans le *tableau 4-4*.

Nb votants→		2	2	1
Rank	1 ^{ier}	Radison Blu	Grand Micasa	Radison Blu
	2 ^{ème}	Azalaï	Azalaï	Grand Micasa
	3 ^{ème}	Grand Micasa	Radison Blu	Azalaï

Tableau 4-4: Préférences des votants-GRECO

Pour créer un vote correspondant à la situation décrite dans cet exemple, nous avons besoin de trois étapes :

Etape 1 : Création des données de base du vote (Voir Figure 4-4)

- Saisie des informations de base du vote : Titre, description, les dates de début et fin du vote et les statuts (en cours, suspendu ou terminé).
- Ajouter les différents candidats qui seront soumis au vote. Ce sont les alternatives proposées durant la réunion.
- Ajouter les votants, qui sont les participants de la réunion en cours donc les cinq décideurs.

Après la spécification de ces informations, le clic sur le bouton « *Create the vote* » permet de valider et de passer à l'étape 2.

1. Vote creating » 2. Vote settings » 3. Procedure to be applied

New vote

Title: * : Choosing the annual meeting place for the Board of Directors
 Description: * : election of a meeting place

Start date: : 2019-3-15 (a)
 End date: : 2019-3-20
 Status: in progress

Alternative to Candidate

Alternatives proposed at the meeting: Sherathon
 Add to vote »
 « Remove from

List of candidates for the vote: Grand MiCasa, Azalai, Radison Blu

User to voter

Meeting Participants: adama, idi, mdrame
 Add to vote »
 « Remove from

Voters List: acoulibaly, gcamilleri, kjacqueline, pzarate, ftangara

+ Create the vote

Figure 4-4: Ecran de données de base du vote

Etape 2 : Paramètres du vote (Voir Figure 4-5)

- a. Les informations de base saisie dans l'étape antérieure sont affichées à savoir les titres, la description les dates, le nombre de votants et de candidats.
- b. Précision des paramètres de la recommandation
 - **Type de d'attribution**, précise comment la procédure recommandée sera associée au vote en cours de création. Il prend deux valeurs possibles (Automatique ou Manuel). Si c'est Manuel, le système après l'obtention des procédures recommandées, présente cette liste et le facilitateur choisit une d'entre elles. Par contre si c'est 'Automatique', le système attribue automatiquement la procédure mieux classée de la liste de recommandation utilisant notre filtrage hybride. Dans notre exemple, on a choisi la valeur 'Automatique'

The screenshot shows a web interface for creating a vote. At the top, there are three steps: '1. Vote creating', '2. Vote settings', and '3. Procedure to be applied'. The 'Voting data' section is highlighted in blue and contains the following information:

- Title:** Choosing the annual meeting place for the Board of Directors
- Description:** election of a meeting place
- Date start:** 15 mars 2019
- End date:** 20 mars 2019
- Voter(s):** 5
- Candidats:** 3

The 'Recommendation parameters' section is highlighted in green and contains the following information:

- Type of procedure:** Don't matter (marked with a red circle 'b')
- Choice of method:** Automatic

A red arrow labeled '(a)' points to the 'Candidats' count. A blue arrow points to the 'Suivant >>' button at the bottom right.

Figure 4-5: Ecran de paramètres de recommandation

Après la spécification de ces paramètres, un clic sur le bouton « Suivant », permet de valider les informations et passer à l'étape 3.

Etape 3 : Validation de la procédure de vote à utiliser (Voir Figure 4-6)

C'est l'étape finale de la création de notre exemple. Le facilitateur confirme la procédure de vote que le système a recommandé. Dans notre exemple le système a proposé la procédure de Condorcet.

Le clic sur le bouton 'Terminer', permet de conclure la création du vote et ajoutant toutes les informations dans la base de données.

1. Vote creating » 2. Vote settings » 3. Procedure to be applied

Voting data:

Title: Choosing the annual meeting place for the Board of Directors

Description: election of a meeting place

Date start : 15 mars 2019

End date: 20 mars 2019

5 Voter(s) **3 Candidats**

Voting procedure

List of recommendations

Condorcet ← Recommended procedure

Terminer

Figure 4-6: Ecran de précision de la procédure recommandée

Une fois le vote crée avec les données de l'exemple, les votants peuvent participer au scrutin en soumettant leur préférence suivant le **tableau 4-4**. Par exemple le figure 4-6 montre la préférence d'un votant (Grand MiCasa>Azalai>Radison Blu).

RECOMMENDATION User (pzarate) Langue

Vote participation

Title: Choosing the annual meeting place for the Board of Directors

Description: election of a meeting place

Date start: 15 mars 2019

Date end: 20 mars 2019

Making preference list

Vote candidates list

a)

Add to vote »

« Remove from

Vote's candidate preference list

Grand MiCasa
Azalai
Radison Blu

b)

Submit the ballot

Figure 4-7: Ecran de participation au vote

Quand tous les votants soumettent leurs préférences, GRECO détermine le résultat du vote en utilisant la procédure recommandée. C'est l'hôtel **Radison Blu** qui va abriter le conseil annuel du comité comme montre l'écran suivant



Figure 4-8: Ecran du résultat du vote

En cliquant sur le bouton « **voting procedure scoring** », le facilitateur peut noter la procédure appliquée, dans cet exemple c'est la procédure de Condorcet comme le montre la **Figure 4-9**.

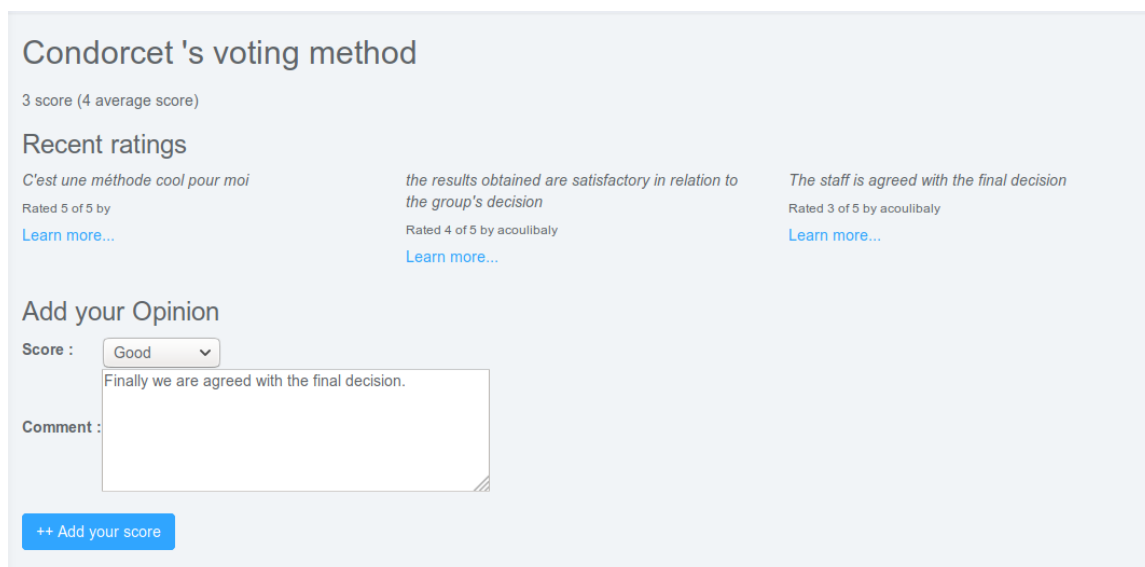


Figure 4-9: Ecran de notation de la procédure de vote

4.8. Conclusion

Nous avons présenté l'architecture détaillée de GRECO à travers la description des entités nécessaires à savoir : *vote, candidat, votant, préférence, méthode de vote, critère et évaluation*. La technologie utilisée, composée de Django, Panda, Numpy et MySQL pour l'implémentation de notre solution, a été brièvement introduite.

Le choix d'une procédure de vote qui est l'objet principal d'étude a été décrit. Le facilitateur peut sélectionner directement une procédure de vote à utiliser dans la prise de décision. Il a aussi le choix de se laisser assister par GRECO qui peut lui proposer des procédures adaptées au contexte de la réunion.

Le fonctionnement interne du moteur de recommandation a été expliqué. De plus nous avons proposé un algorithme de recommandation hybride qui utilise le filtrage en base de contenu et un filtrage collaboratif.

Des captures d'écran ont été proposées et les algorithmes implémentant les différentes procédures de vote ont été expliqués en pseudo code permettant une meilleure compréhension technique du lecteur. La démonstration et la présentation de quelques fonctionnalités de GRECO ont été faites en s'appuyant sur la résolution d'un exemple concret utilisant le résultat d'une recommandation.

CHAPITRE 5

5. Validation expérimentale et Benchmark

5.1. Introduction

Les systèmes de recommandation sont de plus en plus populaires tant sur le plan commercial que dans le milieu de la recherche, où de nombreuses approches ont été suggérées pour formuler des recommandations. Afin de pouvoir comparer ces différents systèmes, il faut mettre en place diverses méthodes qui vont permettre de poser une base à partir de laquelle on va pouvoir comparer les différents systèmes de recommandation.

5.2. Type d'expérimentation

L'évaluation de systèmes de recommandation suit, en général, une des trois méthodes : hors ligne, études sur un échantillon d'utilisateurs ou évaluation en ligne. (SHANI, et al., 2011)

5.2.1. Expérimentation hors ligne

L'évaluation hors ligne est la plus simple à réaliser et la moins risquée. Il s'agit globalement de diviser les données disponibles en deux parties, la partie d'apprentissage et la partie test, avant d'utiliser la partie d'apprentissage pour prédire la partie test. Ce type d'évaluation ne pose pas de problème de fuite d'utilisateurs, on peut ainsi prendre le risque de tester des approches très fluctuantes. Elle permet d'intégrer facilement une grande masse d'utilisateurs. Elle n'est pas très sensible aux potentiels changements dans le comportement de l'utilisateur. Elle essaie de reproduire un comportement que l'utilisateur a eu alors que le système de recommandation n'intervenait pas pour conseiller l'utilisateur, mais elle n'est pas capable de mesurer l'impact d'une telle intervention (SHANI, et al., 2011)

5.2.2. Expérimentation basée sur un échantillon d'utilisateurs

Cette méthode consiste à recruter un groupe de volontaires, auxquels on demande d'exécuter des tâches bien précises en utilisant le système de recommandation, de surveiller et d'enregistrer leurs comportements durant l'expérimentation. Ensuite, on peut éventuellement poser des questions aux participants concernant leurs impressions sur l'expérimentation et le système de recommandation.

