Diversification des k meilleures réponses à des requêtes par l'exemple Diversifying top-k Answers in a Query by Example Setting

Grégory Smits, Marie-Jeanne Lesot, Olivier Pivert, Marek Reformat

Diversification des k meilleures réponses à des requêtes par l'exemple

Diversifying top-k Answers in a Query by Example Setting

Grégory Smits[1] Marie-Jeanne Lesot[2] Olivier Pivert[3] Marek Reformat[4]

[1] IMT Atlantique, Lab-STICC, gregory.smits@imt-atlantique.fr

[2] Sorbonne Université, CNRS, LIP6, marie-jeanne.lesot@lip6.fr

[3] Université de Rennes, IRISA, olivier.pivert@irisa.fr

[4] University of Alberta, reformat@ualberta.ca

Résumé :

Etant donné une table T et une requête Q, les k meilleures réponses sont les k tuples de T qui satisfont au mieux Q. L'intégration d'une contrainte de diversité vise à éviter de renvoyer des tuples redondants, qui sont trop similaires les uns aux autres. Cet article propose une stratégie de diversification dans le cadre des requêtes par l'exemple, en particulier pour les approches qui traitent d'exemples représentatifs très différents les uns des autres. Il propose une nouvelle définition de diversité qui dépend de la requête, pour évaluer si le résultat reflète la diversité des exemples représentatifs fournis par l'utilisateur, afin de couvrir toutes les composantes de la requête. L'article propose une mesure numérique de cette diversité, un algorithme permettant de générer de telles k meilleures réponses diversifiées, ainsi que son intégration dans une approche de requête flexible.

Mots-clés :

Requête par l'exemple, réponses diversifiées.

Abstract:

For a given table T and a user query Q, the top-k answers are the k tuples from T that best match Q. The integration of a diversity constraint aims at avoiding returning redundant tuples, that are too similar one to another. This paper addresses the diversification question in the Query By Example setting, especially for approaches that can deal with possibly very different representative examples provided by the user. It proposes a new definition for diversity that depends on the query, in order to measure whether the result set illustrates the diversity of the representative examples provided by the user, covering all components of the query. The paper proposes a numerical measure to assess diversity in that sense, an algorithm to identify such a diversified top-k set, optimising both the query satisfaction and the diversity measure, as well as its integration into a flexible querying approach.

Keywords:

Querying by example, diversified top-k answers.

1 Introduction

L'interaction avec les systèmes de gestion de bases de données (SGBD) repose sur une algèbre de requête et, le plus souvent, un langage formel qui n'est pas naturel aux utilisateurs non informaticiens. Le paradigme des requêtes par l'exemple, ou Query By Example, QBE, introduit dans [14], facilite l'étape de formulation de la requête, exprimée par un ensemble d'exemples représentatifs des réponses attendues à partir desquels le mécanisme QBE infère une requête formelle qui peut être soumise au SBGD. Cette expression du besoin d'information signifie que différents types de réponses sont acceptables : la requête induite est de nature disjonctive.

Les réponses fournies par les SGBD à des requêtes sont le plus souvent définies comme les k tuples de la base de données (où k est un paramètre fixé par l'utilisateur) qui satisfont au mieux la requête et ne sont pas trop similaires les uns aux autres [13] : l'objectif est de fournir une vue complète des réponses d'intérêt possibles contenues dans la base de données, en les contraignant à différer les uns des autres. Un ensemble de réponses *diversifié* est classiquement défini comme maximisant la dissimilarité des réponses renvoyées prises deux à deux.

Cet article s'intéresse au problème de la diversification des résultats dans le cas des requêtes par l'exemple, en soulignant que ce cadre nécessite une définition de diversité spécifique : elle doit tenir compte de la diversité des exemples qui définissent la requête et ne peut dépendre uniquement de l'ensemble de réponses candidates. Plus précisément, il propose de définir un ensemble de tuples répondant à une requête disjonctive comme diversifié ssi il couvre tous

les exemples représentatifs fournis par l'utilisateur à partir desquels la requête est inférée, afin de tenir compte de toutes les composantes de cette requête. Les contributions de l'article sont (1) une adaptation de la notion de diversité dans le cadre QBE, (2) un algorithme permettant de construire un ensemble des k meilleurs résultats qui optimise à la fois la satisfaction de la requête et la mesure de diversité proposée, (3) une implémentation de cet algorithme dans un cadre de QBE flexible.

