Pós-Graduação em Ciência da Computação

Tarcísio Daniel Pontes Lucas

Mineração de Subgrupos em Bases de Dados de Alta Dimensionalidade

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2019

Tarcísio Daniel Pontes Lucas

Mineração de Subgrupos em Bases de Dados de Alta Dimensionalidade

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computational
Orientadora: Teresa Bernarda Ludermir
Coorientador: Renato Vimieiro

Recife
2019

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

L933m Lucas, Tarcísio Daniel Pontes
Mineração de subgrupos em bases de dados de alta dimensionalidade / Tarcísio Daniel Pontes Lucas. – 2019.
90 f.: il., fig., tab.

Orientadora: Teresa Bernarda Ludermir.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2019.
Inclui referências e apêndice.

1. Inteligência computacional. 2. Mineração de subgrupos. 3. Computação evolucionária. I. Ludermir, Teresa Bernarda (orientadora). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2019-096

Tarcísio Daniel Pontes Lucas

Mineração de Subgrupos em Bases de Dados de Alta Dimensionalidade

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação

Aprovado em: 13/03/2019.

____
Orientadora: Profa. Dra. Teresa Bernarda Ludermir

BANCA EXAMINADORA

____
Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

____
Prof. Dr. Ricardo Bastos Cavalcante Prudêncio
Centro de Informática / UFPE

____
Prof. Dra. Gisele Lobo Pappa
Departamento de Ciência da Computação/ UFMG

____
Prof. Dr. Leandro Minku
Departamento de Informática/ Universidade de Birmingham

____
Prof. Dr. Marcilio Carlos Pereira de Souto
Laboratório de Informática Fundamental/ Universidade de Orleans

Dedico esta tese aos meus pais (Lucas e Terezinha), aos meus irmãos (Rodrigo, Felipe e Andreza), à minha esposa (Marianna), aos meus filhos (Lucas e Luiza) e aos meus sobrinhos (Júlia, Leonado, Sofia e Lara).

AGRADECIMENTOS

Esta tese representa a realização de um sonho. Antes de tudo, agradeço a Deus pelo dom da vida e por tudo que nela está. Agradeço à minha orientadora, Teresa Bernarda Ludermir, pela confiança e pelo tempo gasto nas minhas orientações. Agradeço ao meu coorientador Renato Vimieiro por ter me ajudado a voltar ao jogo e pelas orientações durante esse processo. Finalmente, agradeço à minha família por todo apoio dado durante esse período.

RESUMO

Esta tese tem o objetivo de propor soluções para a mineração de subgrupos no contexto de bases de dados de alta dimensionalidade. A mineração de subgrupos (do inglês *subgroup discovery*) representa uma poderosa ferramenta para análise exploratória de dados, uma vez que apresenta informações normalmente não detectadas pela estatística tradicional. O objetivo da mineração de subgrupos é identificar conjuntos de características que discriminem um grupo alvo dos demais (ex. tratamentos médicos de sucesso dos fracassados). Existem diversas heurísticas para mineração de subgrupos, mas nenhuma delas com foco em bases de alta dimensionalidade. Isso representa uma importante lacuna na área, uma vez que se torna mais natural a necessidade de se extrair informações de conjuntos de dados de alta dimensionalidade. Nas áreas de *bioinformática* e *classificação de documentos*, por exemplo, é comum a extração de conhecimento a partir de bases com número de atributos na ordem de 10^4. É comum também nos algoritmos de mineração de subgrupos o uso de muitos parâmetros de ajuste não trivial. Isso dificulta o uso de tais técnicas, principalmente por usuários de áreas não relacionadas à *mineração de dados*. Nesse contexto, nós propomos a primeira heurística para mineração de subgrupos com foco em bases de dados de alta dimensionalidade que utiliza apenas dois parâmetros. Outro problema da área é assegurar que os subgrupos retornados não sejam redundantes entre si e que representem de forma ampla os dados do alvo da investigação. No entanto, subgrupos considerados redundantes podem representar soluções mais fáceis de serem aplicadas num problema. Assim, nós propomos uma forma inovadora de controlar a redundância, minimizando o risco do descarte prematuro de subgrupos relevantes e gerando mais informações para o usuário. Por fim, nós desenvolvemos um modelo baseado em mineração de subgrupos para o problema de descrição do perfil de comunidades (do inglês *group profiling*), que consiste no processo de construção de perfis descritivos para comunidades em redes sociais. A proposta teve como principais diferenciais gerar descrições multivariadas e com alta cobertura global.

Palavras-chave: Mineração de subgrupos. Computação evolucionária. Descoberta de conhecimento. Bases de dados de alta dimensionalidade.

ABSTRACT

This doctoral aims to propose solutions for *subgroup discovery* problems focusing on high dimensional data sets. *Subgroup discovery* represents a powerful tool for exploratory data analysis as it presents information normally not detected by traditional statistical methods. The purpose of *subgroup discovery* is to identify sets of characteristics that discriminate one target group from the other (e.g. successful medical treatments of failures). There are several heuristics for *subgroup discovery*, but none of them focuses on high dimensional data sets. This represents an important gap in the area as it becomes more natural to extract information from high dimensional data sets. In the *bioinformatics* and *document classification* realms, for example, it is common to have knowledge extraction from data sets with number of attributes on the order of 10^4. The use many non-trivial adjustment parameters is also common in *subgroup discovery* algorithms. In this context, we propose the first heuristic for subgroup mining focusing on high dimensional data sets that use only two parameters. Another problem in this area is to ensure that the returned subgroups are not redundant with each other and that they represent broadly the data of the research. However, subgroups considered redundant may represent easier solutions to a problem. Thus, we propose an innovative way of controlling redundancy, minimizing the risk of premature discarding of relevant subgroups and generating more information for the user. Finally, we have developed a subgroup mining model for the group profiling problem, which is the process of constructing descriptive profiles for communities in social networks. The distinct aspect of the research was the proposal to generate multivariate descriptions with high global coverage.

Keywords: Subgroup discovery. Evolutionary computing. Knowledge discovery. High dimensional data sets.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Minerar conhecimento a partir de base de dados de alta dimensionalidade é um desafio cada vez mais comum para empresas, governos e pesquisadores. Na área de *bioinformática*, por exemplo, diversas bases são geradas com o objetivo de encontrar relações entre expressões gênicas e doenças como o câncer (Gravier, Eleonore *et al.*, 2010; Nakayama *et al.*, 2007; Chin *et al.*, 2006). No entanto, tais bases possuem comumente número de atributos na ordem de 10^4 e número de exemplos na ordem de 10^2. Já na área de *classificação de documentos*, não é raro o uso de bases de dados com número de atributos e exemplos na ordem de 10^4 (Madani *et al.*, 2013; Kotzias *et al.*, 2015). Por fim, pesquisas e questionários realizados por governos também levam frequentemente à necessidade de se investigar bases com número de atributos próximos de 10^3. No Brasil, por exemplo, o portal transparência `dados.gov.br` disponibiliza mais de 6 mil conjuntos de dados do governo em diversas áreas, sendo parte deles de alta dimensionalidade.

Dentre os tipos de informações investigadas a partir de dados, encontrar as características que diferenciam dois ou mais grupos é uma das tarefas mais importantes na *mineração de dados* (Liu *et al.*, 2015). Na área de educação, por exemplo, identificar o que diferencia as melhores escolas das piores pode indicar políticas para tornar as escolas mais atrativas. Já na saúde, identificar o que diferencia os tratamentos bem-sucedidos dos fracassados, por exemplo, pode resultar em melhorias nas condutas médicas. Dessa forma, entender o que diferencia um grupo alvo dos demais pode ser o ponto de partida para a solução de problemas relevantes em diversas áreas.

A mineração de subgrupos (do inglês *subgroup discovery*) é uma área da *mineração de dados* que tem o objetivo de identificar os conjuntos de características que melhor diferenciam um grupo alvo dos demais (ex. escolas de referência das demais) (Helal, 2016; Atzmueller, 2015; Herrera *et al.*, 2011). Na prática, tal área representa uma ferramenta capaz de mostrar, de forma legível, informações não acessíveis por métodos tradicionais de análise exploratória de dados. Dessa forma, a mineração de subgrupos foi utilizada na descoberta de conhecimento em problemas de diferentes áreas, como bioinformática (Li & Wong, 2002; Quackenbush, 2001), medicina (Carmona *et al.*, 2013, 2011), *marketing* (Carmona *et al.*, 2012; del Jesus *et al.*, 2007b), *e-learning* (Romero *et al.*, 2009) e acidentes de trânsito (Kavšek & Lavrac, 2004; Kavšek *et al.*, 2002). A mineração de subgrupos é também pesquisada sob diferentes terminologias na literatura, como *Subgroup Discovery* (Herrera *et al.*, 2011), *Discriminative Pattern* (Liu *et al.*, 2015), *Emerging Patterns* (Dong & Li, 1999; Vimieiro & Moscato, 2014) e *Contrast Sets* (Bay & Pazzani, 2001; Azevedo, 2010). No entanto, tais terminologias foram definidas como sendo o mesmo problema por Novak *et al.* (2009).

Na mineração de subgrupos, cada subgrupo pode ser representado por uma regra no formato $cond \rightarrow target_{label}$, onde $cond$ é uma combinação de características/itens (ex. $salarioProfessor = alto, estrutura = excelente$) e $target_{label}$ é o alvo da investigação (ex. $avaliacaoEscola = alta$). Dessa forma, a mineração de subgrupos pode ser considerada uma área intermediária entre os problemas clássicos de *Classificação* e *Descrição* baseadas em regras, uma vez que seu objetivo é descrever dados utilizando regras, como na área de *Descrição*, mas considerando as que melhor discriminam uma classe alvo das demais (Herrera *et al.*, 2011).

A análise manual de todos os subgrupos para um dado problema é uma tarefa inviável, dado o grande número de possibilidades. Dessa forma, diversos algoritmos foram propostos com o objetivo de extrair os subgrupos considerados mais relevantes para um dado problema. Assim, numa segunda etapa, o especialista no domínio do problema pode se dedicar à análise dos subgrupos retornados e decidir quais são realmente relevantes para o seu problema.

Dentre as heurísticas de mineração de subgrupos propostas, têm se destacado as baseadas em pesquisa de feixe (do inglês *beam search*) (Gamberger & Lavrac, 2002; Lavrač *et al.*, 2004; Van Leeuwen & Knobbe, 2012) e computação evolucionária (Carmona *et al.*, 2014, 2015; Martín *et al.*, 2016). Alguns trabalhos mostram uma superioridade das abordagens evolucionárias em relação às baseadas em pesquisa de feixe (Carmona *et al.*, 2010; Luna *et al.*, 2014), mas tais tipos de abordagens ainda não foram comparadas de forma ampla no contexto de bases de dados de alta dimensionalidade. Além disso, ainda não foi proposta uma heurística com foco em bases de dados de alta dimensionalidade, sendo esse um importante problema em aberto na área de mineração de subgrupos (Atzmueller, 2015; Carmona *et al.*, 2014). No contexto desta pesquisa consideramos bases de alta dimensionalidade aquelas com pelo menos 100 atributos, embora o trabalho lide frequentemente com bases com número de atribtuos na ordem de 10^3 e 10^4.

Outra caracterísca comum nas atuais heurísticas de mineração de subgrupos é a grande quantidade de parâmetros e a complexidade de ajuste destes. Parâmetros comumente utilizados, como suporte mínimo, são complexos de serem definidos. Se o valor for muito pequeno, pode não representar uma limitação relevante no espaço de busca, se muito grande, pode restringir de forma exagerada as opções do algoritmo. Além disso, o valor adequado para esse tipo de parâmetro varia de acordo com o problema. As abordagens evolucionárias costumam possuir ainda diversos outros parâmetros, tais como tamanho da população, taxas de cruzamento e mutação, número máximo de avaliações e número máximo de gerações (Carmona *et al.*, 2014, 2015). No contexto de bases de alta dimensionalidade o ajuste de tais parâmetros é ainda mais desafiador, devido ao custo computacional associado a cada teste realizado. Tudo isso pode representar uma barreira relevante para a mineração de subgrupos , principalmente para pesquisadores de áreas não relacionadas à mineração de dados.