Ce type de test peut répondre à un large éventail de questions. Il permet de suivre le comportement d'un utilisateur au cours de son interaction avec le système de recommandation et d'observer si ce dernier a influencé le comportement de l'utilisateur. Le questionnaire permet aussi de collecter des données qualitatives pour expliquer les résultats quantitatifs.

Par contre ce type d'évaluation est coûteux car il n'est pas facile de recruter un nombre suffisant d'utilisateurs, parfois il faut les motiver par des récompenses ou des dédommagements. Le

nombre de participants est souvent limité, et on ne peut pas tirer de conclusions concernant un grand nombre d'utilisateurs. De plus, à cause des contraintes de temps des participants, on ne peut pas leur demander de faire des tests excessivement longs. Néanmoins, chaque scénario doit être répété plusieurs fois afin d'assurer la fiabilité du résultat (SHANI, et al., 2011).

5.2.3. Expérimentation par évaluation en ligne

Elle est appliquée sur les vrais utilisateurs du système en temps réel (KOHAVI, et al., 2009). Ce test peut être une simple comparaison des chiffres d'affaires avant et après l'application du système de recommandation. On l'applique sur un échantillon d'utilisateurs (tirés au hasard), on observe leurs réactions, et on les compare avec ceux du reste de la population. Ce type d'évaluation comporte des risques. En effet, on peut perdre un utilisateur si le système recommande des items non pertinents. Pour cette raison, on recommande de procéder préalablement à une évaluation hors risque afin de garantir un minimum de qualité de recommandation (SHANI, et al., 2011).

5.2.4. Choix du type expérimentation

Chaque type d'expérimentation comporte des avantages et des limites. Il est possible d'appliquer les trois séparément pour valider un système.

Etant donné que GRECO est un outil de facilitation qui sera généralement déployé dans un environnement dont l'accès est souvent restreint et destiné à des groupes de prise de décision, donc l'expérimentation hors ligne sera mieux adaptée pour l'évaluer. Ce type d'expérimentation est facile à mettre en place et permettra facilement de reproduire le comportement de l'utilisateur qui est le facilitateur.

L'expérimentation d'un système renferme des métriques d'évaluation. Ces métriques sont nombreuses, les points suivants décrivent certains aspects.

5.3. Les mesures ou outils disponibles

Afin d'effectuer des tests permettant d'obtenir un indice de performance correspondant à la capacité qu'a un système à produire de bonnes recommandations nous avons recherché des métriques. De bonnes recommandations permettent de satisfaire les utilisateurs en plaçant dans les premières places de listes qui leur sont présentées, des éléments qui leur plaisent.

Voici un aperçu des différentes méthodes qui existent afin d'obtenir un indice permettant de représenter la capacité qu'a un système à faire de bonnes recommandations.

Ces méthodes peuvent être rangées en 3 catégories. La première contient les méthodes pour mesurer la précision d'un système effectuant de la prédiction de note. La seconde contient les méthodes de mesure de prévision d'utilisation. Et enfin, la troisième contient les méthodes permettant d'évaluer les systèmes mettant en œuvre des tris des solutions/objets proposés. Ces méthodes sont tirées de (SHANI, et al., 2011).

5.3.1. Mesures de précision prédictive

Dans certaines applications, comme dans le service de location de film de Netflix (netflix.com), le système de recommandation tente de prédire la note qu'un utilisateur donnerait à un élément (de 1 à 5 étoiles). Dans de tels cas, afin de mesurer la performance du système, il est nécessaire de mesurer la précision des prédictions.

La racine de l'erreur moyenne quadratique, ou **Root Mean Squared Error** (RMSE) est peut-être la métrique la plus couramment utilisée pour évaluer la précision des notes prédites. Le système génère des notes prédites \hat{r}_{ui} pour un ensemble de tests T pour des paires utilisateur-objet (u, i) pour lesquels les vraies notes r_{ui} sont connues.

En règle générale, les r_{ui} sont connues bien qu'elles soient cachées lors d'une expérience hors ligne, ou parce qu'elles ont été obtenues grâce à une étude sur les utilisateurs ou une expérience en ligne.

L'écart quadratique moyen entre la note prédite et la note réelle est donné par la formule suivante.

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2} \quad (F5.1)$$

Une alternative très utilisée à RMSE est la mesure de l'erreur absolue moyenne **Mean Absolute Error** (MAE).

$$MAE = \frac{1}{|T|} \sum_{(u,i) \in T} |\hat{r}_{ui} - r_{ui}| \quad (F5.2)$$

MAE évalue la qualité des prédictions fournies par le système de recommandation. À cet effet, le MAE mesure la déviation absolue moyenne entre une estimation prévue et l'estimation vraie de l'utilisateur.

Il existe d'autres mesures dans cette catégorie, pour plus d'informations voir (SHANI, et al., 2011).

5.3.2. Mesure de prévision d'utilisation

Dans de nombreux systèmes de recommandation, le système ne permet pas de prédire les préférences de l'utilisateur à propos d'éléments à lui proposer, tel que ce qui est fait lors de la prédiction de notes pour des films, mais essaie de recommander aux utilisateurs des éléments qu'ils pourraient utiliser (acheter), ou plus classiquement, qu'ils pourraient trouver intéressant (pertinent). Un exemple majeur de ces systèmes correspond aux moteurs de recherche.

Par exemple dans Netflix, lorsque des films sont ajoutés dans la liste personnelle des films de l'utilisateur (sa collection), le système propose un ensemble de films pouvant être intéressant, étant donné le film ajouté. Dans ce cas, nous ne nous intéressons pas à savoir si le système prévoit correctement les notations de ces films, mais plutôt de savoir si le système prévoit bien que l'utilisateur va ajouter ces films à sa collection (acheter le film).

Afin d'évaluer les prévisions d'utilisation, nous possédons généralement un ensemble de données constituées des éléments que chaque utilisateur a utilisés (ajouté à sa collection par exemple), et qui sont donc considérés comme étant pertinents pour l'utilisateur. Ensuite, nous sélectionnons un utilisateur, nous masquons une partie de ses sélections, et nous utilisons le système de recommandation afin qu'il prévoie un ensemble d'éléments que l'utilisateur va utiliser. Nous obtenons alors quatre résultats possibles pour les articles recommandés et cachés. Ces résultats sont présentés dans le tableau suivant :

	Élément recommandé	Élément non recommandé
Élément pertinent	Vrai positif (V_P)	Faux négatif (F_n)
Élément non pertinent	Faux positif (F_P)	Vrai négatif (V_n)

Tableau 5-1: Classification des résultats possibles d'une recommandation d'un élément pour l'utilisateur

Dans le cas où l'expérimentation se fait hors ligne, puisque les données ne sont pas généralement recueillies par le système de recommandation au cours de l'évaluation, nous sommes obligés de supposer que les éléments non choisis ne l'auraient pas été, et ce même s'ils avaient recommandé, c'est à dire qu'ils sont sans intérêt ou inutiles à l'utilisateur. Cette hypothèse peut être fautive, comme lorsque l'ensemble des articles non choisis contient quelques articles intéressants que l'utilisateur n'a pas choisis. Par exemple, un utilisateur peut ne pas avoir choisi un élément parce qu'il n'était pas au courant de son existence, mais après

que le système de recommandation lui ait proposé cet élément, l'utilisateur peut décider de le sélectionner. Dans ce cas-là, le nombre de faux positifs est surestimé.

Nous pouvons ainsi compter le nombre d'éléments de chaque case du tableau et calculer les quantités suivantes :

$$\mathbf{précision} = \frac{\#V_p}{\#V_p + \#F_p} \quad (F5.3)$$

La précision correspond au nombre d'éléments pertinents (V_p) rapporté au nombre d'éléments total proposé par le système.

$$\mathbf{rappel} = \frac{\#V_p}{\#V_p + \#F_n} \quad (F5.4)$$

Le rappel est défini par le nombre d'éléments pertinents par rapport au nombre d'éléments pertinents total.

$$\mathbf{Taux Vp} = \frac{\#F_p}{\#F_p + \#V_n} \quad (F5.5)$$

$$\mathbf{F_{-}mesure} = 2 * \frac{\mathbf{precision} * \mathbf{rappel}}{\mathbf{precision} + \mathbf{rappel}} \quad (F5.6)$$

Avec # désignant le nombre d'éléments.

Cette mesure allie précision et rappel. C'est la moyenne harmonique de la précision et du rappel, la traditionnelle F-mesure.

Un système de recommandation parfait fournira des réponses dont la précision et le rappel sont égaux à 1. Dans ce cas-là, l'algorithme trouve la totalité des éléments pertinents (rappel), et il ne fait aucune erreur (précision). Dans la réalité, les systèmes sont plus ou moins précis, et plus ou moins pertinents. Il sera possible d'obtenir un système très précis (par exemple un score de précision de 0,99), mais peu performant (par exemple avec un rappel de 0.10, qui signifiera qu'il n'a trouvé que 10% des réponses possibles). Dans le même ordre d'idée, un algorithme dont le rappel est fort (par exemple 0.99 soit la quasi-totalité des éléments pertinents), mais la précision faible (par exemple 0.10) fournira en guise de réponse de nombreux éléments non pertinents en plus de ceux pertinents : il sera donc difficilement exploitable.

5.3.3. Mesure d'ordre (ranking mesure)

Afin de mesurer l'ordre défini par un système de recommandation par rapport à un ordre de référence (bon ordre), plusieurs méthodes sont utilisées. Dans les cas où les notations (données explicites) sur les éléments, fournies par les utilisateurs, sont disponibles, nous pouvons classer

les articles les mieux notés dans l'ordre décroissant des notes, avec des éléments liés. Il est important de noter que ces liaisons n'ont rien à voir avec les liens/dépendance/interactions entre des attributs. Dans notre cas, quand une paire d'objets est liée dans le classement établi (a obtenu la même note, ou le même score), cela signifie que l'utilisateur est en fait indifférent entre les deux objets. Ainsi, un système parfait ne doit pas classer un objet comme étant supérieur à l'autre.