L'article est structuré de la façon suivante : après avoir présenté le contexte et les motivations dans la section 2, il décrit, dans la section 3, la mesure de diversité proposée, un algorithme permettant de l'optimiser ainsi qu'une intégration technique de ce dernier, l'implémentant dans la stratégie de requête par l'exemple DCQ [7], et illustre la pertinence des résultats qu'elle permet d'obtenir. La section 4 présente de premières expériences pour évaluer la qualité de l'approche proposée, la section 5 conclut l'article et discute des perspectives qu'il ouvre.

2 Contexte et motivation

Cette section positionne l'approche proposée par rapport aux systèmes de QBE existants et aux stratégies de diversification.

2.1 Le paradigme QBE

Principe général La stratégie de requête par l'exemple, introduite par Zloof [14], a pour objectif de simplifier l'interaction d'un utilisateur non expert avec un SBGD [11] : elle prend en entrée un ou plusieurs exemples de tuples fournis par l'utilisateur, ou les évaluations, par l'utilisateur, d'exemples prototypiques reflétant le contenu de la base de données. Nous nous plaçons dans le premier cas.

La table considérée, qui peut être le résultat d'une requête de jointure est notée T, son schéma $\{A_1, \ldots A_p\}$. T contient un ensemble de tuples $\{t_1, \ldots, t_n\}$ où, pour tout i, $t_i \in D_1 \times \ldots \times D_p$ et D_j est le domaine de l'attribut A_j. L'utilisateur fournit un ensemble $\mathcal{E} = \{e_1, \ldots, e_m\}$ d'exemples représentatifs, qui illustrent ce qu'il cherche. Le système de QBE calcule alors, pour chaque tuple candidat $t \in T$ un score de satisfaction noté $s_\mathcal{E}(t)$, qui quantifie à quel point t répond à la requête Q induite par $\mathcal{E}$. Comme détaillé ci-dessous, les approches existantes diffèrent par la façon de calculer le score de satisfaction. Sans perte de généralité, il peut être considéré comme une valeur numérique de $[0, 1]$.

Deux hyperparamètres sont utilisés pour contrôler l'ensemble de résultats : un entier k qui indique le nombre de résultats attendus et un seuil $\alpha \in]0, 1]$ sur le degré de satisfaction minimal souhaité : pour toute requête Q, l'ensemble de candidats est défini comme $\Sigma_Q^\alpha = \{t \in T / s_\mathcal{E}(t) \geq \alpha\}$. Enfin, $\Sigma_Q^{k,\alpha}$ représente le sous-ensemble de Σ_Q^α contenant au plus k tuples qui satisfont au mieux la requête Q, c-à-d de degrés de satisfaction maximaux. Il est possible que $|\Sigma_Q^{k,\alpha}| < k$ s'il y a moins de k réponses qui satisfont Q avec un score d'au moins α. Les principes précédents s'appliquent de façon générale aux SGBD. Dans le cas QBE, les notations sont légèrement modifiées, en remplaçant Q par $\mathcal{E}$: les ensembles de résultats sont notés $\Sigma_\mathcal{E}^\alpha$ et $\Sigma_\mathcal{E}^{k,\alpha}$ respectivement.

Principales approches existantes Trois familles de méthodes QBE peuvent être distinguées. La première n'infère pas explicitement de requête formelle et considère que les exemples fournis par l'utilisateur sont indépendants les uns des autres : elle identifie les tuples de la base qui sont similaires à au moins l'un des exemples (par rapport à tous les attributs) et, s'ils sont fournis, dissimilaires d'au moins l'un des contre-exemples (par rapport à au moins l'un des attributs). L'approche proposée par De Calmès et al. [2] repose sur un système de raisonnement à partir de cas pour identifier les réponses candidates. Elle définit le score de satisfaction $s_\mathcal{E}$ comme une combinaison de la si-

milarité aux exemples positifs et la dissimilarité aux contre-exemples. Zadrozny et al. [12] proposent une approche par k-plus proches voisins pour identifier les tuples qui sont proches des réponses attendues fournies par l'utilisateur.