Nesse contexto, *seria possível desenvolver um algoritmo de mineração de subgrupos baseado em computação evolucionária eficiente no contexto de bases de dados de alta dimensionalidade e com poucos parâmetros facilmente ajustáveis?*

Por fim, um terceiro desafio na área de mineração de subgrupos é a redundância entre os subgrupos retornados pelos algoritmos (Bosc *et al.*, 2017; Van Leeuwen & Knobbe, 2012). A forma mais comum de combate à redundância para um conjunto de subgrupos é atribuindo pesos aos exemplos das bases de dados, de forma a desvalorizar subgrupos que tenham exemplos em comum. No entanto, essa estratégia pode acarretar um alto custo computacional no contexto de bases de dados de alta dimensionalidade com muitos exemplos. Além disso, a punição dada à redundância de cobertura dos exemplos pode não ser suficiente para evitar a existência de dois ou mais subgrupos cobrindo exatamente os mesmos exemplos. Outro aspecto importante é que subgrupos considerados redundantes podem representar informações relevantes, como uma forma mais viável de resolver um problema ou um conhecimento inédito na área de aplicação. Assim, a diferença entre redundância e informação relevante pode estar associada ao domínio da aplicação, o que torna o problema complexo de resolver.

Assim, *seria possível desenvolver um novo caminho para combate à redundância viável no contexto de bases de alta dimensionalidade, reduzindo o risco de descarte prematuro de subgrupos relevantes e gerando mais informações para o usuário?*

A mineração de subgrupos também pode ser utilizada como ferramenta em outros problemas de mineração de dados de alta dimensionalidade. Um deles é a *descrição do perfil de comunidades* (do inglês *Group Profiling*), cujo objetivo é descrever um grupo de pessoas que compartilha valores pessoais e/ou interesses comuns (Tang *et al.*, 2008). Existem várias aplicações relacionadas à *descrição do perfil de comunidades*, como entender estruturas sociais, visualização e navegação de redes, identificação de mudanças em temas de grupo e *marketing* direto (Tang *et al.*, 2011).

Os atuais métodos para descrever comunidades comumente resultam em descrições univariadas (Gomes *et al.*, 2018; Tang *et al.*, 2011; Gomes *et al.*, 2016, 2013). No entanto, métodos univariados negligenciam interações interessantes entre características, o que poderia melhorar a descrição geral de uma comunidade. Outra questão atualmente não abordada pelos atuais métodos diz respeito à cobertura das descrições. Isso pode representar uma limitação relevante, uma vez que descrições com baixa cobertura representam apenas um pequeno subconjunto dos membros de uma comunidade.

Nesse contexto, *seria a mineração de subgrupos um caminho promissor para a geração de descrições abrangentes e mais informativas para o problema de descrição do perfil de comunidades?*

Dessa forma, considerando a relevância da mineração de subgrupos na descoberta de conhecimento discriminante, o potencial de aplicação em outras áreas da mineração de dados, bem como as limitações dos atuais algoritmos, segue a descrição dos objetivos dessa pesquisa.

1.1 OBJETIVOS

Esta tese tem o objetivo de propor soluções para a mineração de subgrupos no contexto de bases de dados de alta dimensionalidade. Os avanços na área foram realizados em três frentes:

1. Propor algoritmo de mineração de subgrupos com foco em base de dados de alta dimensionalidade e na simplicidade de uso.
2. Propor método de combate à redundância de forma a reduzir o risco de descarte prematuro de subgrupos relevantes e gerar mais informações para o usuário no contexto de bases de dados de alta dimensionalidade.
3. Propor modelo de descrição de comunidades abrangente e multivariado com base no uso de mineração de subgrupos.

1.2 PRODUÇÃO BIBLIOGRÁFICA

Essa seção lista a produção bibliográfica desenvolvida neste doutorado. Dessa forma, foram publicados dois artigos em congressos e um em periódico. Existe ainda um terceiro trabalho submtido e um quarto que foi publicado, mas não relacionado ao tema da tese.

1.2.1 Publicação em periódico

- LUCAS, T. D. P.; SILVA, T. C.; VIMIEIRO, R.; LUDERMIR, T. B. A new evolutionary algorithm for mining top-k discriminative patterns in high dimensional data. APPLIED SOFT COMPUTING, v. 59, p. 487-499, 2017.

1.2.2 Publicação em congressos

- LUCAS, T. D. P.; VIMIEIRO, R.; LUDERMIR, T. B. SSDP: A Simple Evolutionary Approach for Top-K Discriminative Patterns in High Dimensional Databases. In: 2016 5th BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 2016, Recife. p. 361-366.
- LUCAS, T. D. P.; VIMIEIRO, R.; LUDERMIR, T. B. SSDP+: a Diverse and More Informative Subgroup Discovery Approach for High Dimensional Data. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE CEC), 2018, Rio de Janeiro. p. 1-8.
- LUCAS, T. D. P.; GOMES, J. E. A.; VIMIEIRO, R.; LUDERMIR, T. B.; PRUDÊNCIO, R. B. C. A multivariate method for Group Profiling using Subgroup Discovery. **(submetido)**.

1.2.3 Publicação não relacionada à tese

- BEZERRA, C. ; SCHOLZ, R. ; ADEODATO, P. ; LUCAS, T. D. P. ; ATAIDE, I. Evasão Escolar: Aplicando Mineração de Dados para Identificar Variáveis Relevantes. In: XXVII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2016, Uberlandia. p. 1096-1105.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Esta tese está organizada em formato de artigos. Assim, os três capítulos seguintes são os principais artigos desenvolvidos na pesquisa. No Capítulo 2 nós propomos o algoritmo *SSDP* (do inglês *Simple Search Discriminative Pattern*) (Lucas *et al.*, 2017), a primeira abordagem evolucionária para mineração de subgrupos com foco em bases de dados de alta dimensionalidade. O modelo proposto possui apenas dois parâmetros facilmente ajustáveis. Já no Capítulo 3 nós apresentamos o *SSDP+* (Lucas *et al.*, 2018), uma evolução do *SSDP* que trouxe uma forma inovadora de lidar com o problema de redundância, gerando mais informações para o usuário. No Capítulo 4 nós nós propomos o *MGP-SD* (do inglês *Multivariate Group Profiling - Subgroup Discovery*), o primeiro modelo multivariado para *descrição do perfil de comunidades* com controle de cobertura global. O *MGP-SD* é baseado no algoritmo *SSDP+*. Já no Capítulo 5 nós apresentamos as nossas conclusões e trabalhos futuros. Por fim, no Anexo A nós disponibilizamos o artigo que apresentou o protótipo que deu origem a versão final do SSDP (Pontes *et al.*, 2016).

2 SSDP: SIMPLE SEARCH DISCRIMINATIVE PATTERNS

Este capítulo é uma reprodução completa do artigo *A new evolutionary algorithm for mining top-k discriminative patterns in high dimensional data*, publicado em 2017 na revista *Applied Soft Computing*, volume 59, p. 487-499 (Lucas *et al.*, 2017).

2.1 ABSTRACT

This paper presents an evolutionary algorithm for Discriminative Pattern (DP) mining that focuses on high dimensional data sets. DPs aims to identify the sets of characteristics that better differentiate a target group from the others (e.g. successful vs. unsuccessful medical treatments). It becomes more natural to extract information from high dimensionality data sets with the increase in the volume of data stored in the world (30GB/s only in the internet). There are several evolutionary approaches for DP mining, but none focusing on high-dimensional data. We propose an evolutionary approach attributing features that reduce the cost of memory and processing in the context of high-dimensional data. The new algorithm thus seeks the best (top-k) patterns and hides from the user many common parameters in other evolutionary heuristics such as population size, mutation and crossover rates, and the number of evaluations. We carried out experiments with real-world high-dimensional and traditional low dimensional data. The results showed that the proposed algorithm was superior to other approaches of the literature in high-dimensional data sets and competitive in the traditional data sets.

2.2 INTRODUCTION

This paper presents an evolutionary algorithm for Discriminative Pattern (DP) mining that focuses on high dimensional data sets. Discriminative pattern mining is a data mining task that has the objective of identifying sets of items that distinguish a target group from the others, for example: successful from unsuccessful treatments, unhealthy from healthy cells, spam from other emails, or even positive from negative sentiments in sentiment analysis. The necessity to investigate new methods, especially heuristics methods mining these patterns, comes from the fact that data generated/collected from many domains have different characteristics from those of last decade. The vast amount and high dimensionality of data sets in this so-called *Era of Big Data* render the application of existing methods infeasible.

Studies estimate that in the internet alone around 30GB of data, including texts, images and videos, are produced each second. Another important source of data is the biomedical sciences, particularly the *Omics* (genomics, proteomics, transcriptomics, ...), since the price for sequencing samples has dramatically dropped in the last years. These areas have two things in common: (1) they are major contributors to today's massive amount of available data (big data); (2) data from these domains is usually very high dimensional with tens of thousands to millions of attributes. In this sense they present new challenges to data mining and machine learning researchers. Among these challenges is the need for new tools for exploratory data analysis.

Discriminative patterns are an important tool for exploratory data analysis, since recurring patterns in the data are summarized in a simple way (Liu *et al.*, 2015). This is particularly suitable to explaining/describing differences among groups of samples in the data. Discriminative pattern mining has simultaneously evolved with different terminologies, *Subgroups Discovery* (Atzmueller, 2015; Herrera *et al.*, 2011); *Emerging Patterns* (Dong & Li, 1999; Blinova *et al.*, 2003); and *Contrast Sets* (Bay & Pazzani, 2001), until they were unified by (Novak *et al.*, 2009). There are many applications reported in the literature in different domains such as: medicine (Carmona *et al.*, 2013, 2011), bioinformatics (Li & Wong, 2002; Quackenbush, 2001), marketing (Carmona *et al.*, 2012; del Jesus *et al.*, 2007b), e-learning (Romero *et al.*, 2009) and traffic accidents (Kavšek & Lavrac, 2004; Kavšek *et al.*, 2002).

Little attention has been given to mining discriminative patterns in high dimensional domains in spite of the great number of applications in the literature. High dimensionality of data sets is an intricate problem for current methods for discriminative pattern mining. It represents a computationally difficult problem for most of existing methods because of their combinatorial nature. Most of the exact methods, e.g. (Kavšek *et al.*, 2006; Vimieiro & Moscato, 2014), enumerate subsets of attributes, avoiding and discarding paths in the search space that exclusively yield uninteresting patterns. In fact Vimieiro (2012) already discussed in 2012 the issues related to exact methods for mining discriminative patterns. He argues that the feasibility of such methods is not only limited by computational aspects (time and memory usage), but also by the number of returned patterns. In many occasions the problem is just shifted from analyzing raw data to analyzing a huge number of patterns. This motivates the investigation of heuristics for mining discriminative patterns.

There are plenty of heuristics for mining discriminative patterns, including many based on *evolutionary computing* (del Jesus *et al.*, 2007b,a; Carmona *et al.*, 2010; Pachón *et al.*, 2011; Rodríguez *et al.*, 2012; Luna *et al.*, 2014; Carmona *et al.*, 2015; Pulgar-Rubio *et al.*, 2016). The vast majority of these methods target traditional, low-dimensional data sets. As their exact counterparts, they also use interestingness measures to guide the search. These constraints are mostly related to the frequency (support) and discriminative power of patterns. Thus, they explicitly deal with the computational issues associated to exact methods, but might not solve the second issue related to the number of patterns. The algorithms for mining discriminative patterns usually return the best patterns in one of two ways: (1) based on constraints, which return patterns that satisfy some constraint, as minimum support; and (2) based on top-*k*, which return the *k* best patterns. Both options have their relevance depending on the analysts' goals, but the top-*k* approach provides more flexibility (Atzmueller, 2015). Notwithstanding, an evolutionary top-*k* DPs mining approach has not been proposed yet.

This context motivates us to pose the following research question: *is it possible to devise a new evolutionary heuristic that tackles both the combinatorial issues and huge amount of patterns associated with high dimensional data?* To address this question, we present a new evolutionary heuristic SSDP (Simple Search Discriminative Patterns). We aim at providing end-users a viable and easy to use tool for analyzing high dimensional data. Our approach allows the user to choose the most appropriate interestingness measure and requires only the number of patterns that she intends to analyze. The algorithm then seeks the best (top-*k*) patterns, hiding from the user many common parameters in other evolutionary heuristics such as population size, mutation and crossover rates, and the number of evaluations.