Dans le cas où nous disposons seulement de données d'utilisation, il peut être approprié de construire un classement de référence où les éléments choisis par l'utilisateur sont classés au-dessus des éléments non choisis. Cependant, ceci n'est valable que si nous savons que l'utilisateur connaissait ces articles non choisis, de sorte que nous pouvons en déduire que l'utilisateur a en effet préféré les éléments choisis plutôt que les articles non choisis, et ce en toute connaissance de cause.

5.3.3.1. Normalized Distance-based Performance Measure (NDPM)

Dans certains systèmes il est nécessaire de définir un ordre total entre les éléments, et ce sans prendre en compte la présence ou non d'éléments liés. Dans ce genre de systèmes, l'évaluation ne doit pas pénaliser un système qui ordonne un élément par rapport à un autre alors que ceux-ci sont liés dans l'ordre de référence.

Dans ce cas, on utilise la méthode **Normalized Distance-based Performance Measure** (NDPM (YAO, 1995)) définie comme suit :

Avec r_{ui} correspondant à l'index (place dans la liste) déterminé, et \hat{r}_{ui} correspondant à l'index de référence pour un objet et un utilisateur. Dans cette méthode on évalue toutes les paires possibles d'éléments constituant la liste. Cette évaluation des paires d'éléments consiste à déterminer si l'ordre défini par le système entre les éléments de la paire évaluée est le même dans l'ordre de référence.

Exemple :

Prenons une liste de référence (et donc un ordre de référence) : A, B, C. Et prenons une liste ordonnée générée par le système : A, C, B. La méthode va évaluer les paires suivantes : AB, AC et BC.

Afin de définir NPDM il nous faut définir quelques valeurs :

$$C^+ = \sum_{ij} \text{sign}(r_{ui} - r_{uj}) \text{sign}(\hat{r}_{ui} - \hat{r}_{uj}) \quad (F5. 7)$$

$$C^- = \sum_{ij} \text{sign}(r_{ui} - r_{uj}) \text{sign}(\hat{r}_{uj} - \hat{r}_{ui}) \quad (F5. 8)$$

$$C^u = \sum_{ij} \text{sign}^2(r_{ui} - r_{uj}) \quad (F5. 9)$$

$$C^s = \sum_{ij} \text{sign}^2(\hat{r}_{ui} - \hat{r}_{uj}) \quad (F5. 10)$$

$$C^{u0} = C^u - (C^+ + C^-) \quad (F5. 11)$$

Donc C^u correspond au nombre de paires d'éléments pour lesquels l'ordre de référence définit un ordre entre les éléments, c'est-à-dire le nombre de paires d'éléments non liés (et idem pour C^s , mais avec l'ordre défini par le système) et correspondent respectivement au nombre de paires d'éléments concordant et discordant. C'est-à-dire au nombre de paires d'éléments qui sont dans un ordre identique dans l'ordre de référence et l'ordre défini par le système. C^{u0} correspond au nombre de paires où le système de référence ne lie pas d'éléments, mais où le système le fait.

Si on reprend notre exemple :

La paire AB est concordante, la paire AC est concordante et la paire BC est non concordante. NDPM est donc défini comme suit :

$$NDPM = \frac{C^- + 0.5C^{u0}}{C^u} \quad (F5. 12)$$

Ainsi, la mesure NDPM donne un score parfait de 0 à un système qui permet de prédire correctement chaque relation de préférence par rapport à l'ordre de référence. Le pire score de 1 est attribué à un système qui contredit toutes les relations de préférence par rapport à l'ordre de référence. Ne pas prédire une relation de préférence est seulement pénalisé à moitié par rapport au fait de le contredire. Et également, on ne pénalise pas un ordre défini entre des éléments, qui sont liés dans l'ordre de référence (on ne connaît pas la préférence de l'utilisateur entre ces deux éléments).

5.3.3.2. Mesures de corrélation

Dans le cas où l'on possède un ordre total entre les éléments, et que lorsque deux éléments sont liés (cela signifie que l'utilisateur est en fait indifférent à la différence entre les deux objets.) l'on veut que le système ne donne pas d'ordre supérieur à l'un des deux éléments par rapport à l'autre, des mesures de corrélations (coefficient de corrélation) telles que celles de Spearman Rho ou Kendall Tau (KENDALL, 1938) (KENDALL, 1945) sont utilisées. On remarque que ces mesures tendent à être très similaires dans la pratique (FREDRICKS, et al., 2007).

Le coefficient de Spearman Rho est calculé de la façon suivante (di étant la différence entre le rang de l'observation i et celui de sa valeur, et n est le nombre d'éléments) :

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} \quad (F5. 13)$$

Cependant la méthode Kendall Tau possède également une variante, Kendall Tau-b, qui effectue des ajustements afin de mieux prendre en compte les éléments liés, c'est pourquoi nous allons définir uniquement les méthodes Kendall Tau et Kendall Tau-b. On remarque que dans le cas où des éléments sont liés, chaque item lié devrait se voir assigner un rang moyen dans la liste, c'est-à-dire que si deux éléments sont liés à la 2ème et 3ème place, alors on leur assigne un rang de 2.5. (KENDALL, 1945)

Le coefficient de corrélation de Kendall Tau est défini comme suit :

$$\tau = \frac{c^+ - c^-}{\sqrt{c^u} \sqrt{c^s}} \quad (F5. 14)$$

Il permet ainsi de mesurer à quel point l'ordre défini par le système est proche de l'ordre de référence. La valeur de ce coefficient est comprise entre -1 et 1. 1 correspond à un ordre défini par le système égal à l'ordre de référence. Tandis que -1 correspond à l'ordre inverse de l'ordre de référence. 0 correspond à la situation où l'ordre défini par le système n'est pas corrélé avec l'ordre de référence.

Afin de mieux prendre en compte les cas où les éléments sont liés entre eux, une variante dite Kendall Tau-b a été définie comme suit :

$$\tau_b = \frac{c^+ - c^-}{\frac{1}{2} \sqrt{(n(n-1) - U_x)(n(n-1) - U_y)}} \quad (F5. 15)$$

$$\text{Avec } U_x = \sum (u_x - 1) \text{ et } U_y = \sum u_y (u_y - 1) \quad (F5. 16)$$

Les symboles u_x et u_y correspondent respectivement au nombre d'apparitions de chaque valeur X et Y dans la liste d'éléments.

Pour donner quelques exemples, prenons une liste X définie par (A, B, C, D, E). Et une liste Y contenant les mêmes éléments, mais dans un ordre différent.

Coefficient de Kendall Tau-b par rapport à 2 listes	A, B, C, D, E
A, B, C, D, E	1
E, D, C, B, A	-1

A, B, C, E, D	0.8
A, B, E, C, D	0.6
A, C, D, E, B	0.4
A, C, E, D, B	0.2

Tableau 5-2: Exemple de valeur de coefficient de Kendall Tau-b

Comme on peut le voir dans le tableau 5-2, la valeur de 1 correspondant au classement parfait, et -1 au classement parfait inversé, et 0 correspondant à un classement ne se rapprochant pas de l'ordre parfait. On remarque que le fait d'avoir une valeur de 0.8 correspond à une situation presque parfaite (E a été déplacé d'un rang). De même, une valeur de 0.6 correspond à une situation très proche du parfait (E a été déplacé de 2 rangs). La valeur de 0.4 correspond à une situation qui commence à être moins correcte, mais qui tout de même offre un ordre de qualité acceptable (B a été déplacé de 3 rangs). La valeur de 0.2 correspond à une situation peu correcte, en effet l'ordre commence à ne plus être proche de la situation parfaite (B a été déplacé de 3 rangs et E de 1 rangs).

Ainsi on peut affirmer que des valeurs supérieures ou égales à 0.4 peuvent être considérées comme étant bonnes et celles supérieures à 0.6 comme étant très bonnes.

5.3.3.3. Méthode de K-cross fold

Pour effectuer des tests, on utilise fréquemment une méthode appelée K-cross fold. Cette méthode est utilisée dans le domaine de l'apprentissage pour effectuer des tests afin d'évaluer les capacités d'un algorithme d'apprentissage, et ce en réutilisant plusieurs fois la même base de test.

Voici son principe :

1. En fonction de k, on divise la base de données de test en k parties.
2. On sélectionne la première k partie, et on la met de côté, cette k partie sera utilisée en tant que base de données de validation, alors que les autres parties vont être utilisées en tant que base d'apprentissage,
3. On utilise dans le système, comme données en entrée, la base d'apprentissage que l'on vient de définir,
4. On teste ensuite le système sur la base de validation. Et grâce aux données obtenues, on va pouvoir utiliser diverses métriques pour évaluer le système.

5. On répète les opérations 2 à 4, en changeant à chaque fois la k partie utilisée, afin d'utiliser toutes les k parties en tant que base de validation.

Afin d'évaluer un système, cette méthode est souvent employée, et à l'issue des k validations, on calcule généralement la moyenne des résultats obtenus par les métriques employées.

5.4. Expérimentations de GRECO

Notre moteur de recommandation est basé sur une approche hybride comportant le filtrage basé sur le contenu, et le filtrage collaboratif.

Le filtrage basé sur le contenu est possible grâce à la caractérisation des procédures de vote comme étudié dans le chapitre 3. La communauté qui est derrière le filtrage collaboratif de GRECO va évoluer lentement parce qu'elle dépend du nombre de réunion du groupe et du facilitateur dans GRUS.

Les métriques précédemment décrites sont facilement applicables aux systèmes de recommandation (Amazon, MovieLens, Netflix, etc) renfermant une grande communauté d'utilisateurs et d'objets/Items. Dans GRECO le modèle est différent, car nous travaillons sur un outil d'aide à la prise de décision destiné à des groupes de taille réduite et les recommandations se font sur un nombre réduit de procédures de vote. Donc nous allons suivre une méthodologie de validation expérimentale personnalisée comprenant dans un premier temps un test des algorithmes des procédures de vote implémentés. Dans un second temps, nous allons procéder à une validation par benchmark des exemples classiques de la littérature de la théorie du vote en suivant le modèle hors ligne. Nous évaluerons la précision des notes attribuées aux procédures de vote implémentées en calculant le RMSE et le MAE pour notre système. Et nous terminerons par les calculs de la précision, le rappel et la F-mesure pour mesurer la précision d'utilisation.