L'approche Disjunctive Concept Querying, DCQ [7], n'infère pas non plus de requête formelle, mais exploite des dépendances entre exemples fournis, pour apprendre une mesure de similarité appropriée qui extrait les corrélations entre attributs : DCQ repose sur l'intégrale de Choquet pour calculer un score de satisfaction $s_{\mathcal{E}}$ qui permet d'interpréter les exemples fournis comme différents types de résultats attendus. Ainsi, comme détaillé dans la section 3.3, DCQ permet d'identifier des sous-ensembles d'exemples représentatifs en soulignant l'importance de combinaisons fréquentes de valeurs d'attributs, sans éliminer les exemples plus exceptionnels qui ne ressemblent pas aux autres membres de $\mathcal{E}$.

Une troisième famille d'approches construit une requête formelle à partir des exemples et contre-exemples fournis : les exemples positifs peuvent par exemple être analysés globalement pour identifier les prédicats flous les plus représentatifs (c-à-d les plus fréquents), en tenant compte de la contrainte que ces prédicats ne couvrent pas également de réponses non-souhaitées [5]. Une autre approche [12] définit la condition de recherche induite comme la composition de termes flous tirés d'un vocabulaire pré-défini qui discrétise les domaines de chaque attribut. Cette approche offre à l'utilisateur une description linguistique des valeurs partagées par les exemples positifs qui ne sont pas partagées par les contre-exemples.

2.2 Diversification des résultats

La combinaison de la notion de satisfaction avec celle de diversité est une question qui a été beaucoup abordée, initialement dans le domaine des systèmes de recommandation [10], voir [1] pour une synthèse récente. Elle est maintenant mise en œuvre dans de nombreuses applications, par exemple la génération d'explications [6]. Dans le cas des réponses aux requêtes des bases de données [13], l'objectif est d'éviter que de très nombreux tuples similaires satisfaisant la requête fournie conduisent à un ensemble de k meilleurs résultats contenant un seul type de réponse.

La diversité est le plus souvent définie et mesurée comme le résultat d'une comparaison deux à deux des tuples de l'ensemble considéré : en notant Σ ce dernier et $dist$ une mesure de distance, elle est définie comme

$$div(\Sigma) = \sum_{t,t' \in \Sigma} dist(t, t') \qquad (1)$$

Etant donné un ensemble de réponses candidates Σ_Q^α, c-à-d des tuples associés à un degré de satisfaction suffisant, un mécanisme de diversification a pour but d'identifier un sous-ensemble de Σ_Q^α, noté $\tilde{\Sigma}_Q^{k,\alpha}$, qui contient k réponses qui maximisent la diversité :

$$\tilde{\Sigma}_Q^{k,\alpha} = \arg \max_{\Sigma \subseteq \Sigma_Q^\alpha, \text{ t.q.} |\Sigma|=k} div(\Sigma) \qquad (2)$$

L'optimisation de cette fonction de coût est un problème NP-complet [3], certaines variantes heuristiques appliquent un algorithme de clustering à l'ensemble Σ_Q^α afin d'identifier sa structure en sous-groupes de réponses similaires [8]. Le résultat diversifié est alors composé des tuples les plus représentatifs de chacun de ces clusters. L'étape de clustering augmente la complexité du système de requête ainsi que le temps de calcul. Il peut aussi arriver qu'elle conduise à une partition non pertinente si l'ensemble initial est trop petit.

3 Diversification proposée

Cette section décrit la stratégie proposée pour diversifier un ensemble de résultats dans le cadre de requête par l'exemple : elle présente successivement la mesure de diversité proposée, un algorithme permettant d'identifier un ensemble optimal, en termes de score de satisfaction et de diversité, ainsi qu'une implémentation de ce dernier.