SSDP was first presented as a preliminary work at the 5th Brazilian Conference on Intelligent Systems (BRACIS 2016) (Pontes *et al.*, 2016). However, we made additional progress as following. We improved our experiments to assess the performance of our approach with both real-world high-dimensional and traditional low dimensional data. We compared the results from our algorithm with other traditional evolutionary methods, which had not been previously done. The aim of these new experiments was to evaluate both the effectiveness of SSDP on mining high-dimensional data, which it has been designed for, and its suitability to different contexts (low dimensional data, which it has not been designed for). Since we omit many common parameters as discussed above, we also conducted experiments to investigate different settings of these parameters and their impact on our method. Such an analysis had not been done in the previous conference paper, despite being extremely important to confirm whether the choices made indeed return relevant patterns compared to other settings. Finally, we also revised the entire manuscript and made significant changes to improve its readability.

The remainder of this manuscript is organized as follows. We formalize the problem of mining discriminative patterns in section 2.3. We formally define the concept of a discriminative pattern and the interestingness measures to assess its relevance. In section 2.4, we review the literature, providing a critical analysis of the state of the art. We identify the issues related to the current methods for mining discriminative patterns. We present our algorithm in section 2.5. Then we discuss the experiments conducted to assess the performance of our algorithm and compare the results with other algorithms in section 2.6. We conclude the manuscript with some final remarks in section 2.7.

Table 1: A toy example of a data set. In this simulated data, the target is to identify the differences between successful and unsuccessful medical treatments for a given disease.

example	genre	age	medicine	label
e_1	M	senior	B	success
e_2	F	senior	B	success
e_3	M	senior	A	success
e_4	M	adult	A	success
e_5	F	child	A	success
e_6	F	child	A	failure
e_7	M	child	B	failure
e_8	F	child	B	failure
e_9	M	adult	A	failure
e_{10}	F	adult	A	failure

2.3 DISCRIMINATIVE PATTERNS

Let D be a labeled data set with a set A of categorical/discrete attributes. According to the class label, the set of samples from D can be partitioned into $D^+ = \{e_1^+, e_2^+, ..., e_{|D^+|}^+\}$ and $D^- = \{e_1^-, e_2^-, ..., e_{|D^-|}^-\}$, respectively the positive (target) examples and the remaining (negative examples). Let $dom(A_i)$ be the domain of values for attribute $A_i \in A$. We call features or items the set of all pairs $(attribute, value)$, that is $I = \bigcup A_i \times dom(A_i) = \{i_1, i_2, ..., i_{|I|}\}$. We say that an example d has an item $x = (A_i, v) \in I$ if d has value v for the attribute A_i.

We call a *discriminative pattern* a set $dp \subseteq I$. The *size* of a discriminative pattern dp is the number of items in dp, that is $size(dp) = |dp|$. Every dp might be associated (cover) a set of positive and negative examples, which we formally define as $c^+(dp) = \{d \in D^+ \mid d \text{ has all items in } dp\}$, and $c^-(dp) = \{d \in D^- \mid d \text{ has all items in } dp\}$. The size of these two sets define the *positive* and *negative support* of a discriminative pattern, i.e. its frequency among positive and negative examples, and their sum defines the overall support of the patterns.

Table 1 contains a toy example of data set, for which the aim is to identify the differences between successful and unsuccessful medical treatments for a given disease. In this example, Table 1 represents the data set D and $label = success$ is the target of investigation . Thus, $D^+ = \{e_1, e_2, e_3, e_4, e_5\}$ are the positive examples (where $label = success$) and $D^- = \{e_6, e_7, e_8, e_9, e_{10}\}$ are the negative examples (where $label \neq success$). Meanwhile Table 2 represents the universe of items $I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$ and the respective positive and negative covered examples. In this context, $dp = \{i_5\}$ is an interesting discriminative pattern, once $c^+(dp) = |\{e_1 e_2, e_3\}| = 3$ and $c^-(dp) = |\emptyset| = 0$. On the other hand, $dp = \{i_1, i_7\}$ is not an interesting pattern as it is equally frequent among positive and negative samples ($c^+(dp) = |\{e_1\}| = 1$ and $c^-(dp) = |\{e_7\}| = 1$).

The definition of the relevance/interestingness of a discriminative pattern is given by a measure (Flach *et al.*, 1999). Flach *et al.* (1999) present a thorough review on several types of evaluation/interestingness measures for discriminative patterns. They discuss how the measures relate to each other, often describing the same, while, in spite of it, there is still no consensus interference. Table 3 summarizes the characteristics of the algorithms reviewed here.

Table 2: Universe of items I and respective covered examples for the data presented in Table 1. In this table, items are in rows and examples in columns. There is a cross if an example has the corresponding item.

I	**(attribute, value)**	D^+					D^-				
		e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
i_1	(genre, M)	×		×	×			×		×	
i_2	(genre, F)		×			×	×		×		×
i_3	(age, child)		×			×	×		×		
i_4	(age, adult)				×					×	×
i_5	(age, senior)	×	×	×							
i_6	(medicine, A)			×	×	×	×			×	×
i_7	(medicine, B)	×	×					×	×		

about the best one. This choice often depends on the problem or specialist's convictions. In this way, it is important for discriminative pattern mining algorithms to accept different options of evaluation metrics to meet user needs.

One of the most used evaluation metric is the weighted relative accuracy (WRAcc), given by Equation 2.1:

$$WRAcc(dp) = \frac{TP+FP}{|D|}\left(\frac{TP}{TP+FP} - \frac{|D^+|}{|D|}\right), \tag{2.1}$$

where $TP = |c^+(dp)|$ (the positive support) and $FP = |c^-(dp)|$ (the negative support).

As described by Flach *et al.* (1999), the WRAcc is a trade-off between generality and accuracy. The first part of the equation (outside parethesis) accounts for the generality of the pattern. Patterns covering more samples, i.e. more general, are, at first, preferred to more specific patterns. The second part (inside parenthesis) corresponds to the relative accuracy of the pattern. Patterns showing a gain relative to the fixed rule assigning/describing all samples to/from the positive class are preferred. *WRAcc* values range from -0.25 to $+0.25$, or from -1 to $+1$ in its normalized form ($WRAcc_{normalized} = 4 \times WRAcc$) (Flach, 2003). In this case $+1$ represents a totally pure pattern, describing all the positive examples and none of negative examples, while -1 represents a pattern describing all the negative examples and none of the positive, and 0 indicates that there is no gain relative to the fixed rule describing all examples of the base as positive.

Another well-known metric is the Q_g (Gamberger & Lavrac, 2002), given by Equation 2.2, where g is a generalization parameter, which defaults to 1. The value of g represents the tolerance to negative examples in relation to positives covered by a DP. The higher the g value, the more generic DPs will be. On the other hand, the closer to zero the value of g, the more specific and intolerant to FP the best DPs will be (Gamberger & Lavrac, 2002).

$$Q_g = \frac{TP}{FP+g}, \tag{2.2}$$

Q_g can be used as an alternative to the evaluation metric $GrowthRate = \frac{TP}{|D^+|} / \frac{FP}{|D^-|}$ (Dong & Li, 1999), often used in works mining DPs from bioinformatics data (Li & Wong, 2002; Yu *et al.*, 2004; Vimieiro & Moscato, 2014) without the disadvantage of division by zero, though. Q_g can assume values between 0 and $+\infty$. A complete survey on many evaluation metrics can be found in (Liu *et al.*, 2015; Herrera *et al.*, 2011).

There are also some global metrics, whose purpose is to evaluate a DP set. One of them is the overall support $SUPP^+$ (Helal, 2016), witch measures the percentage of target examples D^+ covered by a DP set (Equation 2.3). This metric is important to evaluate if a DP set significantly covers D^+ or if it is restricted to a small subset of them. $SUPP^+$ can assume values between 0 and 1, where 1 means that a DP set completely cover D^+.

$$SUPP^+ = \frac{|c^+(dp_1) \cup \ldots \cup c^+(dp_k)|}{|D^+|}, \tag{2.3}$$

One of the challenges in discriminative patterns is the redundancy in a DP set. Two of the most common type are coverage and description redundancy (Van Leeuwen & Knobbe, 2012). Coverage redundancy occurs when a DP set has many positive examples in common (e.g. $dp_1 = \{i_2\}$ and $dp_2 = \{i_3\}$ in Table 2, where $c^+(dp_1) = c^+(dp_2) = \{e_2, e_5\}$). A DP set with high coverage redundancy usually has low $SUPP^+$. On the other hand, description redundancy occurs when a DP set has one or more items in common. The DPs $dp_1 = \{i_1\}$, $dp_2 = \{i_1, i_3\}$ and $dp_3 = \{i_1, i_3, i_6\}$ in Table 2, for example, describes only male patients ($i_1 \rightarrow (genre, M)$). Both can result in poor information for end user.

2.4 RELATED WORK

The area of discriminative pattern mining evolved in parallel from three different areas: *Subgroup Discovery* (Atzmueller, 2015; Herrera *et al.*, 2011), *Emerging Patterns* (Dong & Li, 1999) and *Contrast Sets* (Bay & Pazzani, 2001). *Subgroup Discovery* is the extraction of subgroups of interest related to the value of label (Herrera *et al.*, 2011). *Emerging Patterns* are groups where the difference of frequency with respect to two classes diverges to a rate of gain (Dong & Li, 1999). At last, *Contrast Set* are conjunctions of attributes and values that significantly differ in their distributions (Bay & Pazzani, 2001). In 2009, Novak *et al.* (2009) discussed how these areas related to each and classified them as being the same problem. However, one of the first works to use the term *discriminative pattern* were the articles by Gao & Wang (2010) and Pandey *et al.* (2010). Liu *et al.* (2015) were the first to survey the area from the perspective of bioinformatics.

Discriminative pattern mining algorithms usually return the best patterns in one of two ways. The most popular way is constraint-based searching, where the algorithm traverse the search space keeping patterns that satisfy a given constraint while avoiding paths with unpromising patterns. This technique was borrowed from *association rule mining* algorithms (Agrawal

& Srikant, 1994) and, hence, often uses as constraints measures such as minimum support and confidence. Nevertheless, setting the thresholds for those constraints might not be a simple task. If it is too large, the algorithm may not return any results (Han *et al.*, 2002). On the other hand, if it is small, it does not effectively filter uninteresting patterns. This is particularly critical when dealing with high dimensional data sets as a huge number of patterns may satisfy a constraint (Vimieiro, 2012; Vimieiro & Moscato, 2014). In other words the longer patterns found in high dimensional data may yield an exponential number of sub-patterns that also satisfy the constraint; this turns out to be always true if the interestingness measure is anti-monotonic. An alternative to the constraint-based approach is to find patterns based on (implicit) rankings. The aim of this second approach is to find the top-k patterns with the highest values for a given interestingness measure. In this scenario the user provides the number k of patterns to be found and the algorithm searches for the best ones accordingly.

There are several exact and heuristic data mining algorithms (Herrera *et al.*, 2011; Liu *et al.*, 2015; Carmona *et al.*, 2014; Helal, 2016). Among the heuristic algorithms, the ones based on *beam search* (Gamberger & Lavrac, 2002; Lavrač *et al.*, 2004; Van Leeuwen & Knobbe, 2012) and *evolutionary computing* (del Jesus *et al.*, 2007b,a; Carmona *et al.*, 2010; Pachón *et al.*, 2011; Rodríguez *et al.*, 2012; Luna *et al.*, 2014; Carmona *et al.*, 2015; Pulgar-Rubio *et al.*, 2016; Carmona *et al.*, 2014) are most important ones.

The algorithms based on *beam search* are initialized from a predefined number of DPs determined by a *beamSize* parameter. New patterns are generated from the *beamSize* ones from the previous iteration. Therefore, approaches based on beam search restrict memory usage by exploring only part of the search space. One of the first and most prominent algorithm for mining discriminative patterns based on beam search is SD (Gamberger & Lavrac, 2002). The algorithm starts the search by taking the highest quality (according to Q_g and minimal support) items as singleton discriminative patterns. After that, the algorithm replaces the least relevant patterns by the most relevant ones with larger sizes. The SD stops the search when there is no change in list of relevant patterns over one iteration.

One of the greatest disadvantage of beam search algorithms is the lack of diversity. The algorithms usually target only individually good items, which, by the point of view of domain experts, might already be a well-known pattern (Fang *et al.*, 2011; Garriga *et al.*, 2008). In this scenario, evolutionary algorithms represent a perfect fit, having many methods been proposed in the literature. We now review some of the most important evolutionary algorithms for discriminative pattern mining, and refer the reader to the work of Carmona *et al.* (2014), which provides a thorough survey of the area.