5.4.1. Test des algorithmes des procédures de vote implémentés

Afin de tester le bon fonctionnement des algorithmes des procédures de vote, nous allons résoudre une situation de vote en calculant manuellement les résultats puis comparer à ceux fournis générés par GRECO.

Situation 1 :

Lors d'une réunion de gestion d'une entreprise, cinq experts doivent décider de l'adoption d'une nouvelle source d'énergie en ce qui concerne les questions environnementales. Les alternatives proposées sont : éolienne, solaire ou hydroélectrique. Le tableau suivant représente les préférences des électeurs.

Rang	V_1	V_2	V_3	V_4	V_5
1 ^{ier}	Eolien	Solaire	Hydraulique	Solaire	Eolien
2 ^{ième}	Hydraulique	Eolien	Solaire	Hydraulique	Solaire
3 ^{ième}	Solaire	Hydraulique	Eolien	Eolien	Hydraulique

Tableau 5-3: Préférence de votants de la situation 1

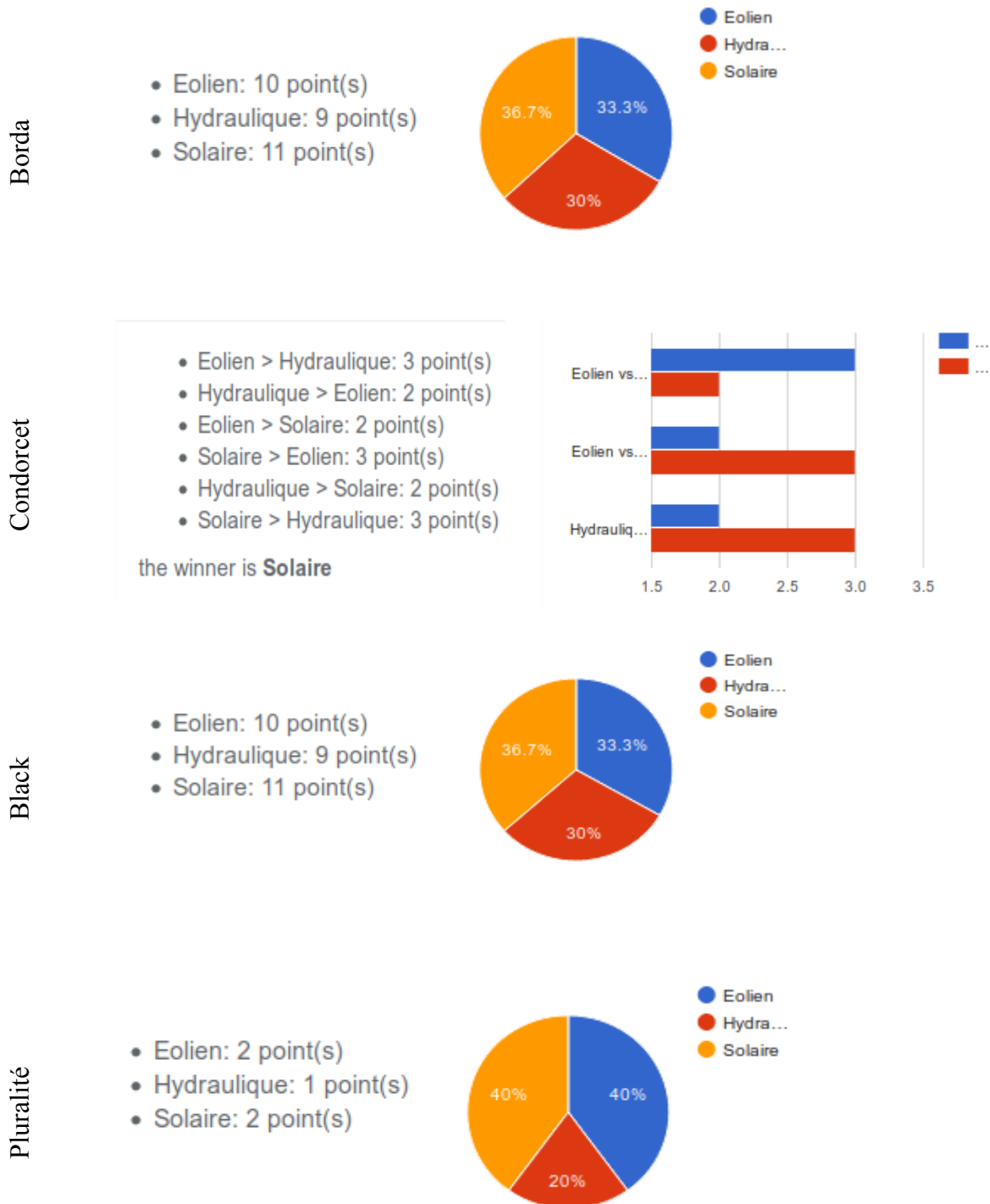
La solution manuelle de la situation antérieure donne les résultats montrés dans le tableau 5-4.

Méthode	Résultat
Pluralité	$S_{pl}(Solaire)=2, S_{pl}(Eolien)=2$ et $S_{pl}(Hydraulique)=1$
Borda	$S_{Borda}(Solaire) = (2*3) + (2*2) + 1 = 11$ $S_{Borda}(Eolien) = (2 * 3) + (1 * 2) + 2 = 10$ $S_{Borda}(Hydraulique) = (3) + (2*2) + 2 = 9$
Condorcet	Duel (<i>Solaire, Eolien</i>)=(3,2) =>Solaire Duel (<i>Solaire, Hydraulique</i>)=(3, 2) =>Solaire Duel (<i>Eolien, Hydraulique</i>)=(3,2) =>Eolien
Black	$S_{Black}(Solaire) = 11$ $S_{Black}(Eolien) = 10$ $S_{Black}(Hydraulique) = 9$
Copeland	$S_{Cop}(Solaire) = +1+1=+2$ $S_{Cop}(Eolien) = +1-1=0$

$$S_{Cop}(\text{Hydraulique}) = -1-1 = -2$$

Tableau 5-4: Résultats manuels situation 1

L'utilisation de GRECO pour déterminer le résultat du vote de la **situation 1**, donne les résultats suivants montrés dans le tableau 5.5.



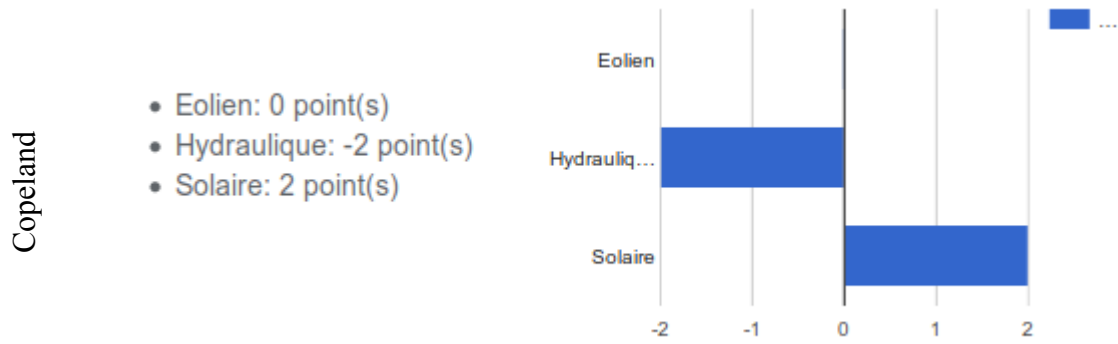


Tableau 5-5: Résultats de la situation 1 avec GRECO

Les résultats montrés dans les **tableaux 5.4** et **5.5** nous permettent de constater que notre implémentation de nos algorithmes donne des valeurs correctes. Les calculs manuels effectués par un humain coïncident parfaitement avec les calculs automatiques de GRECO. Ce qui permet d’apporter une aide précieuse au facilitateur.

5.4.2. Validation par benchmark des exemples classiques de la littérature

Dans cette section, nous allons utiliser quelques exemples classiques de la théorie du vote pour faire des validations sur GRECO. Nous utilisons deux cas de figures pour évaluer GRECO en suivant le protocole suivant :

- Soumettre des exemples typiques de vote à GRECO dont les résultats sont connus dans la théorie du vote.
- Observer la proposition de recommandation appliquée par GRECO
- Faire des conclusions.

Situation 2 :

Considérons une assemblée de 60 votants ayant le choix entre trois propositions A, B et C. Le **tableau 5-6** contient les préférences exprimées par ses votants.

Nombre de votants→		23	17	10	2	8
Rang	1 ^{ère}	A	B	C	B	C
	2 ^{ème}	B	C	A	A	B
	3 ^{ème}	C	A	B	C	A

Tableau 5-6: Situation 2- Préférences des votants

C'est un exemple typique utilisé pour expliquer le paradoxe de Condorcet (NICOLAS, 1785). En appliquant la procédure de Condorcet sur ces préférences, les résultats des duels entre les candidats sont les suivants.

Duel→	A versus B	A versus C	B versus C
	A>B = 33	A>C = 25	B>C = 42
	B>A = 27	C>A = 35	C>B = 28
Gagnant→	A	C	B

Tableau 5-7: Résultats des duels de Condorcet

Avec les résultats des duels nous avons, A qui gagne contre B et perd contre C. En plus C perd contre B, ce qui conduit à une préférence globale cyclique $A>B>C>A$, montrant ainsi la limite de la procédure de Condorcet dans ce cas particulier.

Analyse de la situation 2 dans GRECO

Dans GRECO, on crée un nouveau vote avec les informations suivantes : *Exemple typique du paradoxe de Condorcet* comme titre et *Expérimentation thèse 2* comme description. Le vote doit commencer le **27 mars 2019** et termine le **31 mars 2019**, doit se passer entre **3 Candidats (A, B, C)** auquel participeront **60 votants**. En plus comme paramètres, nous laissons l'option **peu importe**, ce qui va amener le système à chercher la procédure à recommander parmi toute la liste des procédures implémentées. On laissera aussi la méthode de choix en **automatique**, ce qui va permettre une attribution directe de la procédure de vote recommandée au vote qui est en cours de création. La figure suivante permet de constater les informations.