3.1 Mesure de diversité proposée

Comme rappelé dans la section précédente, la définition classique de diversité ne dépend que de l'ensemble résultat et s'exprime en termes de dissimilarité deux à deux des tuples qu'il contient. La définition que nous proposons dépend de plus de l'ensemble requête de réponses attendues $\mathcal{E}$: un ensemble Σ de réponses candidates est dit diversifié par rapport à $\mathcal{E}$ s'il couvre chaque exemple de $\mathcal{E}$, c'est-à-dire si chaque exemple e est associé, en terme de similarité minimale, à au moins l'une des réponses candidates. Ainsi Σ est diversifié s'il reflète la diversité de $\mathcal{E}$. Formellement, en notant sim une mesure de similarité appropriée (cf par exemple [4]) et η un seuil, Σ *est diversifié par rapport à* $\mathcal{E}$ ssi :

$$\forall e \in \mathcal{E}, \exists t \in \Sigma \text{ tq } sim(e,t) \geq \eta, \qquad (3)$$

Plus précisément, la mesure de diversité proposée évalue dans quelle mesure la couverture des exemples représentatifs de $\mathcal{E}$ est équitable : chaque exemple e doit être couvert par le même nombre, soit $\lfloor \frac{k}{|\mathcal{E}|} \rfloor$, de tuples de Σ qui en soient suffisamment proches. En notant $S^e_\Sigma = \{t \in \Sigma \,/\, sim(t,e) \geq \eta\}$ l'ensemble des tuples de Σ suffisamment proches de e, nous proposons le critère de manque de diversité de Σ par rapport à $\mathcal{E}$ suivant, à minimiser

$$mDiv(\Sigma, \mathcal{E}) = \frac{1}{\lfloor \frac{k}{|\mathcal{E}|} \rfloor} \sqrt{\frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \left(|S^e_\Sigma| - \left\lfloor \frac{k}{|\mathcal{E}|} \right\rfloor \right)^2}. \qquad (4)$$

On peut noter que dans cette définition, une réponse candidate t peut couvrir plusieurs exemples représentatifs simultanément. En effet, elle peut appartenir à plusieurs S^e_Σ si elle est suffisamment proche de plusieurs e.

L'objectif est alors d'identifier, à partir d'un ensemble de candidats $\Sigma^\alpha_\mathcal{E}$, le sous-ensemble $\tilde{\Sigma}^{k,\alpha}_\mathcal{E}$ de cardinal k qui maximise la diversité. Sa définition est identique à celle donnée dans l'équation 2, instanciée avec l'opposé de la mesure de manque de diversité proposée. En fonction de $\mathcal{E}$, la table considérée T et les valeurs des paramètres (k, α, η), il est bien sûr possible qu'un tel sous-ensemble n'existe pas.

3.2 Algorithme de diversification proposé

L'algorithme 1 donne la description en pseudo-code de l'approche proposée pour calculer l'ensemble $\tilde{\Sigma}^{k,\alpha}_\mathcal{E}$ qui maximise la diversité par rapport à $\mathcal{E}$. Il prend en entrée l'ensemble $\Sigma^\alpha_\mathcal{E}$ de tuples qui ont un degré de satisfaction suffisant par rapport à la requête $\mathcal{E}$, calculé dans une étape préliminaire.

Une liste vide l_e est d'abord initialisée pour chaque élément $e \in \mathcal{E}$. Ensuite, $\Sigma^\alpha_\mathcal{E}$ est parcouru dans l'ordre décroissant des scores $s_\mathcal{E}(t)$ et chaque candidat t est inséré à la fin des listes l_e telles que $sim(t,e) \geq \eta$. Enfin, les premiers éléments de chaque liste sont ajoutés à l'ensemble final, puis les seconds éléments, et ainsi de suite, jusqu'à ce que $\tilde{\Sigma}^{k,\alpha}_\mathcal{E}$ contienne k éléments ou que $\Sigma^\alpha_\mathcal{E}$ ait été intégralement parcouru.

Comme dans le cas classique, il est possible que l'ensemble résultant final $\tilde{\Sigma}^{k,\alpha}_\mathcal{E}$ ne contienne pas le nombre souhaité de réponses, k. C'est d'abord évidemment le cas si l'ensemble de réponses candidates initial $\tilde{\Sigma}^\alpha_\mathcal{E}$ est de cardinal insuffisant, c-à-d si l'étape préliminaire n'a pas identifié au moins k tuples qui satisfont suffisamment la requête. La seconde raison vient de la contrainte imposée par le test sur la taille de $\tilde{\Sigma}^\alpha_\mathcal{E}$ à la ligne 15 : elle a pour objectif de garantir que l'ordre dans lequel les éléments de $\mathcal{E}$ sont traités n'a pas d'effet sur le résultat. Elle garantit en effet qu'à chaque tour de boucle, toutes les listes non vides représentant les réponses différentes attendues (les l_e) soient traitées, ou aucune. Aussi, $|\tilde{\Sigma}^{k,\alpha}_\mathcal{E}| \leq \min(k, max_e(|l_e|))$.