SDIGA is a mono-objective approach that uses a global search followed by a local search for each iteration. The global search is performed by the genetic algorithm and the local search, via *Hill Climbing*. Two other algorithms are *MESDIF* and *NMEEF*. These algorithms are multi-objective, the first being based on the *SPEA2* (Zitzler *et al.*, 2001) algorithm and the second one on the *NSGA-II* (Deb *et al.*, 2002). MESDIF uses elitism and the concept of *Pareto Front* in its

Table 3: Summary of the characteristics of the main evolutionary algorithms for mining discriminative patterns.

Algorithm	**Objective**	**Size of individuals**	**Initial population**	**top-k**	**Number of parameters**
SDIGA	Mono	$\|I\|$	?	NO	7
MESDIF	Multi	$\|I\|$	?	NO	7
NMEEF	Multi	$\|I\|$	75% of individuals with up to 25% of of items $i \in I$	NO	7
EDER	Mono	$\|A\|$	Based on examples	NO	4
CGBA	Mono	$size(dp)$	Random until all individuals are valid	NO	4
FuGePSD	Mono	$size(dp)$	1% to 50% of items $i \in I$, until all individuals are valid	NO	14
MEFASD	Multi	$\|I'\|$ where $I' \subset I$	?	NO	8

search strategy, and NMEEF uses an operator to reset the population. NMEEF has been one of the most competitive approaches when compared with other algorithms (Luna *et al.*, 2014; Carmona *et al.*, 2015; Helal, 2016). *FuGePSD* (Carmona *et al.*, 2015) uses genetic programming (Koza, 1992) and represents individuals with trees. In addition, *FuGePSD* performs both local and global search while attempting to cover all positive examples of the D^+ database. Finally, *MEFASD-BD* is an approach focused on *big data* in relation to the number of examples. *MEFASD-BD* uses the *MapReduce* paradigm to partition the data set, and concepts of *NMEEF* to mine the DPs. These algorithms use fuzzy logic to deal with numeric attributes. There are, however, other evolutionary approaches for mining DPs that do not use the *Genetic Fuzzy System* (Herrera, 2008). *EDER* (Rodríguez *et al.*, 2012) is a mono-objective approach based on *HIDER* (Aguilar-Ruiz *et al.*, 2001) (HIerarchical DEcision Rules). *EDER* focuses on issues with minority classes in unbalanced databases. *CGBA* (Luna *et al.*, 2014), on the other hand, is an approach that uses evolutionary programming as a search strategy and *context-free grammar* to represent DPs in a readable and flexible way. In addition, *CGBA* dynamically defines crossover and mutation rates while searching without user interference. Table 3 summarizes the characteristics of the algorithms reviewed here.

Despite the large number of evolutionary approaches, none of them was developed with focus on high dimensionality and most of the performance tests considered data sets with less than 40 attributes. In addition, some features of such models can be problematic in the context of high dimensionality. The representation of individuals using one gene per item (or attributes), for example, can bring high cost of memory in the context of high dimensionality. At the same time, limiting the size of individuals in the initial population to percentages of $|I|$ tends to generate large random DPs that do not cover any example of D, which may restrain the convergence of the algorithm. Finally, such models usually have some non-trivial configuration parameters and none of them is top-k.

More recently, researchers are also reconsidering sampling algorithms as an alternative to exact/enumerative methods (Bendimerad *et al.*, 2016; Boley *et al.*, 2011; Moens & Boley, 2014; Kaytoue *et al.*, 2017). Sampling algorithms most often use Monte Carlo Markov Chain methods to find patterns via the distribution of their support or a quality measure based on it. These algorithms are particularly useful for interactive exploratory analysis as samples may be drawn sequentially from the given distribution (Scholz, 2005). Nevertheless, most of the works in this area are still focusing on low dimensional data. Boley *et al.* (2011), for instance, restricted their experiments to UCI data with less than 300 dimensions (and 4000 samples). Since our focus is on batch analysis of very high dimensional data sets, such as those from unstructured textual or biomedical data, we do not consider these approaches in our experiments.

2.5 SSDP: SIMPLE SEARCH DISCRIMINATIVE PATTERNS

SSDP is a mono-objective evolutionary approach for discriminative pattern mining. Its main characteristics are: (1) being adapted to high dimensional data and (2) having few easily adjustable parameters.

In SSDP, individuals represent only the items used in the DP. The rationale for using such a representation lies on the fact that the best patterns usually contain less than 1% of the items. Therefore, each individual of the population is represented by one or more integers. Each integer (or index) corresponds to the position of an item i in I (assuming any total ordering of items). A two-dimensional discriminative pattern $dp = \{2043, 213\}$, for example, represents the set formed by items in positions 2043 and 213 of I. However, when representing individuals as sets of integers, it is necessary to ensure that there is no duplicity (eg. $dp = \{2043, 2043, 213\}$). We implemented individuals using hash tables to avoid duplicity and maintain the performance of the algorithm.

SSDP initializes the searches with patterns of size one and evolves to higher dimensions through its evolutionary operators. The initial population is composed of all one dimensional possible DPs (an individual for each $i \in I$). Such an initialization allows the population size to be determined automatically according to the problem ($populationSize = |I|$). Besides, it ensures that all items $i \in I$ are considered in the search. Initializing the search from one-dimensional solutions is a novelty among evolutionary approaches for mining DPs. However, it is widely used in algorithms based on Beam Search (Gamberger & Lavrac, 2002; Kavšek *et al.*, 2006; Mueller *et al.*, 2009). In addition, in high dimensional bases, initializing a search by randomly generated individuals may restrain the convergence of the algorithm, as we discuss in subsection 2.6.1.

After the initial population is generated, SSDP uses the following genetic operators to generate new candidates. The selection is made by binary tournament. In mutation there are three possibilities with the same probability: (1) a random item is added to the individual (e.g. $i = \{a, b, c\} \rightarrow i' = \{a, b, c, d\}$); (2) a random item is replaced by another (e.g. $i = \{a, b, c\} \rightarrow i' = \{a, b, d\}$); and (3) a random item is removed from the individual (e.g. i=$\{a, b, c\} \rightarrow i' = \{a, b\}$).

Therefore, in the mutation, an individual with size d randomly evolves to dimension d, $d-1$ or $d+1$. It is common that just one item changes in evolutionary approaches for DP mining. That happens because the change in a single item represents a significant transformation in the individual.

With respect to crossover method, there are two possibilities: *crossOverAND* and *crossOverUniform*. The first one generates an individual from the union of the two individuals' items (e.g. $i_1 = \{a\}$ and $i_2 = \{b\} \rightarrow i' = \{a,b\}$). This type of crossing is used only in the initial population, in which all individuals have size=1. While in *crossOverUniform* crossing, two individuals generate two new by uniform crossover with 50% mixing ratio (e.g. $i_1 = \{a,b\}$ and $i_2 = \{c,d\} \rightarrow i'_1 = \{a,d\}$ and $i'_2 = \{b,c\}$).

Crossover and mutation rates initialize at 0.6 and 0.4, respectively, and are adapted according to the search. If, at the end of a generation, there is improvement in the top-k DPs, the algorithm increases the crossover rate at 0.2 and reduces mutation rate at the same value. When there is no improvement in the top-k DPs, the mutation rate increases by 0.2 and the crossover rate decreases by the same value. Thus, the algorithm tends to intensify the search in depth when it is in a promising region, otherwise, it tends to intensify the search in breadth. The mutation and crossover rates always sum to one. This methodology is an adaptation of the one proposed by Luna *et al.* (2014) for CGBA (described in section 2.4).

The SSDP uses as stopping criterion the stabilization of the group of the k best DPs after the population has been reset twice. A population is reset when there is no change in top-k DPs for three consecutive generations and the mutation rate is equal to one. In this process the algorithm randomly generates individuals of fixed size between two and the average size of the top-k DPs. Moreover, 10% of individuals are generated using exclusively items present in top-k DPs.

SSDP does not allow the user to tune some common parameters, such as mutation and crossover rate, population size and minimal support. The algorithm has only two input parameters: the number of DPs (k) and evaluation metric (fitness). SSDP theoretically allows the use of any interestingness measure as fitness. Currently, SSDP implementation includes three evaluation metrics: Q_g, $WRAcc$ and $SUB = TP - FP$.

Algorithm 1 contains the pseudocode of SSDP. In the algorithm, the population P_k keeps the best k individuals that are relevant. An individual dp_i is considered irrelevant in relation to population P_k if $\exists dp \in P_k | c^+(dp_i) \subset c^+(dp) \wedge c^-(dp) \subset c^-(dp_i)$. The other populations (P, P_{new} and $P*$) allow the presence of duplicated individuals. The control over the individuals is made only in the population P_k in an attempt to minimize the computational costs of the algorithm and at the same time return only non-redundant DPs (P_k) to the end-user. The algorithm was implemented in Java and is available from our supporting website (`https://github.com/tarcisiodpl/ssdp`).

Algorithm 1 SSDP pseudocode

```
Require: $k$, $metricEvaluation$
  $P \leftarrow \{\{i_1\},\{i_2\},...,\{i_{|I|}\}\}$
  $P_k \leftarrow kBestRelevants(P)$
  $reinializationCount \leftarrow 0$
  $mutationRate \leftarrow 0.4$
  $crossoverRate \leftarrow 0.6$
  while $reinializationCount < 2$ do
    while $P_k$ not improve three consecutive generations keeping $mutationRate == 1.0$ do
      if generation == 1 then
        $P_{new} \leftarrow crossoverAND(P)$
      else {generation > 1}
        $P_{new} \leftarrow evolutionaryOperator(P, mutationRate, crossoverRate)$
      end if
      $P* \leftarrow best(P, P_{new})$
      $P_k \leftarrow kBestRelevants(P_k, P_*)$
      $update(mutationRate, crossoverRate)$
      $P \leftarrow P*$
    end while
    $reinializationCount++$
    $P \leftarrow$ restart
  end while
  return $P_k$
```

2.6 EXPERIMENTS

The experiments were performed in two groups of data sets, one with high dimensionality and another one traditional. The high dimensionality group (Table 4) consists of 21 microarray bases, available in the package *datamicroarray* (Ramey, 2016) from R software. The bases have between 456 and $54,613$ numerical attributes, and between 31 and 248 examples. For each data set, the majority class was considered the target (positive) and the remaining were labeled as negative. The attributes were discretized prior to applying the algorithm. Since it is not our goal to discuss the implications of the discretization method on the discriminative pattern mining algorithms, we used the simplest methods based on equal frequency and width with 2 , 4 and 8, bins. Such a preprocessing step resulted in a total of 126 data sets composed exclusively of binary (interval based) attributes (items).

The traditional group (of low dimensionality) is formed by 20 data sets extracted from the UCI repository (Lichman, 2013). The bases have between 10 and $12,960$ examples, between 6 and 69 attributes, and are made exclusively by discrete attributes. Table 5 describes the bases with more details, where $|D|$, $|D^+|$ and $|D^-|$ are, respectively, the amount of examples, positive examples and negative examples of databases. The columns *attributes* and $|I|$ are, respectively, the number of attributes and items $i \in I$.

Figure 1 graphically summarizes all (the 20 UCI and the 126 high dimensional) data sets

Table 4: Summary of the 21 high dimensional data sets used in our experiments to assess the performance of SSDP. The columns $|D|$, $|D^+|$ and $|D^-|$ contains the total number of examples, and the number of positive and negative examples after mapping the most frequent label in the data to positive and the remaining to negative. The column *Attributes* contains the number of numeric attributes in the original data, while the remaining columns contain the number of items in the discretized data set using either equal frequency or width with the corresponding number of bins.