The screenshot shows the 'Paramètres du vote' step of the GRECO interface. At the top, there are three tabs: '1. Création de vote', '2. Paramètres du vote' (which is active), and '3. Procédure à appliquer'. Below the tabs is a blue header 'Données du vote:'. Under this header, there are two columns of input fields. The left column has 'Title:' with the value 'Exemple typique du paradoxe de Condorcet', 'Date start :' with '27 mars 2019', and a green icon representing voters with the number '60' and the text 'Voter(s)'. The right column has 'Description:' with 'Expérimentation thèse 2', 'End date:' with '31 mars 2019', and an orange icon representing candidates with the number '3' and the text 'Candidats'. Below this is a green header 'Paramètres'. Underneath, there are two dropdown menus: 'Type de procédures' set to 'Peu importe' and 'Méthode de choix' set to 'Automatique'. At the bottom right, there is a blue button labeled 'Suivant >>'. The overall interface is clean and modern with a light blue and green color scheme.

Figure 5-1:Création du vote de la situation 2 (paramètres)

En cliquant sur le bouton **suivant**, le système GRECO, en fonction du nombre de votants, du nombre de candidats et du profil du facilitateur connecté, fait ses calculs afin de lui proposer une procédure de vote. L'écran suivant montre le résultat qui sera automatiquement attribuée à cet exemple de vote.

Figure 5-2: Création du vote de la situation 2 (procédure recommandée)

On constate sur la **figure 5-2** que le système propose la procédure de **Borda** pour la détermination du vainqueur du vote de la situation 2.

Après avoir renseigné dans le système les préférences des 60 votants suivant le **tableau 5-6**, nous obtenons les résultats montrés sur la figure suivante.

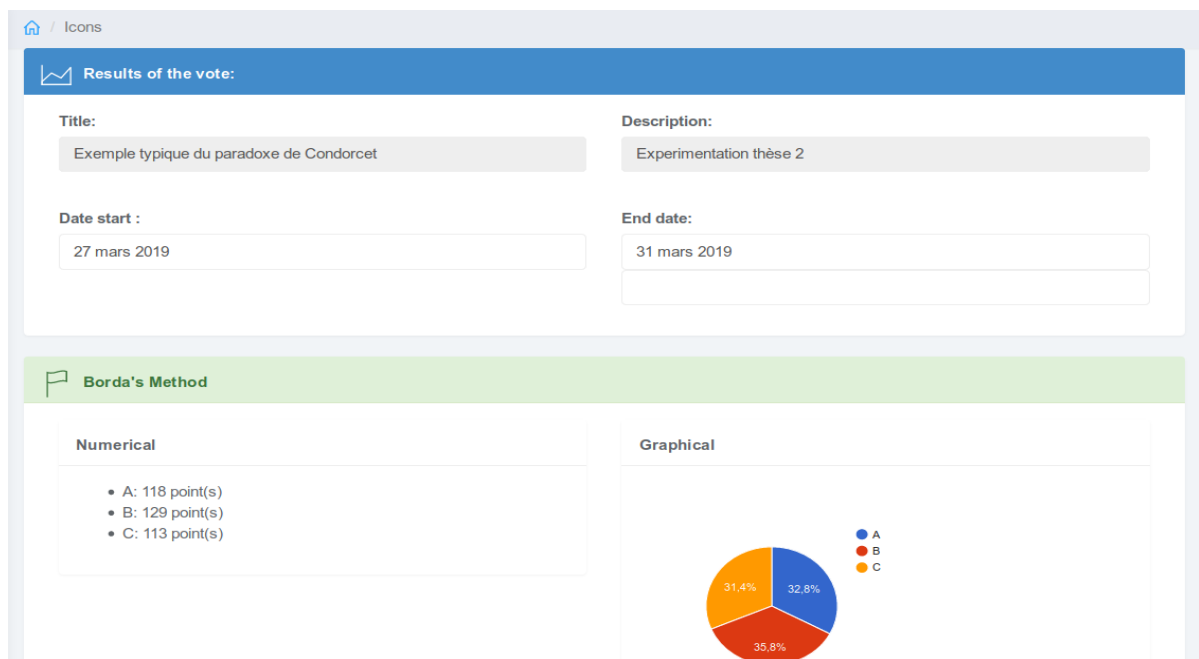


Figure 5-3: Résultat du vote de la situation 2

L'application de la procédure de Borda proposé par GRECO, permet d'élire un gagnant partir des préférences des votants, ce qui nous évite d'observer un vote sans résultat.

Situation 3

Imaginons que dans un conseil, trois décideurs doivent choisir entre trois options A, B et C. Après passage au vote, le tableau suivant montre les préférences exprimées.

Nombre de votants→		1	1	1
Rang	1 ^{ère}	A	B	C
	2 ^{ème}	B	C	A
	3 ^{ème}	C	A	B

Tableau 5-8: Préférences des votants (situation 3)

Avec un calcul manuel, il est impossible de déterminer un gagnant avec les préférences du **tableau 5-8** quelle que soit la procédure de vote appliquée. L'analyse des mêmes préférences avec GRECO donne les résultats suivants.

📄 Results of the vote:

Title: Exemple de l'indifférence

Description: Situation3 de vote Experimentation

Date start : 27 mars 2019

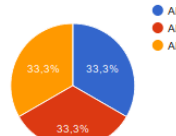
End date: 28 mars 2019

🚩 Borda's Method

Numerical

- Alt1: 6 point(s)
- Alt3: 6 point(s)
- Alt4: 6 point(s)

Graphical



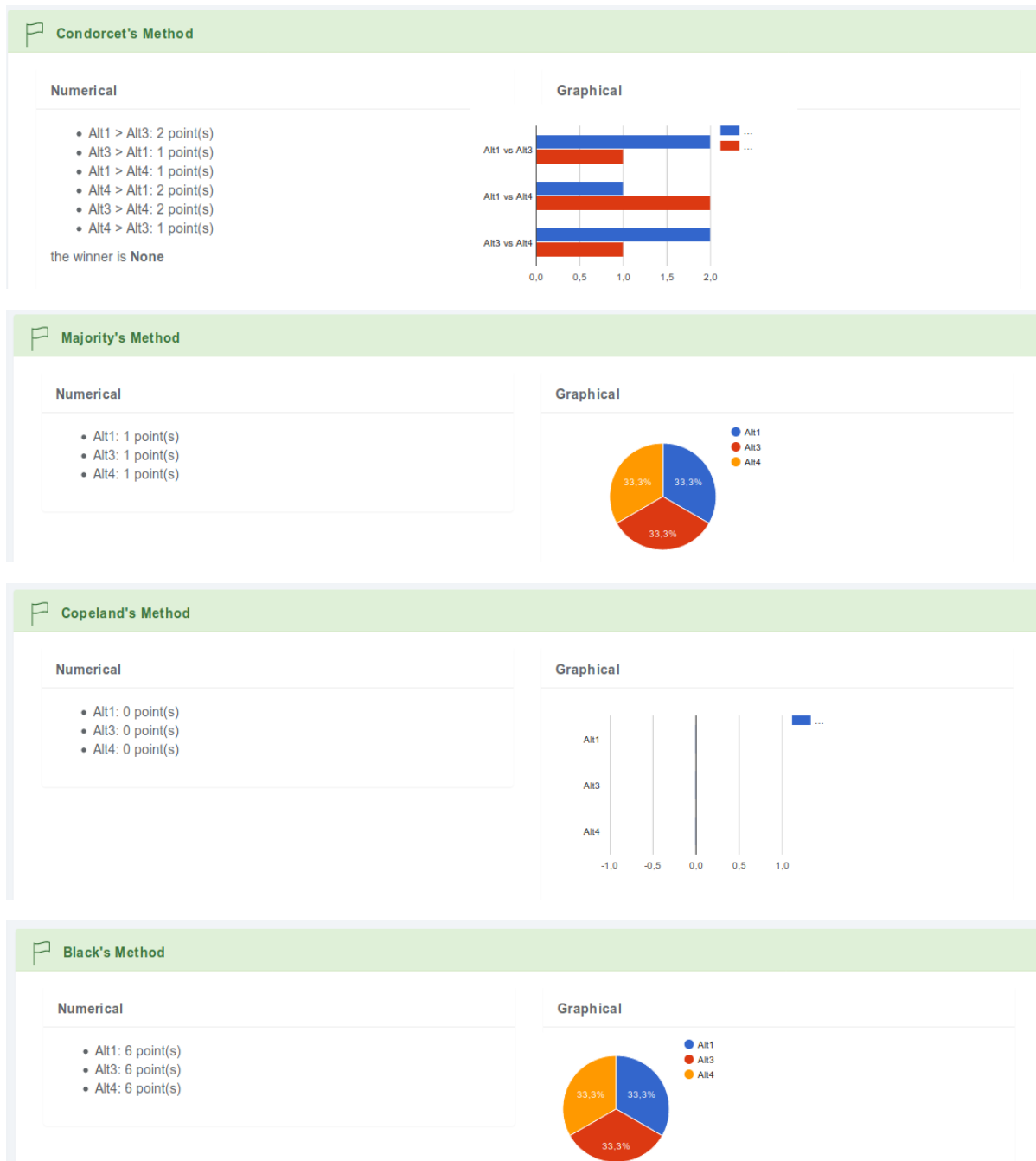


Tableau 5-9: Résultats de GRECO-situation 3

Les chiffres du tableau 5-9 montrent que GRECO donne des résultats concordants avec la théorie du vote. Pour résoudre ce type de situation, on fait recours à un agenda de vote qui met en duel deux candidats et celui qui gagne affronte le candidat restant. C'est de cette manière que l'on arrivera à stopper le cycle formé par les préférences des votants de cet exemple.

5.4.3. Mesures des précisions de la prédiction des votes des procédures de vote implémentées

Dans cette section, nous allons déterminer les métriques RMSE et MAE qui sont utilisées pour évaluer la précision des notes prédites pour les procédures de vote. Si cette précision est bonne donc le système de recommandation GRECO propose les procédures de vote adaptées.