L'utilisation de cet algorithme pour diversifier un ensemble de résultats n'ajoute pas de coût de calcul significatif : le tri des tuples de $\Sigma^\alpha_\mathcal{E}$ en ordre décroissant de leur score $s_\mathcal{E}(t)$ est effectué en $\mathcal{O}(|\Sigma^\alpha_\mathcal{E}| \log_2(|\Sigma^\alpha_\mathcal{E}|))$. L'affectation de ces réponses candidates aux différentes listes

Input: Requête $\mathcal{E}$; réponses candidates $\Sigma_{\mathcal{E}}^{\alpha}$; mesure de similarité sim ; seuil η ; nombre de réponses souhaitées k
Output: Réponses diversifiées $\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha}$

$\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha} \leftarrow \emptyset$
$l_e \leftarrow []$ pour chaque $e \in \mathcal{E}$
$maxle \leftarrow 0$
$sort(\Sigma_{\mathcal{E}}^{\alpha}, s_{\mathcal{E}})$; ▷ trier $\Sigma_{\mathcal{E}}^{\alpha}$ par valeurs
▷ décroissantes de $s_{\mathcal{E}}(t)$
foreach $t \in \Sigma_{\mathcal{E}}^{\alpha}$ **do**
  foreach $e \in \mathcal{E}$ **do**
    if $sim(t, e) \geq \eta$ **then**
      $l_e.append(t)$
      $maxle \leftarrow \max(maxle, |l_e|)$
    end
  end
end
$i \leftarrow 0$
while $|\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha}| + |\mathcal{E}| \leq k$ *and* $i < maxle$ **do**
  foreach $e \in \mathcal{E}$ **do**
    if $i < |l_e|$ **then**
      $\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha}.add(l_e[i])$
    end
  end
  $i \leftarrow i + 1$
end
return $\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha}$

Algorithm 1: Diversification de $\Sigma_{\mathcal{E}}^{\alpha}$

se fait en temps linéaire, elle est bornée par le nombre de tuples à diversifier, soit $|\Sigma_{\mathcal{E}}^{\alpha}|$.

La correction et la complétude de l'algorithme sont formulées dans la proposition suivante :

Proposition 1 *L'algorithme 1 renvoie les k résultats les plus diversifiés par rapport à l'ensemble d'exemples représentatifs $\mathcal{E}$ d'après l'équation 4 du critère de manque de diversité.*

Le principe de la preuve est le suivant : à chaque itération jusqu'à ce que la plus courte des listes l_e ait été parcourue, le nombre de réponses couvrant chaque exemple représentatif de $\mathcal{E}$ est augmenté de 1, garantissant une équi-couverture tant que cela est possible par rapport au contenu de la base de données elle-même.

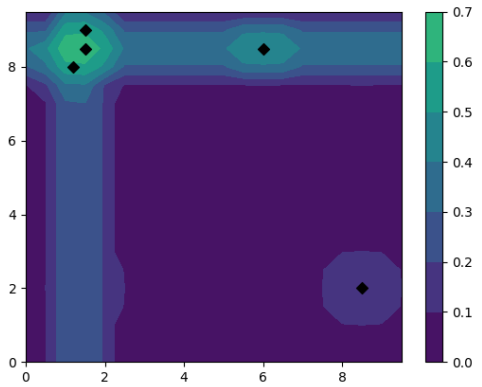

FIGURE 1 – Illustration de DCQ en 2D : $\mathcal{E}$ contient les 5 exemples représentés par les losanges. Les couleurs montrent les degrés de satisfaction calculés par CHOCOLATE.

$\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha}$ est ensuite complété selon le même principe pour les exemples e associés à d'autres réponses, conduisant à une diversité maximale relativement à la base considérée. Le fait que les réponses candidates soient traitées par ordre décroissant de $s_{\mathcal{E}}(t)$ garantit que $\tilde{\Sigma}_{\mathcal{E}}^{k,\alpha}$ contient les meilleurs tuples par rapport à requête.