					Size of I with Equal Frequency			**Size of I with Equal Width**		
Name	$\lvert D\rvert$	$\lvert D^+\rvert$	$\lvert D^-\rvert$	**Attributes**	**2 bins**	**4 bins**	**8 bins**	**2 bins**	**4 bins**	**8 bins**
alon	62	40	22	2000	4000	8000	16000	4000	7909	14734
borovecki	31	17	14	22283	44566	89132	178257	44566	87611	159671
burczynski	127	59	68	22283	44566	89132	178264	44566	88588	170312
chiaretti	128	74	54	12625	25250	50500	101000	25250	50429	98837
chin	118	75	43	22215	44430	88860	177719	44430	88254	169006
chowdary	104	62	42	22283	44566	89118	177748	44566	82456	133038
christensen	217	113	104	1413	2826	5652	11304	2827	5555	10425
golub	72	47	25	7129	14258	28516	57032	14258	28129	52989
gordon	181	150	31	12533	25066	50132	100264	25355	49807	91869
gravier	168	111	57	2905	5810	11620	23240	5811	11530	22165
khan	63	23	40	2308	4616	9232	18464	4616	9219	17930
nakayama	105	21	84	22283	44566	89132	178264	44566	87038	160167
pomeroy	60	39	21	7128	14256	28512	57024	14256	28202	53494
shipp	77	58	19	7129	14258	28516	57032	14258	27475	49714
singh	102	52	50	12600	25200	50132	97804	25200	49558	91633
sorlie	85	32	53	456	912	1824	3648	946	1860	3555
subramanian	50	33	17	10100	20200	40400	80800	20200	40191	77625
sun	180	81	99	54613	109227	218453	436905	109227	215499	410113
tian	173	137	36	12625	25250	50500	101000	25250	49780	94021
west	49	25	24	7129	14258	28516	57032	14258	27231	48924
yeoh	248	79	169	12625	25250	50500	101000	25250	50427	99689

Table 5: Summary of the 20 UCI data sets used in our experiments to assess the performance of SSDP. The columns $|D|$, $|D^{+}|$ and $|D^{-}|$ contains the total number of examples, and the number of positve and negative examples after mapping the most frequent label in the data to positive and the remaining to negative. The column *Attributes* contains the number of attributes in the data, while $|I|$ is the number of items (attribute,value) pairs.

Name	$\|D\|$	$\|D^{+}\|$	$\|D^{-}\|$	**Atributtes**	$\|I\|$
audiology	226	57	169	69	154
kr-vs-kp	3196	1669	1527	36	73
lung-cancer	32	13	19	56	157
molecular-biology_promoters	106	53	53	58	334
soybean	683	92	591	35	99
trains	10	5	5	32	77
splice	3190	1655	1535	61	3465
breast-cancer	289	201	85	9	41
bridges_version2	105	44	61	12	191
car	1728	1210	518	6	21
monks-problems-1_train	124	62	62	6	17
postoperative-patient-data	90	64	26	8	23
primary-tumor	339	84	255	17	37
shuttle-landing-control	15	9	6	6	16
solar-flare_2	1,066	331	735	12	42
spect_test	187	172	15	22	44
tic-tac-toe	958	626	332	9	27
vote	435	267	168	16	32
mushroom	8124	4208	3916	22	116
nursery	12960	4320	8640	8	27

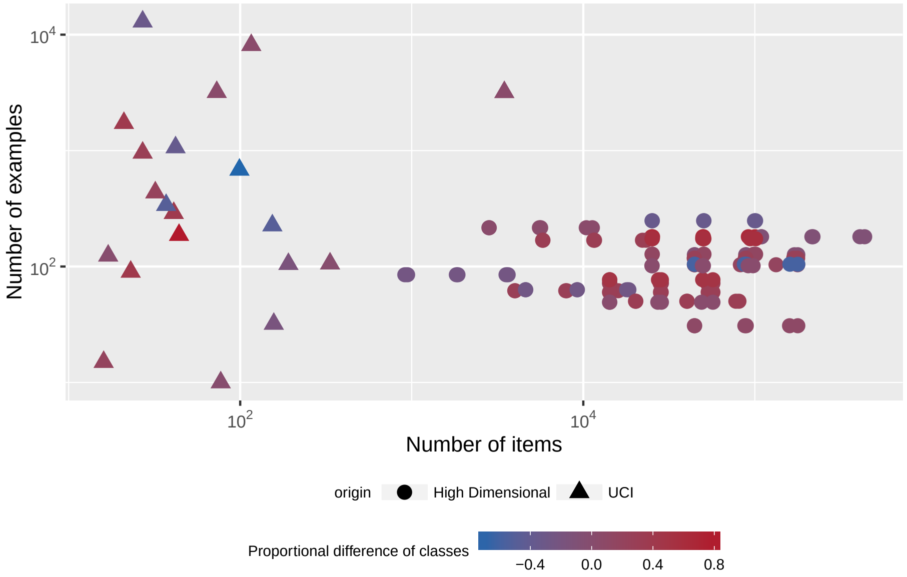

Figure 1: Visual summary of data sets used in experiments to assess the performance of SSDP. Data sets in the high dimensional group (Table 4) are represented by bullets, while UCI (Table 5) representatives are marked by triangles. The color of the points represent the proportional difference between the number of positive and negative examples in the data. Red represents data set with proportionally more positive examples than negative, while blue represents the opposite.

used in our experiments. As we can notice, the data sets are well distributed in the item-example space. The high dimensionality bases have a wide variation in the number of items, but all of them have a small number of examples. On the other hand, the UCI bases show higher variation in the number of examples, but a small number of items. We can also notice that the majority of bases have proportionally the same amount of positive and negative examples, with a slight tendency to have more positive than negative examples. However, we also see some bases where there is an imbalance between the number of positive and negative examples; this occurs for both the UCI and high dimensional bases.

We conducted four types of experiments in this work to assess the performance of SSDP from different perspectives. In our first batch of experiments (subsection 2.6.1), we evaluated different parameter settings for SSDP. We tested the algorithm with different crossover and mutation rates (sometimes fixed during the entire execution, opposite to the original version, which auto-adjust these parameters) and two different stopping criteria. These experiments will help us elucidate whether the choices made for the original version (section 2.5) are indeed good ones. Defined the best parameters for SSDP, we compared the effectiveness of the algorithm against SD and random search (subsection 2.6.2), and also against the state of the art evolutionary algorithms (subsection 2.6.3). Finally, in subsection 2.6.4 we evaluated the SSDP in relation to D^+ coverage and in the redundancy among the returned DPs.

Statistical analysis of the results was performed by using the hypothesis tests *Wilcoxon* and *Friedman*. The *Wilcoxon* is a non-parametric test that has been indicated and used for performance analysis between two algorithms. *Friedman* test (Friedman, 1940) is commonly indicated to assess whether there is statistical difference between more than two algorithms (Demšar, 2006).

When the *Friedman* test rejected the null hypothesis, the next step is to perform another hypothesis test to validate which one or which algorithms are standing out from the others. One of the options is the test with controller. Controller is the baseline algorithm that will be compared to all the others. This method is commonly used when a new algorithm is proposed and researchers needs to compare its performance with other existing methods in the literature (Demšar, 2006). The *Friedman* test statistic with controller is given by:

$$z = \frac{(R_i - R_j)}{\sqrt{\frac{k(k+1)}{6N}}},$$

where R_i-R_j is the mean ranking difference between two algorithms, N is the number of databases and k is the number of algorithms.

The value of z is calculated between the control algorithm and the other algorithms generating $k-1$ values. The respective *p-values* are calculated from the values of z. The null hypothesis is rejected when $p < \alpha$. However, the α must be adjusted for multiple comparisons. We adjusted the significance values with the *Holm* (Holm, 1979) methodology, where the value of α is adapted by the equation $\frac{\alpha}{k-1}$, k being the number of algorithms. Hypothesis tests were done using the implementation available in (of Granada Research Group, 2016).

As discussed above, we divided the experiments in four sections. Section 2.6.1 aims to test variations of the SSDP and experimentally validate some characteristics of the model in the approach of high dimensional bases. Furthermore, Section 2.6.1 shows the convergence and behavior of the SSDP in each generation for the largest database used in our experiments. Section 2.6.2 confronts SSDP with an approach based on beam search, a random search approach and another one based on a trivial search in high dimensionality bases. Next, Section 2.6.3 confronts the SSDP with three evolutionary approaches in traditional and high dimensional databases. Finally, in Section 2.6.4 we confront SSDP with three evolutionary approaches in relation to D^+ coverage and redundancy.

2.6.1 Assessing the impact of different parameter settings on SSDP's performance

In this section we first tested different configurations of the SSDP algorithm. We tried different mutation and crossover rates: 100-100, 10-90, 50-50 and *Auto* (section 2.5). We also attempted to use different stopping criteria: (1) stop when there is no change in the top-k patterns for 3 generations, referred to as *(3x)*; and (2) reset the population when the algorithm stops because of the first criterion, and halt the execution when it reaches the first criterion for the third time, referred to as *(3x3)*. Thus, eight different versions/configurations of SSDP were tested,

Table 6: Summary of the eight SSDP versions tested.

Version	Crossover rate	Mutation rate	Stop criterion
SSDP_Auto_3x3 (**or just SSDP**)	Auto	Auto	3x3
SSDP_90x10_3x3	90%	10%	3x3
SSDP_50x50_3x3	50%	50%	3x3
SSDP_100x100_3x3	100%	100%	3x3
SSDP_90x10	90%	10%	3x
SSDP_50x50	50%	50%	3x
SSDP_100x100	100%	100%	3x
SSDP_Auto	Auto	Auto	3x

which are summarized in Table 6. The SSDP(3x3)_Auto (Table 6) version corresponds to the final version of the SSDP, described in section 2.5. Each experiment was repeated ten times, with the metric evaluation Q_g ($g = 1$) and $k = \{5, 10, 20, 50\}$.

Figure 2 presents the average Q_g and time for the eight SSDP versions for $k = \{5, 10, 20, 50\}$. We notice that the versions with stopping criterion *(3x3)* stood out from the others with respect to the average Q_g. The improvement in quality is even more noticeable for higher values of k. We see that for $k = 50$ the use of the second stopping criterion yields an improvement of near 10%. We also notice that the original version of SSDP discussed in section 2.5 and SSDP_100x100_3x3 present similar performances for all values of k. Nevertheless, we also observe in the figure that the latter has the highest average time for all values of k. In terms of time, we observe that the choice of the second stopping criterion roughly doubles computing time of all algorithms.

In order to evaluate whether the visual difference observed in Figure 2 was statistically significant, we applied the *Friedman* test with a null hypothesis that there is no variation in the mean Q_g between the various configurations. Table 7 displays the average rankings of the different settings for the different values of k and their respective *p-values*. We observe that in all cases there was a statistical difference in the performance of the configurations. We also note that the SSDP_100x100_3x3 and SSDP_Auto_3x3 presented very similar performances, and there were no significant discrepancies between them with different values of k. This leads us to carry out a second test, in which we will evaluate if there is statistical difference between the best ranked and the other configurations.

This second hypothesis test confronted the configuration of best average rank in the Friedman test (Table 7 in bold) with the other configurations for each value of k ($\alpha = 0.05$). Table 8 summarizes the *p-value* and the significance level required by the *Holm* method to reject the null hypothesis. We notice that SSDP_Auto_3x3 and SSDP_100x100_3x3 versions were statistically at least as good as the others for all value of k. However, from the perspective of average quality of the patterns, there is no difference between the best two configurations. Therefore, SSDP_Auto_3x3 configuration was chosen the best because it had the lowest computing time in all experiments.

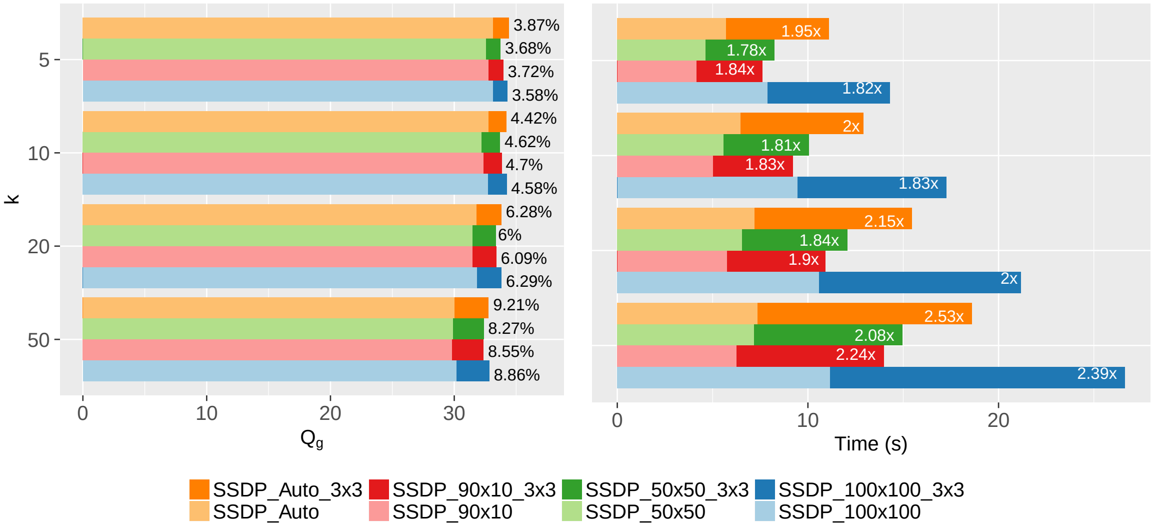

Figure 2: This figure graphically displays the average Q_g (left) and time in seconds (right) for the eight different configurations of SSDP. The algorithms were tested with the 126 microarray databases for $k = \{5, 10, 20, 50\}$. In both charts, the algorithms were grouped according to the choice of crossover and mutation rates. Dark-shaded colors represent the configurations for which population was reset twice, before the algorithm was halt. The numbers in both charts represent the increase in quality and time because of the choice to reset the population.