Afin de pour pouvoir calculer ces deux mesures pour GRECO, nous allons utiliser la méthode de validation croisée *K-cross fold* (Voir le point 5.3.3.3) et le protocole suivant :

1. Créer 100 votes
2. Associer les cinq procédures implémentées aux 100 votes
3. Evaluer les résultats de ces 100 votes
4. Former un jeu de données (Data set) qui va nous permettre de calculer RMSE et MEA en utilisant le module **Surprise**² de **scikit**³

Dans les points 2 et 3 sont, l'association des procédures de vote et attribution des notes pour évaluer les résultats obtenus se font de façons aléatoires. (Voir annexe 1)

Résultats de l'expérimentation

Après exécution de la méthodologie décrite, nous obtenons les résultats illustrés sur les deux figures 5-4 et 5-5.

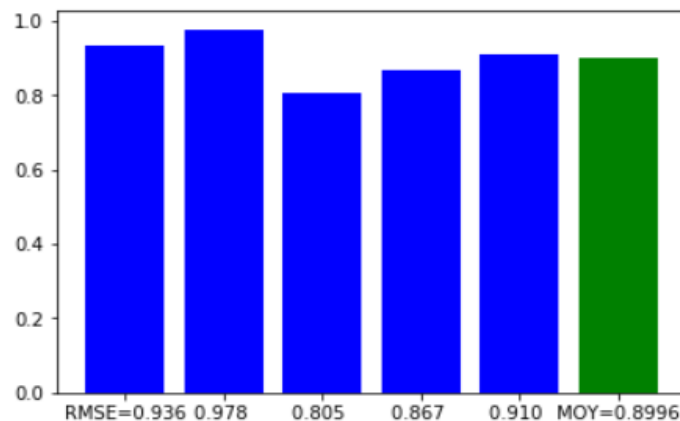


Figure 5-4: GRECO- RMSE

² <https://surprise.readthedocs.io/en/stable/index.html>

³ <https://www.scipy.org/scikits.html>

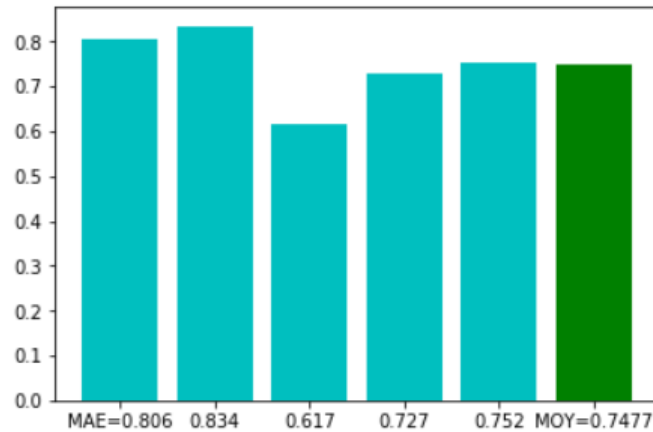


Figure 5-5: GRECO-MAE

Les valeurs moyennes du RMSE et du MAE sont respectivement **0.90** et **0.747**. Ce qui nous confirme que notre technique de notation (implicite et explicite) des procédures de vote, peut permettre GRECO, de formuler des bonnes recommandations au facilitateur en comparaison des valeurs standards du (Gantner, et al., 2019).

5.4.4. Mesure de précision d'utilisation (Facilitateur)

Dans cette experimentation nous allons déterminer la précision, le rappel et la moyenne harmonique des deux valeurs nommée F_mesure (voir le point 5.3.2). Ces valeurs vont nous permettre d'évaluer la capacité de notre système à proposer des procédures de vote qui correspondent aux attentes des utilisateurs(Facilitateur).

Afin de mener cette experimentation, nous allons suivre la même méthodologie que le point **5.4.3** car le jeu de données généré nous permet de déterminer la précision, le rappel et le F_mesure pour GRECO.

Résultats de l'expérimentation

Les résultats obtenus sont illustrés dans le tableau suivant

Précision	Rappel	F-mesure
0,95	0,526666667	0,67765237
0,55	0,4	0,463157895
0,69	0,616666667	0,65127551
0,57	0,5	0,53271028

1	0,4	0,571428571
0,752	0,488666667	0,579244925

Tableau 5-10: GRECO -Mesures de prédiction d'utilisation

La figure 5-6 donne une représentation graphique des valeurs de la précision, le rappel et la moyenne harmonique des deux valeurs.

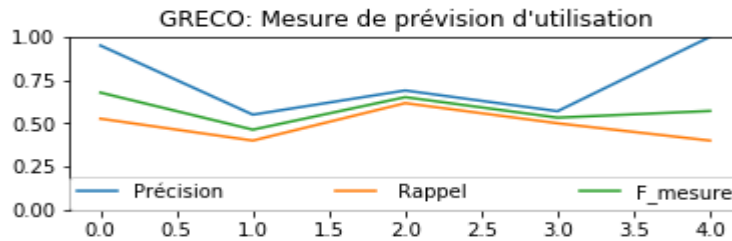


Figure 5-6: Précision -Rappel-F-mesure de GRECO

En moyenne nous obtenons **0,752; 0,488et 0,579** pour la précision, le rappel et le F_mesure. On peut conclure que les prédictions des procédures de vote faites à partir des notations des utilisateurs sont correctes malgré le peu d'évaluations disponibles dans GRECO pour l'instant.

5.5. Conclusion

Nous avons expliqué les principales méthodes de test d'un système de recommandation à savoir les expérimentations hors ligne, les expérimentations basées sur l'étude d'un échantillon d'utilisateurs et les expérimentations par évaluation en ligne. La mise en place de ces méthodes nécessite l'adaptation des mesures comme la précision de la prédiction de note, les mesures de prévision d'utilisation et des mesures d'ordre. Chaque mesure regroupe plusieurs métriques qui peuvent être utilisées pour évaluer divers aspects d'un système de recommandation. Notre système à cause de sa spécificité n'a pas pu être comparé directement aux standards du domaine de la recommandation comme *MovieLens*, *amazon*, *netflix*, etc... Par contre on a pu réaliser des expérimentations avec quelques exemples classiques de la théorie du vote pour valider nos algorithmes et les principales fonctionnalités de notre système et aussi estimer les métriques comme le RMSE⁴ et MAE⁵. En dernier point nous avons réalisé des expérimentations pour déterminer certaines mesures de prévision⁶ d'utilisation comme la précision, le rappel et la F-mesure.

⁴ <https://www.kaggle.com/acoulibaly/greco-rmse-mae-chap5>

⁵ <https://www.kaggle.com/acoulibaly/greco-rmse-mae-chap5>

⁶ <https://www.kaggle.com/acoulibaly/greco-precision-recall-these>

Conclusion générale

Au début de ces travaux de recherches, notre objectif était de mettre en place un outil d'aide à la décision pour un facilitateur. L'outil devrait être capable de recommander une procédure de vote adéquate selon le contexte de travail.

Pour atteindre notre but, nous avons étudié les différentes procédures de vote pour comprendre leur fonctionnement. Ce qui nous a conduit à classifier les procédures de vote en deux types à savoir celles par classement ou sans classement. Les procédures par classement se peuvent se diviser en deux sous-groupes : les Condorcet cohérentes et les non cohérentes. Une caractérisation nous a permis d'établir des critères d'évaluation des différentes procédures de vote. Un état de l'art sur les systèmes de recommandation nous a permis de connaître les différents types et approches utilisées dans leur mise en place.

Puis nous avons utilisé les critères d'évaluation des procédures de vote pour proposer un système de recommandation de procédures de vote qui implémente les méthodes de Borda, Condorcet, Pluralité, Black et Copeland. Un système qui utilise l'approche hybride pour proposer des recommandations. Dans notre approche hybride nous avons utilisé le filtrage basé sur le contenu de la caractérisation des procédures de vote et filtrage collaboratif autour des évaluations émises par les facilitateurs dans l'utilisation de l'outil.

Nous avons présenté les différents modèles d'expérimentation et certaines métriques nécessaires pour évaluer un système de recommandation. A cause de la particularité de notre système de recommandation nous avons mis au point des validations personnalisées. Dans un premier temps nous avons testé les algorithmes des procédures de vote implémentées avec quelques exemples classiques de la théorie du vote. Les résultats obtenus ont été concluants. Ensuite nous avons testé la capacité de recommandations avec quelques métriques.

Malgré quelques difficultés rencontrées, nous avons atteint les objectifs fixés au début de ce travail de thèse. En plus nous avons proposé un cadre de travail de recommandation de procédures de vote.

Etant donné que l'application des systèmes de recommandation dans ce genre de problématique n'est pas trop répandue, les perspectives sont nombreuses pour améliorer ce travail :

- Une première perspective concerne l'intégration d'autres procédures de vote dans le système. En effet la solution ne renferme que cinq procédures alors qu'il existe

presqu'une vingtaine. De plus les procédures intégrées ont un fonctionnement par classement.

- Actuellement le système prend en compte seulement les procédures de vote par classement, une seconde perspective serait d'ajouter la possibilité de voter en utilisant les procédures sans classement.

Une troisième perspective serait d'explorer l'apport des systèmes de recommandations basés sur l'ontologie. Cette approche combinée avec les caractérisations des procédures de vote pourra améliorer la modélisation des profils d'utilisateurs (facilitateur).

Personnellement, cette thèse m'a été bénéfique car elle m'a permis d'approfondir mes connaissances scientifiques comme chercheur dans les domaines de la prise de décision, et du choix social. J'ai aussi appris des outils de développement comme GROOVY-GRAIL, DJANGO et un peu la data science.