3.3 Implémentation : *Div*-DCQ

Div-DCQ propose une implémentation de l'algorithme précédent pour la stratégie de reqête par l'exemple DCQ [7], dont une implémentation en PostgreSQL est disponible. DCQ permet de traiter des requêtes correspondant à des concepts disjonctifs. La diversification est alors particulièrement pertinente car DCQ garantit que tous les exemples représentatifs sont pris en compte pendant le calcul du score de satisfaction des réponses candidates. Il est donc important de garantir leur couverture dans la phase de diversification.

La première étape de DCQ, illustrée sur la figure 1, infère la fonction de satisfaction $s_{\mathcal{E}}$ à partir de $\mathcal{E}$ en appliquant la méthode CHOCOLATE [9]. $s_{\mathcal{E}}$ peut être interprétée comme la fonction d'appartenance au concept dis-

jonctif flou dont les points de $\mathcal{E}$ constituent des exemples. Deux propriétés d'importance sont à souligner (voir [9]) : d'abord CHOCOLATE permet, comme attendu, de généraliser les exemples de réponses attendues fournis par l'utilisateur. Ainsi, sur la figure 1, le fait que trois réponses attendues se situent dans la même région, autour du point (1.5, 8.5), donne en effet plus d'importance à cette région. Toutefois, contrairement à une agrégation par la moyenne par exemple, la fonction d'appartenance induite n'ignore pas les deux exemples atypiques, ce qui constitue la seconde propriété d'intérêt. Néanmoins, elle donne plus d'importance au point (6, 8.5), dont l'ordonnée est partagée avec d'autres exemples.

Techniquement, la procédure *infer_concept* ci-dessous est utilisée pour inférer la fonction de satisfaction, nommée ici *myQBE*, qui peut être appliquée à la table *testData*, qui peut être une vue d'un résultat d'une jointure plus complexe :

```
CALL infer_concept('testData','myQBE',
{ "x"=>1.5,"y"=>8.5,"x"=>1.2,"y"=>8,
"x"=>1.5,"y"=>9,"x"=>6,"y"=>8.5,
"x"=>8.5,"y"=>2});
```

La fonction *myQBE* peut ensuite être intégrée dans la clause de sélection d'une requête. L'exemple ci-dessous collecte les 200 meilleurs tuples ($k = 200$) de *testData* qui satisfont *myQBE* avec un degré d'au moins 0.2 ($\alpha = 0.2$).

```
SELECT *, get_mu() as mu FROM testData
WHERE myQBE() >= 0.2 LIMIT 200;
```

Pour l'exemple représenté sur la figure 1 et une table *testData* contenant, à titre d'illustration, 2000 tuples générés selon des distributions normales autour des 5 exemples illustratifs considérés, la partie haute de la figure 2 montre les 200 meilleurs résultats de la requête ci-dessus. Comme la zone autour du point (1.5, 8.5) obtient les meilleurs scores, les 200 meilleurs points sont tous situés dans cette région exclusivement : ils manquent de diversité et de représentativité par rapport aux différentes réponses attendues par l'utilisateur.

L'approche de diversification proposée peut être activée en ajoutant simplement le mot clé *DIVERSIFY* à la clause de sélection, indiquant

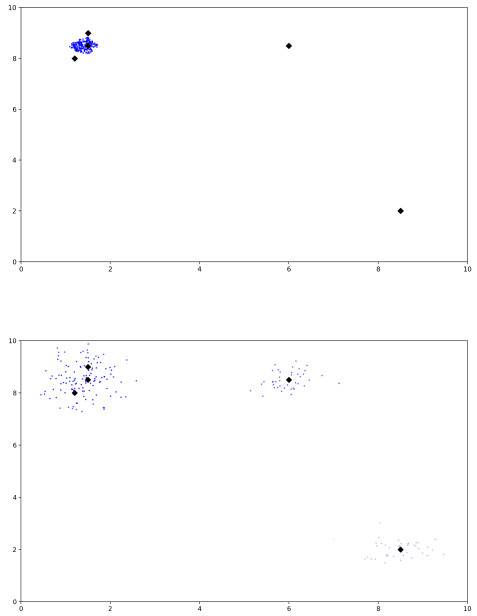

FIGURE 2 – Résultats de la requête de la fig. 1 : (haut) 200 meilleurs, (bas) avec diversification.

qu'une étape de diversification doit être appliquée *a posteriori* à l'ensemble de réponses candidates :

```
SELECT DIVERSIFY *, get_mu() as mu
FROM testData
WHERE myQBE() >  0.2 LIMIT 200 ;
```

Le graphe en bas de la figure 2 montre les résultats obtenus, qui couvrent alors tous les exemples représentatifs des réponses attendus. Ils tiennent compte de leurs propres redondances et sont, comme souhaité, plus denses autour des trois exemples fournis proches.