Table 7: *Friedman* test comparing eight versions of SSDP for $k = \{5, 10, 20, 50\}$ ($\alpha = 0.05$).

Version	**Ranking**			
	k=5	**k=10**	**k=20**	**k=50**
SSDP_100x100	4.85	5.07	5.25	5.24
SSDP_100x100_3x3	3.04	2.95	**2.80**	**2.56**
SSDP_Auto	4.82	4.98	5.19	5.70
SSDP_Auto_3x3	**2.95**	**2.91**	2.81	2.91
SSDP_90x10	5.81	5.75	5.89	6.04
SSDP_90x10_3x3	3.88	3.61	3.68	3.45
SSDP_50x50	6.33	6.38	6.32	6.40
SSDP_50x50_3x3	4.33	4.33	4.07	3.69
p-value	1.44E-10	1.48E-10	1.40E-10	1.81E-10

Table 8: Friedman hypothesis test with control algorithm for $k = \{5, 10, 20, 50\}$, $\alpha = 0.05$ (adapted by the *Holm* method).

	k=5		
Rank	**Version**	**p-value**	α **(Holm)**
7	SSDP_50x50	0.0000	0.0071
6	SSDP_90x10	0.0000	0.0083
5	SSDP_100x100	0.0000	0.0100
4	SSDP_Auto	0.0000	0.0125
3	SSDP_50x50_3x3	0.0000	0.0166
2	SSDP_90x10_3x3	0.0025	0.0250
1	**SSDP_100x100_3x3**	**0.7772**	**0.0500**
Control	**SSDP_Auto_3x3**	**-**	**-**
	k=10		
Rank	**Version**	**p-value**	α **(Holm)**
7	SSDP_50x50	0.0000	0.0071
6	SSDP_90x10	0.0000	0.0083
5	SSDP_100x100	0.0000	0.0100
4	SSDP_Auto	0.0000	0.0125
3	SSDP_50x50_3x3	0.0000	0.0166
2	SSDP_90x10_3x3	0.0236	0.0250
1	**SSDP_100x100_3x3**	**0.8976**	**0.0500**
Control	**SSDP_Auto_3x3**	**-**	**-**
	k=20		
Rank	**Version**	**p-value**	α **(Holm)**
7	SSDP_50x50	0.0000	0.0071
6	SSDP_90x10	0.0000	0.0083
5	SSDP_100x100	0.0000	0.0100
4	SSDP_Auto	0.0000	0.0125
3	SSDP_50x50_3x3	0.0000	0.0166
2	SSDP_90x10_3x3	0.0043	0.0250
1	**SSDP_Auto_3x3**	**0.9692**	**0.0500**
Control	**SSDP_100x100_3x3**	**-**	**-**
	k=50		
Rank	**Version**	**p-value**	α **(Holm)**
7	SSDP_50x50	0.0000	0.0071
6	SSDP_90x10	0.0000	0.0083
5	SSDP_Auto	0.0000	0.0100
4	SSDP_100x100	0.0000	0.0125
3	SSDP_50x50_3x3	0.0002	0.0166
2	SSDP_90x10_3x3	0.0038	0.0250
1	**SSDP_Auto_3x3**	**0.2578**	**0.0500**
Control	**SSDP_100x100_3x3**	**-**	**-**

Table 9: Average Q_g obtained by SSDP using different initialization alternatives, where zero means that the algorithm did not return any valid solution, and "–" means there was a memory exhaustion (12GB limit).

Base	**|I|**	**original**	**0.1%**	**1%**	**5%**	**10%**
alon	4,000	24.1	24.1	**0**	**0**	**0**
gravier	5,860	46.6	45.9	**0**	**0**	**0**
tian	25,250	56.2	54.1	**0**	**0**	–
yeoh	25,300	76.2	75.6	**0**	**0**	–
sun	109,226	57.2	**0**	–	–	–

Regarding the initial population, we conducted experiments to compare the method used in SSDP and populations randomly generated with individuals of predetermined sizes equal to 0.1%, 1%, 5% and 10% of $|I|$. Table 9 presents the average Q_g obtained by using the different population sizes for five discretized data sets with two intervals. In this table *original* represents the method used by the SSDP, zero means that the algorithm did not return any valid solution, and "–" means there was a memory exhaustion (12GB limit). As we can see, the larger the size of the randomly generated initial population, the more difficult it is for SSDP to converge to valid solutions. This happens because large individuals randomly generated tend to not represent a valid solution as they do not cover any example. Besides, in an index representation such as SSDP, the average size of individuals has a strong impact on memory consumption. In this context, the strategy of initializing searches with one dimension individuals, besides helping in the convergence of the model, reduces memory consumption compared to the other tested options.

Finally, we ran experiments to evaluate SSDP's convergence. Figure 3a shows the values of average fitness of populations P and P_k for each generation of the model, applied to the *sun* database, for $k = 50$. The accentuated evolution of the fitness shows the SSDP's capacity to quickly converge. The points of strong fall in the average fitness of P are the moments in which the population is reset. We see that, for this example, the first reset of P was successful, since the algorithm continued to improve the population P_k for several times in sequence. We also observed that, at some moments, the average fitness of the population P is above the population P_k. This indicates that P has many duplicated high quality individuals. This duplicated is tackled in SSDP principal by mutation and reset operator, when the population P is recreated.

Figure 3b shows the evolution of DPs average size in populations P and P_k. In the first generation P and P_k are composed strictly by singletons (DPs of size one). After that, poor quality patterns are replaced by better quality ones. We can observe from the average size of P that SSDP tends to initially direct the searches towards larger dimensions but it may also change direction to smaller dimensions if required.

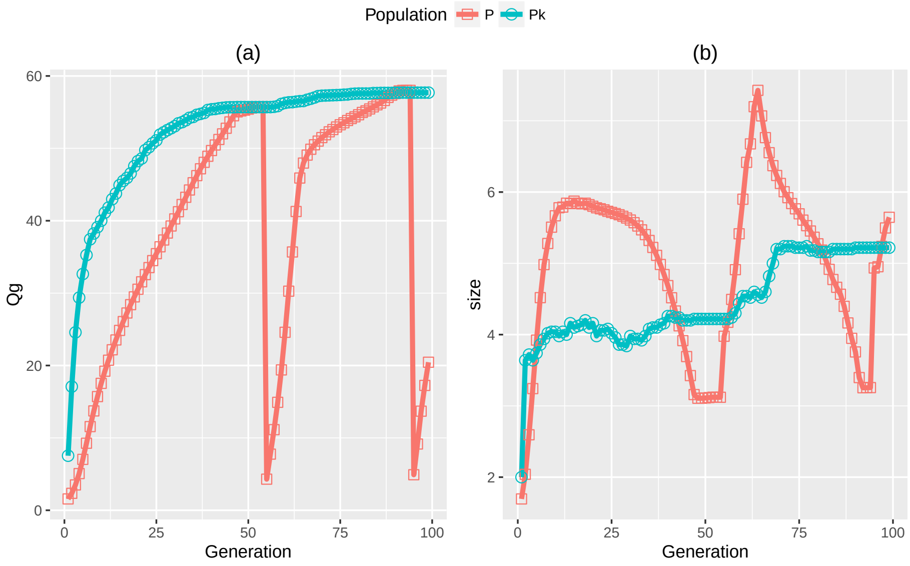

Figure 3: This figure depicts the evolution of fitness (left) and average size (right) of the populations P and P_k for each generation of SSDP for $k = 50$ with the data set *sun*.

2.6.2 Comparing SSDP to beam search

This section aims to confront SSDP with a heuristic approach based on beam search and validate it as a heuristic for discriminative pattern mining from high dimensional data. SSDP was compared with the approaches described below, all implemented in *Java*. Each experiment was repeated 10 times, with the objective function Q_g ($g = 1$) and $k = \{5, 10, 20, 50\}$.

- Random3M: three million DPs up to four dimensions randomly generated. The objective of this experiment is to compare SSDP to a random search.
- Trivial: DPs with highest fitness among all combinations of up to four dimensions, but using only the best k items. The purpose of this comparison is to validate SSDP's ability to find non-trivial DPs.
- SD: The aim is to confront SSDP with a competitive beam search heuristic. SD used the following parameters: $beamWidth = k$ and $minimumSupport = \frac{\sqrt{|D^+|}}{|D|}$ (this parameter was set according to the recommendations of Gamberger & Lavrac (2002)).

Figure 4 shows the mean Q_g and time of the SSDP, SD, Trivial and Random3M approaches for $k = \{5, 10, 20, 50\}$. We notice that, while the beam search heuristic SD is very similar to random search, our evolutionary heuristic SSDP found higher quality patterns for all values of k. In fact, we notice that SSDP achieved roughly 50% higher quality than the random search (Trivial and Random3M) and 30% higher than SD. On the other hand, SD achieved only 15% improvement compared to random search. In terms of time, we observe a linear growth in computing time for SD and a sub-linear growth for SSDP. Interestingly, despite requiring

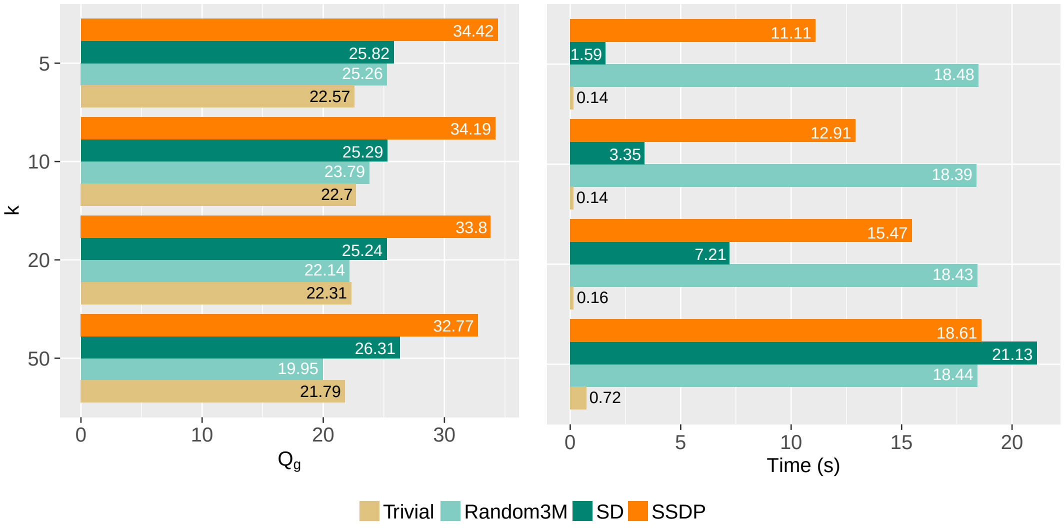

Figure 4: This figure graphically displays the average Q_g (left) and time in seconds (right) for SSDP, SD, Trivial and Random3M. The algorithms were tested with the 126 microarray databases for $k = \{5, 10, 20, 50\}$. The numbers in both charts represent the Q_g and time of the algorithm.

15% more computing time to find the top 50 patterns than SSDP, SD did not find higher quality patterns.

We applied the *Wilcoxon test* to verify whether the difference of performances between SSDP and SD was statistically significant. The null hypotheses that SSDP performs equally well to SD for the different k values were all rejected for a level of significance $\alpha = 0.01$. The *p-values* obtained for $k = \{5, 10, 20, 50\}$ were respectively 6.17E-16, 4.59E-16, 5.67E-16 and 9.50E-14. Thus, SSDP was statistically superior to SD in the context of high dimensionality for $k = \{5, 10, 20, 50\}$. Furthermore, the SSDP's superiority over Random3M and Trivial approaches validate the proposed model in relation to random search and the ability to find non-trivial patterns.