Publications de la thèse

Articles de revues internationales / International journal papers

- [Morteza Yazdani](#), [Pascale Zaraté](#), [Adama Coulibaly](#), [Edmundas Kazimieras Zavadskas](#). *A group decision making support system in logistics and supply chain management*. Dans / In : *Expert systems with Applications*, [Elsevier](#), Vol. 88, p. 376-392, décembre / december 2017.
URL : <http://www.sciencedirect.com/science/article/pii/S0957417417304840> - <http://oatao.univ-toulouse.fr/19118/>
[BibTeX](#)

Conférences et workshops internationaux / International conference papers

- [Adama Coulibaly](#), [Pascale Zaraté](#), [Guy Camilleri](#), [Jacqueline Konate](#), [Fana Tangara](#). *A Voting Procedures Recommender System for Decision-making (regular paper)*. Dans / In : *Group Decision and Negotiation (GDN 2019)*, Loughborough, UK, 11/06/2019-15/06/2019, Vol. 351, Danielle MORAIS, Ashley CARRERAS, Adiel TEIXEIRA DE ALMEIDA, Rudolf VETSCHERA (Eds.), [Springer](#), Lecture Notes in Business Information Processing, p. 80-91, juin / june 2019.
[BibTeX](#)
- [Adama Coulibaly](#), [Pascale Zaraté](#), [Guy Camilleri](#). *Implementing Voting Tools in GRUS (short paper)*. Dans / In : *International Conference on Decision Support Systems Technologies (ICDSST 2017)*, Namur, Blegium, 29/05/2017-31/05/2017, Isabelle Linden, Shaofeng Liu, Jason Papathanasiou, Christian Colot (Eds.), [ICDSST 2017](#), p. 67-72, mai / may 2017.
URL : <http://oatao.univ-toulouse.fr/19057/>
[BibTeX](#)

Annexe

Annexe 1 : Data set de GRECO

procedure_id	facilitateur_id	rating
2	45	3
5	32	4
1	9	3
3	45	3
1	23	5
1	9	4
1	23	3
4	23	5
4	9	5
2	2	3
1	23	4
5	32	4
5	9	5
4	23	5
4	32	4
5	45	3
5	32	3
1	32	5
4	32	5
1	32	4
5	45	4
1	23	3
2	9	5
5	9	4
5	45	3
4	23	3
4	9	4
4	2	4
3	2	5
3	23	3
5	2	5
4	2	3
5	45	4
5	9	5
5	45	4
3	32	3
4	23	3
2	2	4
5	45	5
5	32	3

5	9	3
5	32	4
1	23	4
5	23	5
4	45	4
3	32	5
2	2	5
2	9	3
2	9	4
5	23	3
4	45	5
1	45	3
4	23	5
4	45	4
1	32	4
2	23	3
1	2	3
2	9	5
1	45	5
2	32	5
2	45	5
4	23	4
5	23	3
5	45	5
4	23	5
2	23	5
1	9	4
3	32	3
1	9	4
4	23	4
4	32	4
1	2	3
1	2	3
4	32	4
1	23	3
2	45	4
4	23	4
1	9	4
2	9	3
2	32	3
4	45	3
3	9	5
4	23	5
4	45	4
5	32	5

2	32	3
3	9	5
3	23	3
2	9	5
3	45	4
3	23	5
3	32	4
4	45	3
4	2	4
5	32	4
2	2	3
4	2	5
4	2	5
5	23	5
2	23	5

Bibliographie

- ADLA, Abdelkader. 2010.** *Aide à la Facilitation pour une prise de Décision Collective : Proposition d'un Modèle et d'un Outil.* Toulouse : Université Paul Sabatier - Toulouse III, 2010.
- ADOMAVICIUS, Gediminas et TUZHILIN, Alexander. 2005.** Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering.* 2005, 6, pp. 734-749.
- ADOMAVICIUS, Gediminas, et al. 2005.** Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS).* 2005, Vol. 23, 1, pp. 103-145.
- AURAY, Nicolas. 2007.** Le modèle souverainiste des communautés en ligne : impératif participatif et désacralisation du vote. 2007, pp. 135 - 144.
- BAEZA-YATES, Ricardo, RIBEIRO, Berthier de Araújo Neto et al. 2011.** *Modern information retrieval.* [éd.] ACM Press. New York : Addison-Wesley, 2011.
- BALINSKI, Michel et LARAKI, Rida. 2012.** *Le jugement majoritaire : une nouvelle théorie du vote.* Paris : s.n., 2012.
- BALINSKI, Michel et LARAKI, Rida,. 2012.** Jugement majoritaire vs. vote majoritaire. *cahier de recherche 2012-37.* 2012.
- BAUJARD, Antoinette, GAVREL, Frédéric, IGRSHEIM, Herrade, et al. 2014.** vote par approbation, vote par note. *Revue économique.* 2014, Vol. 65, 3.
- BÉHUE, Virginie, FAVARDIN, Pierre, et LEPELLEY, Dominique. 2009.** La manipulation stratégique des règles de vote: une étude expérimentale. *Recherches Économiques de Louvain/Louvain Economic Review.* 2009, Vol. 75, 4, pp. 503-516.
- BENOUARET, Idir. 2017.** *Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels.* Université de Technologie de Compiègne. 2017. Thèse de doctorat.
- BERKHIN, Pavel. 2006.** A survey of clustering data mining techniques. *Grouping multidimensional data.* Springer, 2006, pp. 25-71.
- BOURSIN, Jean-Louis. 2004.** Les paradoxes du vote. [éd.] Odile Jacob. 2004, p. 253.
- BOUYSSOU, Denis, MARCHANT, Thierry et PERNY, Patrice. 2006.** Théorie du choix social et aide multicritère à la décision. Concepts et méthodes pour l'aide à la décision. 2006, Vol. 3, pp. 235-270.
- BRAMS, Steven J. et FISHBURN, Peter C. 2002.** Voting procedures. *Handbook of social choice and welfare.* 2002, Vol. 1, pp. 173-236.
- BRAMS, Steven J., KILGOUR, D. Marc et SANVER, M. Remzi. 2007.** A minimax procedure for electing committees. *Public Choice.* 2007, Vol. 132, 3-4, pp. 401-420.
- BURKE, Robin. 2002.** Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction.* 2002, Vol. 12, 4, pp. 331-370.
- BURKE, Robin. 2000.** Knowledge-based Recommender Systems. *Encyclopedia of library and information systems.* 2000, Vol. 69, 32, pp. 180-200.

- CANDILLIER, Laurent. 2001.** *Apprentissage automatique de profils de lecteurs*. Lille : Equipe GRAPPA, 2001.
- CHIEN, Yung-Hsin et GEORGE, Edward I. 1999.** A bayesian model for collaborative filtering. *AISTATS*. 1999.
- CONITZER, Vincent, DAVENPORT, Andrew et KALAGNANAM, Jayant. 2006.** Improved bounds for computing Kemeny rankings. *AAAI*. 2006, pp. 620-626.
- CRESPO, Rubén González, MARTÍNEZ, Oscar Sanjuán, LOVELLE, Juan Manuel Cueva, et al. 2011.** Recommendation System based on user interaction data applied to intelligent electronic books. *Computers in Human Behavior*. 2011, Vol. 27, 4, pp. 1445-149.
- DAS, Abhinandan S., DATAR, Mayur et GARG, Ashutosh. 2007.** Google news personalization: scalable online collaborative filtering. *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 271-280.
- DAUBECHIES, Ingrid. 2002.** Weighted Voting Systems. [éd.] Math Alive. Voting and Social Choice, 26 Janv 2002.
- DAVENPORT, Andrew et KALAGNANAM, Jayant. 2004.** A computational study of the Kemeny rule for preference aggregation. *AAAI*. 2004, pp. 697-702.
- DE BORDA, Jean-Charles. 1781.** Mémoire sur les élections au scrutin. 1781.
- DUCAN, Black. 1958.** The Theory of Committees and Elections. 1958.
- DURAND, François. 2015.** *Vers des modes de scrutin moins manipulables*. Paris : UNIVERSITÉ PIERRE ET MARIE CURIE, 2015. THÈSE DE DOCTORAT.
- DURAND, Sylvain. 2000.** *Sur quelques paradoxes en théorie du choix social et en décision multicritère*. Modélisation et simulation. Université Joseph-Fourier, Grenoble I. 2000. p. 157, Thèse. tel-00006743.
- EKSTRAND, Michael D., et al. 2011.** Collaborative filtering recommender systems. *Foundations and Trends® in Human-Computer Interaction*. 2011, Vol. 4, 2, pp. 81-173.
- FELFERNIG, Alexander, FRIEDRICH, Gerhard and JANNACH, Dietmar. 2006.** An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*. 2006, Vol. 11, 2, pp. 11-34.
- FELSENTHAL, Dan S. and NURMI, Hannu. 2018.** *Voting Procedures for Electing a Single Candidate - Proving Their (In)Vulnerability to Various Voting Paradoxes*. s.l. : Springer, 2018.
- FELSENTHAL, Dan S. 2010.** Review of paradoxes afflicting various voting procedures where one out of m candidates ($m \geq 2$) must be elected. 2010, pp. 1-50.
- FISHBURN, Peter C. 1974.** Simple Voting Systems and Majority Rule. *Behavioral Science*. 1974, Vol. 19, 3, pp. 166-176.
- FISHBURN, Peter C. 2015.** *The theory of social choice*. s.l. : Princeton University Press, 2015.
- FISHBURN, Peter C. et BRAMS, Steven J. 1984.** Manipulability of voting by sincere truncation of preferences. *Public Choice*. 1984, Vol. 44, 3, pp. 397-410.
- FISHBURN, Peter C. et GEHRLEIN, William V. 1976.** Borda's rule, positional voting, and Condorcet's simple majority principle. *Public Choice*. 1976, Vol. 28, 1, pp. 79-88.