4 Expérimentations

Cette section présente les expérimentations réalisées pour étudier le coût de calcul et les résultats renvoyés par *Div*-DCQ, sur des données artificielles en dimension 4, générées par un mélange de 12 gaussiennes elliptiques définies sur le même domaine $[0, 10]$.

Temps de calcul La figure 3 montre le temps de calcul en fonction de la taille de la base

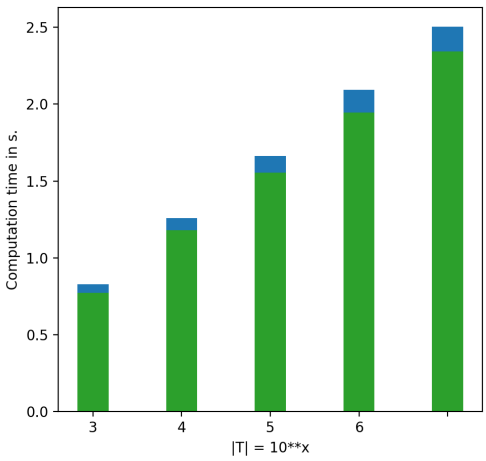

FIGURE 3 – Temps de calcul moyen sur 20 requêtes en fonction de la taille de la base de données. En vert, étape de recherche des réponses candidates, en bleu, étape de diversification.

de données, de 10^3 à 10^7. Pour chaque valeur, 20 requêtes sont exécutées avec $k = 50$. Elle confirme que la plus grande partie du temps de calcul est consacré à l'identification de $\Sigma^{\alpha}_{\mathcal{E}}$, l'étape de diversification représente en moyenne 9% du temps total. Dans ces expériences, un parcours séquentiel de T est effectué, l'utilisation d'indices pourrait accélérer l'étape de recherche, mais de telles optimisations sont à la charge du SGBD.

Compromis satisfaction/diversité La figure 4 compare la moyenne, sur 20 requêtes, du degré moyen de satisfaction $s_{\mathcal{E}}$ (graphe du haut) ainsi que le score de manque de diversité $mDiv$ (graphe du bas), pour les k meilleures réponses, obtenues avec et sans diversité

Elle montre que, pour une diminution faible de satisfaction, *Div*-DCQ permet une amélioration significative de la diversité, en particulier pour des valeurs faibles de k. Pour celles-ci en effet, les réponses sans diversification conduisent souvent à la situation représentée sur le graphe du haut de la figure 2, dans laquelle les tuples résultats sont uniquement situés autour des

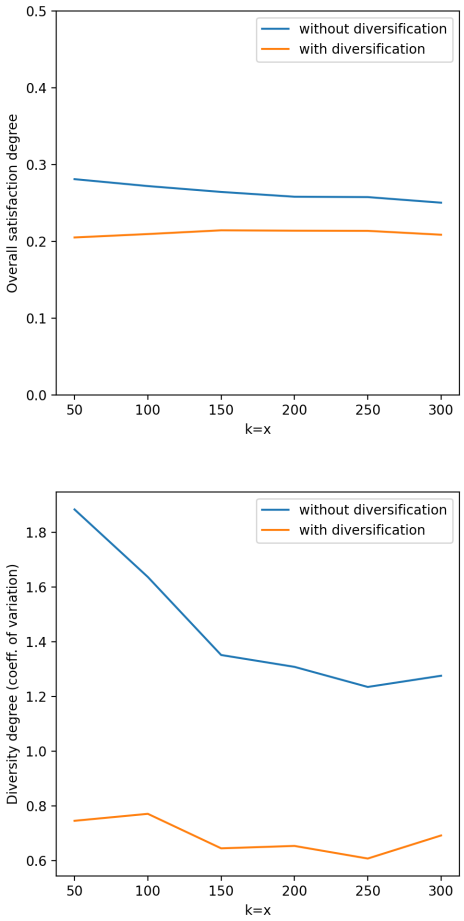

FIGURE 4 – Score de satisfaction (haut, à maximiser) et de manque de diversité (bas, à minimiser) de l'ensemble des résultats, avec et sans diversification, pour différentes valeurs de k.

exemples représentatifs majoritaires, en ignorant les autres.