The exact algorithm based on *Beam Search SDMap* (Atzmueller & Puppe, 2006) has been tested as well, using the available implementation on software *KEEL* (Alcalá-Fdez *et al.*, 2009) with default parameters. However, the algorithm had problems with memory exhaustion (12G limit) and processing time (2 hours time limit) in the tested high-dimensional databases (Table 4). With respect to traditional databases (Table 5), *SDMap* converged to valid results on only four of the 20 bases tested.

2.6.3 Comparing SSDP to other evolutionary approaches

This section compares SSDP to other evolutionary approaches using both traditional and high dimensional data sets. SSDP was confronted with SDIGA (del Jesus *et al.*, 2007b), MESDIF (del Jesus *et al.*, 2007a) and NMEEF (Carmona *et al.*, 2010), algorithms available in the KEEL machine learning suite (Alcalá-Fdez *et al.*, 2009). Default parameters in KEEL were

Table 10: $WRAcc_{normalized}$, time, number of DPs (k), average size and rank obtained by MESDIF, NMEEF, SDIGA and SSDP algorithms with 20 UCI data sets.

Algorithm	$WRAcc_{normalized}$	**time(s)**	k	**size**	**Avg. Rank**
MESDIF	0.080	2.85	3	15.06	3.4
NMEEF	**0.412**	2.55	8.1	3.02	**1.775**
SDIGA	0.188	5.70	2.6	1.28	3.025
SSDP	0.376	**0.17**	5	2.25	1.8
				p-value for the Friedman test	1.39E-05

Table 11: *Friedman* hypothesis test with control algorithm for MESDIF, SDIGA, NMEEF and SSDP algorithms ($\alpha = 0.05$)

Rank	**Algorithm**	**p**	**Holm** ($\alpha = 0.05$)
3	MESDIF	0.00006	0.016
2	SDIGA	0.0021	0.025
1	**SSDP**	**0.9511**	**0.05**
Control	**NMEEF**	–	–

used for the algorithms, and $k = 5$ and *WRAcc* as fitness function for the SSDP. We changed the choice of the fitness function for SSDP because all other algorithms use *WRAcc* as their fitness function.

Table 10 shows the mean $WRAcc_{normalized}$ (section 2.3), time, number of DPs (k) and size for the algorithms tested with the 20 UCI data sets (Table 5). We notice that NMEEF and SSDP were the best performing algorithms regarding the average $WRAcc_{normalized}$. Regarding computing time, SSDP proved to be faster than the others (it takes only a tenth of the time required by NMEEF). The table also shows the result for the *Friedman* hypothesis test considering the $WRAcc_{normalized}$. The test showed that there was a statistical difference between the algorithms (*p-value*=$1.39E-05$), with NMEEF as the best ranked algorithm, followed closely by SSDP. Then, Table 11 shows the result of the multiple *Friedman* using the NMEEF as control and *Holm* method for correcting the significance levels (α). The test showed that NMEEF was statistically better than MESDIF and SDIGA, but there is no evidence regarding SSDP. This confirms SSDP as a competitive approach also for traditional data without the need to adjust any parameters.

On the other hand, the experiments in high dimensional bases were limited to 10 of the bases described in Table 4 due to the high computational cost of the simulations. Table 12 presents the average $WRAcc_{normalized}$, time, number of DPs and size obtained by the evolutionary algorithms. In initial experiments SDIGA had difficult to converging in less than three hours with several databases and was excluded from the comparison. We notice however that the other two algorithms, NMEEF and MESDIF, did not converge to valid solutions, despite using more computational resources than SSDP.

We performed a last experiment to compare NMEEF to SSDP. In this experiment we set a population of $1,000$ individuals and $1,000,000$ evaluations (NMEEF-1k-1M). Table 13 shows the $WRAcc_{normalized}$, number of DPs (k), time and number of tests did by the SSDP and

Table 12: $WRAcc_{normalized}$, time, number of DPs (k) and average size obtained by the MESDIF, NMEEF and SSDP algorithms in ten microarray databases.

Algorithm	$WRAcc_{normalized}$	time(s)	k	size
MESDIF	0.0039	1,241.4	3	16,458.03
NMEEF	0.0115	2,069.2	10.1	56.87
SSDP	**0.6304**	7.7	5	2.42

Table 13: WRAcc, *k*, time, number of tests and patterns obtained by SSDP and NMEEF-1k-1M algorithms in ten microarray databases.

Base	Algorithm							
	NMEEF-1k-1M				SSDP			
	$WRAcc_{norm}$	k	Time(s)	Tests (10^6)	$WRAcc_{norm}$	k	Time(s)	Tests (10^6)
alon	0.26	3	1,984	1	0.572	5	0.422	0.116
burczynski	0	0	63,697	1	0.684	5	8.254	1.247
chiaretti	0	0	36,882	1	0.584	5	4.789	0.808
chin	0.388	1	31,301	1	0.624	5	5.928	0.799
christensen	0.592	1	3,408	1	0.896	5	0.297	0.056
gravier	0.232	1	5,745	1	0.440	5	0.905	0.185
nakayama	0	0	58,419	1	0.600	5	7.893	1.515
tian	0	0	43,218	1	0.296	5	4.072	0.505
yeoh	0	0	104,533	1	0.848	5	8.798	0.782
sun	–	–	–	–	0.186	5	36.315	3.386

NMEEF-1k-1M for the tested databases, where "–" means that the algorithm did not finish in less than 48 hours. We observe that NMEEF (the best performing evolutionary algorithm for traditional data) still did not return any valid discriminative pattern in six of the ten tested data sets. On the other bases, the average $WRAcc_{normalized}$ was lower than those obtained by SSDP. In addition, NMEEF's computing time was considerably higher than SSDP's for all bases.

We conclude from these experiments that the evolutionary models NMEEF, MESDIF and SDIGA are not suitable to high dimensionality, even if the parameters are tuned (in the case of NMEEF). Moreover, these algorithms proved to be costly in terms of processing time and returned poor results. SSDP, on the other hand, obtained valid DPs for all high dimensional data sets using considerably lower processing time.

2.6.4 Redundancy and coverage in SSDP

This section aims to evaluate SSDP in relation to D^+ coverage and in redundancy between top-k DP set. The experiments were made in UCI data sets (Table 5), for $k = 5$ and $WRAcc$ as metric evaluation. SSDP was confronted with the evolutionary algorithms NMEEF, MESDIF and SDIGA using default parameters.

The coverage in relation to D^+ was evaluated by overall support ($SUPP^+$, Equation 2.3). The algorithms SSDP, NMEEF, SDIGA and MESDIF obtained respectively 86.2%, 89.1%, 86.6% and 37.6% as mean $SUPP^+$ (Table 14). In this way, SSDP was competitive in relation to

Table 14: Local mean support ($supp^{+}_{mean}$) and mean overall support ($SUPP^{+}$, Equation 2.3) for algorithms SSDP, NMEEF, MESDIF and SDIGA for 20 UCI databases (Table 5), where $supp^{+}_{mean} = \frac{1}{k}\sum supp^{+}(dp)$, $supp^{+}(dp) = \frac{|c^{+}(dp)|}{|D^{+}|}$ and $SUPP^{+} = \frac{|c^{+}(dp_1) \cup ... \cup c^{+}(dp_k)|}{|D^{+}|}$.

Algorithm	**mean** $SUPP^{+}$	**mean** $supp{+}_{mean}$	mean $SUPP^{+}$ - mean $supp{+}_{mean}$
SSDP	0.862	0.582	0.28
NMEEF	0.891	0.766	0.125
MESDIF	0.376	0.222	0.154
SDIGA	0.866	0.676	0.19

NMEEF and SDIGA and superior to MESDIF.

Already the coverage redundancy was evaluated by difference between the mean of overall support $SUPP^{+}$ and mean local support ($supp^{+}_{mean}$), where $supp^{+}_{mean} = \frac{1}{k}\sum supp^{+}(dp)$, $supp^{+}(dp) = \frac{|c^{+}(dp)|}{|D^{+}|}$ and $SUPP^{+} = \frac{|c^{+}(dp_1) \cup ... \cup c^{+}(dp_k)|}{|D^{+}|}$. In this way, $supp^{+}_{mean} \approx SUPP^{+}$ indicates that a DP set was restricted to describing approximately the same examples of D^{+}. Thus, Table 14 shows that SSDP generated less average coverage redundancy than NMEEF, MESDIF and SDIGA.

Table 15 shows the mean local support ($supp^{+}_{mean}$) and the overall support ($SUPP^{+}$) of top-5 DPs returned by SSDP in each UCI database. In this way, Table 15 shows that SSDP covered more than 60% of D^{+} in 19 of 20 databases and more than 90% in seven of them. The difference between $SUPP^{+}$ e $supp^{+}_{mean}$ also shows that DP set returned by SSDP usually were not restricted to cover the same examples of D^{+}.

Finally, description redundancy was analyzed by counting the number of databases where all returned DPs have a common item. This kind of redundancy occurred in SSDP, NMEEF, SDIGA and MESDIF in respectively 6, 8, 12 and 17 of 20 databases. Thus, all algorithms presented significant description redundancy, but with less frequency in SSDP.

So, in these experiments we conclude that the SSDP was more efficient than the NMEEF, MESDIF and SDIGA using default parameters in relation to redundancy between returned DP set. But the proposed model presented some difficulties. The covered in relation to D^{+} was unstable, ranging between 100% and 58.4%. Although the description redundancy was more critical, presenting in 6 out of 20 databases. Thus, we believe that the proposed model still deals a little inefficiently with the redundancy problem.

Some contents like SSDP implementation, some high dimensionality databases used in the tests, tables with the results of each experiment of this paper, including other evaluation metrics such as support, confidence level, TP (true positive), FP (false positive) and *p-value* are available on this website (`https://github.com/tarcisiodpl/ssdp`).

2.7 CONCLUSION

This paper presents SSDP, the first evolutionary approach for mining top-*k* discriminative patterns in high dimensional data sets. Extraction of discriminating information from high

Table 15: Mean local support ($supp^{+}_{mean}$) and the overall support ($SUPP^{+}$) obtained by the SSDP in 20 UCI databases (Table 5), for $k = 5$ and $WRAcc$ as metric evaluation, where $supp^{+}_{mean} = \frac{1}{k}\sum supp^{+}(dp)$, $supp^{+}(dp) = \frac{|c^{+}(dp)|}{|D^{+}|}$ and $SUPP^{+} = \frac{|c^{+}(dp_1)\cup \ldots \cup c^{+}(dp_k)|}{|D^{+}|}$.

Database	$supp^{+}_{mean}$	$SUPP^{+}$	$SUPP^{+} - supp^{+}_{mean}$
audiology	0.947	0.94	0.007
breast-cancer	0.86	0.767	0.093
bridges-version2	0.977	0.7	0.277
car	0.99	0.383	0.607
kr-vs-kp	0.715	0.715	0
lung-cancer	0.846	0.615	0.231
molecular-biology-promoters	0.924	0.339	0.585
monks-problems-1-train	0.661	0.316	0.345
mushroom	0.977	0.779	0.198
nursery	1	0.44	0.56
postoperative-patient-data	0.609	0.281	0.328
primary-tumor	0.619	0.59	0.029
shuttle-landing-control	1	0.666	0.334
solar-flare-2	1	0.721	0.279
soybean	0.978	0.939	0.039
spect-test	0.738	0.454	0.284
splice	0.807	0.249	0.558
tic-tac-toe	0.584	0.3	0.284
trains	1	0.56	0.44
vote	0.947	0.847	0.1

dimensional data is a common challenge in areas such as bioinformatics and text mining. Evolutionary approaches have been shown to be an efficient option for mining discriminative patterns in traditional data sets. However, none of them were developed with a focus on high dimensionality.

SSDP has been designed from the beginning to the context of high dimensionality bases. The representation of individuals, for example, only considers the items used by DPs as a way to reduce the computational cost of memory. Our approach for generating the initial population seeks to increase the convergence of the algorithm and ensure that all items are considered in the search. At last, SSDP controls redundant individuals in the top-k DPs as a way to reduce the computational cost and increase the relevance of patterns.