- FOMBA, Soumana. 2018.** *Décision multi-critère : Un système de recommandation pour le choix de l'opérateur d'agrégation.* UT1 Capitole. 2018. Thèse de doctorat.
- FREDRICKS, Gregory A et NELSEN, Roger B. 2007.** On the relationship between Spearman's rho and Kendall's tau for pairs of continuous random variables. *Journal of statistical planning and inference.* 2007, Vol. 137, 7, pp. 2143-2150.
- Gantner, Zeno, et al. 2019.** MyMediaLite Recommender System Library. *MyMediaLite*. [En ligne] 3.11, University of Hildesheim. , 05 2019. <http://mymedialite.net/examples/datasets.html>.
- GEANAKOPOLOS, John. 2005.** Three brief proofs of Arrow's impossibility theorem. *Economic Theory.* 2005, Vol. 26, 1, pp. 211-215.
- GEHRLEIN, William V. 1997.** Condorcet's paradox and the Condorcet efficiency of voting rules. *Mathematica Japonica.* 1997, Vol. 45, pp. 173-199.
- GHAZANFAR, Mustansar Ali et PRÜGEL-BENNETT, Adam. 2014.** Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Application.* 2014, Vol. 41, 7, pp. 3261-3275.
- HARDT, Steve et LOPES, Lia C.R. 2015.** Google Votes: A Liquid Democracy Experiment on a Corporate Social Network. 2015, pp. 1-16.
- HENRIET, Dominique. 1985.** The Copeland choice function an axiomatic characterization. *Social Choice and Welfare.* 1985, Vol. 2, 1, pp. 49-63.
- HERLOCKER, Jonathan L, KONSTAN, Joseph A et BORCHERS, Al et al. 1999.** An algorithmic framework for performing collaborative filtering. *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1999, pp. 230-237.
- HERLOCKER, Jonathan L., KONSTAN, Joseph A., TERVEEN, Loren G., et al. 2004.** Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS).* 2004, Vol. 22, 1, pp. 5-53.
- HUDRY, Olivier. 2003.** Votes et paradoxes: les élections ne sont pas monotones! *Mathématiques et sciences humaines. Mathematics and social sciences.* 2003, 163.
- HYMAN, Ross. 2011.** Divisor Method Proportional Representation in Preference-Ballot Elections. *Voting matters.* 2011, Vol. 28, pp. 15-30.
- JANNACH, Dietmar, ZANKER, Markus and FELFERNIG, Alexander. 2010.** Recommender systems: an introduction. *Cambridge University.* 2010.
- JAWAHEER, Gawesh, SZOMSZOR, Martin et KOSTKOVA, Patty. 2010.** Comparison of implicit and explicit feedback from an online music recommendation service. *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems.* 2010, Vol. ACM, pp. 47-51.
- JOHNSON, Stephen C. 1967.** Hierarchical clustering schemes. *Psychometrika.* 1967, Vol. 32, 3, pp. 241-254.
- KARYPIS, George. 2001.** Evaluation of item-based top-n recommendation algorithms. [éd.] Proceedings of the tenth international conference on Information and knowledge management. 2001, pp. 247-254.

- KENDALL, Maurice G. 1938.** A new measure of rank correlation. *Biometrika*. 1938, Vol. 30, 1/2, pp. 81-93.
- KENDALL, Maurice G. 1945.** The treatment of ties in ranking problems. *Biometrika*. 1945, Vol. 33, 3, pp. 239-251.
- KLAMLER, Christian. 2004.** The Dodgson ranking and its relation to Kemeny's method and Slater's rule. *Social Choice and Welfare*. 2004, Vol. 23, 1, pp. 91-102.
- KOHAVI, Ronny, CROOK, Thomas et LONGBOTHAM, Roger. 2009.** Online experimentation at Microsoft. *Data Mining Case Studies*. 2009, Vol. 11, p. 39.
- KONCZAK, Kathrin et LANG, Jérôme. 2005.** Voting procedures with incomplete preferences. *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*. 2005.
- LAMPRIER, Sylvain, AMGHAR, Tassadit, LEVRAT, Bernard, et al. 2008.** Using an Evolving Thematic Clustering in a Text Segmentation Process. *J. UCS*. 2008, Vol. 14, 2, pp. 178-192.
- LENGAIGNE, Benoît et POSTEL, Nicolas. 2004.** Arrow et l'impossibilité : une démonstration par l'absurde. *Revue du MAUSS*. 2004, Vol. 2, 24, pp. 388-410.
- LEPELLEY, Dominique et SMAOUI, Hatem et al. 2012.** Choix collectif et procédures de vote. *Eco Flash*. 2012, pp. 1-20.
- LEPRATTI, Sergio. 2009.** *STRATEGIES ET VOTE*. 2009.
- LOPS, Pasquale, DE GEMMIS, Marco et SEMERARO, Giovanni. 2011.** Content-based recommender systems: State of the art and trends. [éd.] MA. *Recommender systems handbook*. Springer. 2011, Vol. Bpston, pp. 73-105.
- MARTIN, Arnaud. 2012.** *Évolution de profils multi-attributs, par apprentissage automatique et adaptatif dans un système de recommandation pour l'aide à la décision*. Université Toulouse III-Paul Sabatier, Université de Toulouse. 2012. Thèse de doctorat.
- MCLEAN, Iain. 1996.** EJ Nanson, social choice and electoral reform. *Australian Journal of Political Science*. 1996, Vol. 31, 3, pp. 369-386.
- NICOLAS, Jean Antoine, et al. 1785.** Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Par M. le marquis de Condorcet. *Imprimerie Royale*. 1785.
- NURMI, Hannu. 2012.** On the relevance of theoretical results to voting system choice. Electoral systems. Springer, 2012, pp. 255-274.
- NURMI, Hannu.. 2010.** Voting Systems for Social Choice. In : *Handbook of Group Decision and Negotiation*. Springer, Dordrecht. 2010, pp. 167-182.
- PALHA, Rachel Perez, ZARATE, Pascale, DE ALMEIDA, Adiel Teixeira, et al. 2017.** Choosing a Voting Procedure for the GDSS GRUS. In : *International Conference on Group Decision and Negotiation*. Springer, Cham, 2017, pp. 163-174.
- PAPAGELIS, Manos, PLEXOUSAKIS, Dimitris et KUTSURAS, Themistoklis. 2005.** Alleviating the sparsity problem of collaborative filtering using trust inferences. [éd.] Springer. 2005, pp. 224-239.
- PAZZANI, Michael J. et BILLSUS, Daniel. 2007.** Content-based recommendation systems. *The adaptive web*. Springer. 2007, pp. 325-341.

- PEIS, Eduardo, DEL CASTILLO, JM Morales et DELGADO-LÓPEZ, Juan A. 2008.** Semantic recommender systems. analysis of the state of the topic. *Hipertext. net*. 2008, Vol. 6, 2008, pp. 1-5.
- PERNY, Patrice et ZUCKER, Jean-Danie. 2001.** Preference-based search and machine learning for collaborative filtering: the 'Film-Conseil' movie recommender system. *Information, Interaction, Intelligence*. 2001, Vol. 1, 1, pp. 9-48.
- PUBLIQUE-VIE. 2018.** Les différents modes de scrutin et leurs effets. [En ligne] 30 06 2018. [Citation : 01 11 2019.] <http://www.vie-publique.fr/decouverte-institutions/institutions/approfondissements/differents-modes-scrutin-leurs-effets.html>.
- RATLIFF, Thomas C. 2002.** A comparison of Dodgson's method and the Borda count. *Economic Theory*. 2002, Vol. 20, 2, pp. 357-372.
- RESNICK, Paul et VARIAN, Hal R. 1997.** Recommender systems. *Communications of the ACM*. 1997, Vol. 40, 3, pp. 56-58.
- RICCI, Francesco, ROKACH, Lior and SHAPIRA, Bracha. 2015.** Recommender systems: introduction and challenges. *Recommender systems handbook*. springer, 2015, pp. 1-34.
- RICCI, Francesco, ROKACH, Lior et SHAPIRA, Brach. 2011.** Introduction to Recommender Systems Handbook. *Recommender systems handbook*. 2011, Vol. 1, 3, pp. 1-35.
- ROCCHIO, Joseph John. 1966.** Document Retrieval System-Optimization and. 1966.
- SAARI, Donald G. 1995.** Basic geometry of voting. *Springer Science & Business Media*. 1995.
- SAARI, Donald G. 2008.** Disposing dictators, demystifying voting paradoxes. *Mathematical and Computer Modelling*. 2008, Vol. 48, 9-10, pp. 1671-1673.
- SAARI, Donald G. 2012.** Geometry of voting. *Springer Science & Business Media*. 2012.
- SAARI, Donald G. 2006.** Which is better: the Condorcet or Borda winner? *Social Choice and Welfare*. 2006, Vol. 26, 1, p. 107.
- SALLES, Maurice, et al. 2006.** La théorie du choix social: de l'importance des mathématiques. *Center for Research in Economics and Management (CREM)*. University of Rennes 1, University of Caen and CNRS, 2006.
- SALTON, Gerard et WONG, Anita. 1978.** Generation and search of clustered files. *ACM Transactions on Database Systems (TODS)*. 1978, Vol. 3, 4, pp. 321-346.
- SALTON, Gerard, WONG, Anita, et YANG, Chung-Shu. 1975.** A vector space model for automatic indexing. *Communications of the ACM*. 1975, Vol. 18, 11, pp. 613-620.
- SCHAFER, J. Ben, FRANKOWSKI, Dan, HERLOCKER, Jon, et al. 2007.** Collaborative filtering recommender systems. *The adaptive web*. 2007, pp. 291-324.
- SHANI, Guy et GUNAWARDANA, Asela. 2011.** Evaluating recommendation systems. *Recommender systems handbook*. Springer, 2011, pp. 257-297.
- STÉPHANE, Tufféry. 2012.** Data mining et statistique décisionnelle: l'intelligence des données. Editions Technip, 2012.
- SUITT, Jorge GONZALEZ, GUYON, Axel et HENNION, Thibault. 2014.** *Vers un système de vote plus juste ?* s.l. : HAL, 2014.

TABARROK, Alexander. 2015. Paradoxes of Voting. 2015, pp. 1-36.

TEIXEIRA DE ALMEIDA, Adiel et HANNU, Nurmi. 2014. *Aiding the choice of a voting procedure*. Juin 2014. Group Decision and Negotiation 2014.

TIDEMAN, Nicolaus. 2007. Collective decisions and voting: The potential for public choice. *Public Choice*, 2007, Vol. 133, 499-501. 10.1007/s11127-007-9216-3.

VAN NEUWENHIZEN, Jill. 1992. The Borda method is most likely to respect the Condorcet principle. *Economic Theory*. 1992, Vol. 2, 1, pp. 69-83.

WANG, Pei. 1998. Why recommendation is special. [éd.] part of the 15th national conference on artificial intelligence workshop on recommender systems. 1998, pp. 111-113.

WILLETT, Peter. 1988. Recent trends in hierarchic document clustering: a critical review. *Information processing & management*. 1988, Vol. 24, 5, pp. 577-597.

XU, Rui et WUNSCH, Donald C. 2005. Survey of clustering algorithms. 2005.

YAO, Y. Y. 1995. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science*. 1995, Vol. 46, 2, pp. 133-145.

YOUNG, Peyton. 1995. Optimal voting rules. *Journal of Economic Perspectives*. 1995, Vol. 9, 1, pp. 51-64.

ZIEGLER, Cai-Nicolas, et al. 2005. Improving recommendation lists through topic diversification. 2005, pp. 22-32.