5 Conclusion et perspectives

Dans le contexte des requêtes par l'exemple, l'article a proposé une définition de diversité de l'ensemble des tuples renvoyés, adaptée aux particularités de ce type de requête, mesurant sa représentativité de la variété des composantes de la requête. Il a également proposé un algorithme permettant de diversifier a posteriori un ensemble de tuples et montré que son coût de calcul est négligeable par rapport à l'exécution de la requête elle-même. Les

premières expérimentations réalisées montrent qu'il permet une augmentation significative de la diversité pour une diminution faible de la satisfaction moyenne : la stratégie proposée permet une meilleure représentativité des exemples de réponses attendues définissant la requête fournie par l'utilisateur dans le résultat final.

Les travaux en cours visent à approfondir l'étude de l'approche proposée, notamment par rapport aux paramètres comme le nombre d'exemples représentatifs fournis ou le score de satisfaction minimal. Une autre perspective a pour but de trouver une stratégie, ou une heuristique, pour éviter l'étape préliminaire d'identification et d'ordonnancement de tous les candidats avant l'étape de diversification.

Références

[1] P. Castells, N. Hurley, and S. Vargas. Novelty and diversity in recommender systems. In *Recommender systems handbook*, pages 603–646. Springer, 2021.

[2] M. De Calmès, D. Dubois, E. Hullermeier, H. Prade, and F. Sedes. Flexibility and fuzzy case-based evaluation in querying : An illustration in an experimental setting. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(01) :43–66, sep 2003.

[3] L. Ingmar, M. Garcia de la Banda, P. Struckey, and G. Tack. Modelling diversity of solutions. In *Proc. of AAAI*, 2020.

[4] M.-J. Lesot, M. Rifqi, and H. Benhadda. Similarity measures for binary and numerical data : a survey. *Int. Journal of Knowledge Engineering and Soft Data Paradigms*, 1(1) :63–84, 2009.

[5] A. Moreau, O. Pivert, and G. Smits. Fuzzy query by example. In *Proc. of ACM Symposium on Applied Computing, SAC'18*, pages 688–695, 2018.

[6] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proc. of the Conf. on Fairness, Accountability and Transparency, FAccT'20*, pages 607–617, 2020.

[7] G. Smits, M.-J. Lesot, O. Pivert, and R. R. Yager. Flexible querying using disjunctive concepts. In *Proc. of the 14th Int. Conf. on Flexible Query Answering Systems, FQAS21*, pages 29–40. Springer, 2021.

[8] G. Smits and O. Pivert. Linguistic and graphical explanation of a cluster-based data structure. In *Proc. of the 9th Int. Conf. on Scalable Uncertainty Management, SUM15*, pages 186–200. Springer, 2015.

[9] G. Smits, R. Yager, M.-J. Lesot, and O. Pivert. Concept membership modeling using a choquet integral. In *Proc. of the Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'20*, pages 359–372, 2020.

[10] B. Smyth and P. McClave. Similarity vs. diversity. In *4th Int. Conf. on Case-Based Reasoning, ICCBR01*, pages 347–361. Springer, 2001.

[11] J. C. Thomas and J. D. Gould. A psychological study of query by example. In *Proceedings of the May 19-22, 1975, national computer conference and exposition*, pages 439–445, 1975.

[12] S. Zadrozny, J. Kacprzyk, and M. Wysocki. On a novice-user-focused approach to flexible querying : the case of initially unavailable explicit user preferences. *Proc. of the 10th Int. Conf. on Intelligent Systems Design and Applications, ISDA10*, pages 696–701, 2010.

[13] K. Zheng, H. Wang, Z. Qi, J. Li, and H. Gao. A survey of query result diversification. *Knowledge and Information Systems*, 51 :1–36, 2017.

[14] M. M. Zloof. Query-by-example : A data base language. *IBM Syst. J.*, 16(4) :324–343, 1977.