The proposed model also seeks to hide some parameters from the user to become a simpler approach to apply and consequently help in the popularization of discriminative knowledge extraction. Population size, mutation and crossover rates are automatically defined by the algorithm. The stopping criteria is not defined by the number of tests or generations and has been developed to be kept clear for the final user.

The SSDP had some of its characteristics experimentally validated, such as mutation, stopping criteria and initial population. Its performance was assessed using high dimensional and traditional data sets. The algorithm was also compared with other approaches: random, trivial, based on beam search and based on evolutionary computing. In the context of high dimensionality, SSDP obtained statistically better results than all the others algorithms, in relation to the quality of DPs. In traditional databases, SSDP was shown to be a competitive approach without the necessity to make any adjustments in parameters. Finally, in relation to redundancy in DP set, SSDP was better than other evolutionary approaches, but it presented some problems.

Therefore, we concluded that SSDP is an efficient, flexible and simple alternative for the extraction of discriminant knowledge in high dimensional data sets. However, SSDP is the only evolutionary approach that deals exclusively with discrete data. Besides that, the model has few resources to deal with redundancy in top-*k* DPs, restricting itself to eliminating DPs equal or dominated by others. This opens new pathways in the direction of evolving SSDP to deal with numerical data and to find more efficient alternatives to eliminate redundancy.

3 SSDP+: SIMPLE SEARCH DISCRIMINATIVE PATTERNS PLUS

Este capítulo é uma reprodução completa do artigo *SSDP+: a Diverse and More Informative Subgroup Discovery Approach for High Dimensional Data*, publicado em 2018 no congresso internacional *IEEE Congress on Evolutionary Computation* (Lucas *et al.*, 2018).

3.1 ABSTRACT

This paper presents an evolutionary approach for mining diverse and more informative subgroups focused on high dimensional data sets. Subgroup Discovery (SD) is an important tool for knowledge discovery that aims to identify sets of features that distinguish a target group from the others (e.g. successful from unsuccessful treatments). At the same time, to extract information from high dimensional data sets becomes more natural. One of the first and most efficient SD heuristics focused on high dimensional data is the SSDP. However, this model deals superficially with diverse/redundancy in top-k subgroups, which can result in poor information for users. This work presents SSDP+, an extension of the SSDP model to provide diversity in a way that explore the relation between subgroups order to generate a more informative set of patterns.

3.2 INTRODUCTION

We introduce in this paper a new evolutionary approach for mining diverse and more informative subgroups on high dimensional data sets. Subgroup Discovery (SD) is a data mining task that has the objective of identifying sets of items (or features) that distinguish a target group from the others, for example: successful from unsuccessful treatments, unhealthy from healthy cells, or even positive from negative sentiments in sentiment analysis (Herrera *et al.*, 2011). There are many SD applications reported in the literature in different domains such as: medicine (Carmona *et al.*, 2013, 2011), bioinformatics (Li & Wong, 2002; Quackenbush, 2001), marketing (Carmona *et al.*, 2012; del Jesus *et al.*, 2007b), e-learning (Romero *et al.*, 2009) and traffic accidents (Kavšek & Lavrac, 2004; Kavšek *et al.*, 2002).

At the same time, with the increasing volume of information stored in the world (*30G/s* only on the internet), it becomes more common to extract knowledge from high dimensionality data sets. Some important areas, such as *bioinformatics* and *text mining*, usually deal with data sets with tens of thousands of attributes (Kotzias *et al.*, 2015; Ramey, 2016).

There are many heuristic for mining subgroups (Liu *et al.*, 2015; Helal, 2016; Atzmueller, 2015; Herrera *et al.*, 2011). Although these algorithms have good performance with low dimensional data, Lucas et al. (Lucas *et al.*, 2017) showed that they do not hold the same performance with high dimensional data. Thus, they introduced SSDP, one of the most efficient and simple to use heuristic for mining subgroups in high dimensional data. SSDP, though, only deals superficially with diversity/redundancy in top-k subgroups (Lucas *et al.*, 2017). This is a critical limitation because little diversity can result in poor information for users (Bosc *et al.*, 2017; Van Leeuwen & Knobbe, 2012).

There are different approaches to increase diversity of subgroups. The most common way for promoting diversity in subgroups is by assigning weights to the examples in order to penalize similar subgroups in relation to covered examples (Carmona *et al.*, 2010; Gamberger & Lavrac, 2002; Van Leeuwen & Knobbe, 2012; Lavrač *et al.*, 2004; Rodríguez *et al.*, 2012). However, this penalty may not be sufficient to avoid the existence of two or more subgroups covering the same examples and the user do not know which subgroups are similar to each other. Besides that, subgroups considered redundant may represent relevant information, as a more feasible way to solve a problem or an new knowledge. So, the difference between redundancy and relevant information can be associated with the problem domain. Thus, the diversification process usually can result in the discard of relevant information.

This work presents the SSDP+, an improved version of the SSDP algorithm that tackles the problem of redundancy/diversity problem in a flexible way, reducing the risk of loss of relevant information and generating a more informative set of rules for users. We assess the performance of SSDP+ by comparing it with other competitive algorithms. We use in our experiments real-world high dimensional data sets that come from three different domains (bioinformatics, text mining and the humanities/social sciences). We also conducted experiments to highlight SSDP's limitation regarding diversity, and show how SSDP+ solves this issue.

The remainder of this manuscript is organized as follows. We formalize the problem of Subgroup Discovery in Section 3.3. In Section 3.4 we present the proposed model. Next, Section 3.5 shows the experiments and Section 3.6 the conclusions.

3.3 SUBGROUP DISCOVERY PROBLEM FOR DISCRETE DATA SETS

Let D be a labeled data set with a set A of categorical/discrete attributes. According to the class label, the set of samples from D can be partitioned into $D^+ = \{e_1^+, e_2^+, ..., e_{|D^+|}^+\}$ and $D^- = \{e_1^-, e_2^-, ..., e_{|D^-|}^-\}$, respectively the positive (target) examples and the remaining (negative examples). Let $dom(A_j)$ be the domain of values for attribute $A_j \in A$. We call features or items the set of all pairs $(attribute, value)$, that is $I = \bigcup A_j \times dom(A_j) = \{i_1, i_2, ..., i_{|I|}\}$. We say that an example d has an item $i = (A_j, v) \in I$ if d has value v for the attribute A_j.

We call a *subgroup* a set $s \subseteq I$. The *size* of a subgroup s is the number of items in s, that is $size(s) = |s|$. Every s might be associated (cover) a set of positive and negative examples, which we formally define as $c^+(s) = \{d \in D^+ \mid d \text{ has all items in } s\}$, and $c^-(s) = \{d \in D^- \mid d \text{ has all items in } s\}$. The size of these two sets define the *positive* and *negative support* of a subgroup, i.e. its frequency among positive and negative examples.

Table 16 contains a toy example of data set, for which the aim is to identify the differences between successful and unsuccessful medical treatments for a given disease. In this example, Table 16 represents the data set D and $label = success$ is the target of investigation. Thus, $D^+ = \{e_1, e_2, e_3, e_4, e_5\}$ are the positive examples (where $label = success$) and $D^- = \{e_6, e_7, e_8, e_9, e_{10}\}$ are the negative examples (where $label \neq success$).

Meanwhile Table 17 represents the universe of items $I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$ and the respective positive and negative covered examples. In this context, $s = \{i_5\}$ is an interesting subgroup, once $c^+(s) = |\{e_1 e_2, e_3\}| = 3$ and $c^-(s) = |\emptyset| = 0$. On the other hand, $s' = \{i_1, i_7\}$ is not an interesting subgroup as it is equally frequent among positive and negative samples ($c^+(s') = |\{e_1\}| = 1$ and $c^-(s') = |\{e_7\}| = 1$).

Table 16: A toy example data set. In this simulated data, the target is to identify the differences between successful and unsuccessful medical treatments for a given disease.

example	genre	age	medicine	label
e_1	M	senior	B	success
e_2	F	senior	B	success
e_3	M	senior	A	success
e_4	M	adult	A	success
e_5	F	child	A	success
e_6	F	child	A	failure
e_7	M	child	B	failure
e_8	F	child	B	failure
e_9	M	adult	A	failure
e_{10}	F	adult	A	failure

The definition of the relevance/interestingness of a subgroup is given by evaluation metrics (Flach *et al.*, 1999). One of the most used metric is the weighted relative accuracy (WRAcc), given by Equation (3.1):

$$WRAcc(s) = \frac{TP+FP}{|D|}\left(\frac{TP}{TP+FP} - \frac{|D^+|}{|D|}\right), \tag{3.1}$$

where $TP = |c^+(s)|$ (the positive support) and $FP = |c^-(s)|$ (the negative support).

This evaluation metric is a trade-off between coverage ($\frac{TP+FP}{|D|}$) and relative accuracy ($\frac{TP}{TP+FP} - \frac{|D^+|}{|D|}$). *WRAcc* values range from -0.25 to $+0.25$, where $+0.25$ represents a totally

Table 17: Universe of items I and respective covered examples for the data presented in Table 16. In this table, items are in rows and examples in columns. There is a cross if an example has the corresponding item.

I	**(attribute, value)**	D^+	D^-
i_1	(genre, M)	e_1,e_3, e_4	e_7,e_9
i_2	(genre, F)	e_2,e_5	e_6,e_8,e_{10}
i_3	(age, child)	e_2,e_5	e_6,e_8
i_4	(age, adult)	e_4	e_9,e_{10}
i_5	(age, senior)	e_1,e_2,e_3	
i_6	(medicine, A)	e_3,e_4,e_5	e_6,e_9,e_{10}
i_7	(medicine, B)	e_1,e_2	e_7,e_8

pure subgroup, or from -1 to $+1$ in its normalized form ($WRAcc_{normalized} = 4 \times WRAcc$) (Flach, 2003). In this case $+1$ represents a totally pure subgroup, describing all the positive examples and none of negative examples, while -1 represents a pattern describing all the negative examples and none of the positive, and 0 indicates that there is no gain relative to the fixed rule describing all examples of the base as positive.

Another well-known evaluation metric is the Qg (Gamberger & Lavrac, 2002), given by Equation 3.2, where g is a generalization parameter (Gamberger & Lavrac, 2002). In this way, high values of g usually return subgroups more general with less precision and low values of g often return specific subgroups with high precision. The g default value is 1 (Gamberger & Lavrac, 2002).

$$Qg(s) = \frac{TP}{FP + g} \tag{3.2}$$

There are many other evaluation metrics for subgroups (Liu *et al.*, 2015; Herrera *et al.*, 2011). Choosing the best metric often depends on the problem or specialist's convictions. So, it is important that the proposed algorithms accept different options of evaluation metrics.

One of the challenges in *Subgroup Discovery* is the diversity/redundancy in top-k subgroups (Bosc *et al.*, 2017; Van Leeuwen & Knobbe, 2012). The most common kind of redundancy are in relation to coverage and description. The coverage redundancy is when there is an overlap between positive examples covered by two or more subgroups ($[c^+(s_1) \cap c^+(s_2)] \approx [c^+(s_1) \cup c^+(s_2)]$). The subgroups $s_1 = \{i_2\}$ and $s_2 = \{i_3\}$ in Table 17, for example, are redundant in relation to coverage, given that $c^+(s_1) = c^+(s_2) = \{e_2, e_5\}$. In this context, the subgroups describe only a small part of the positive examples, generating little information about the data.

A way to evaluate coverage redundancy is by global positive support metric, that measures the percentage of positive examples D^+ covered for top-k subgroups S_k. The Equation 3.3 formalizes the metric.

$$SUPP^+(S_k) = \frac{|c^+(s_1) \cup \ldots \cup c^+(s_k)|}{|D^+|}, \tag{3.3}$$

Thus, $SUPP^+$ metric can assume values between 0 and 1, where 1 means that a S_k set completely cover D^+. In this context, the lower the redundancy among the top-k subgroups S_k, the larger tends to be $SUPP^+$ value.

On the other hand, description redundancy occurs when a set of subgroups has one or more items in common. The subgroups $s_1 = \{i_1\}$, $s_2 = \{i_1, i_3\}$ and $s_3 = \{i_1, i_3, i_6\}$ in Table 17, for example, describe only male patients ($i_1 \rightarrow (genre, M)$). A simple way to evaluate description redundancy is given by the frequency of the most common item in the top-k subgroups. The Equation (3.4) formalizes the item dominator metric.

$$item_{dom}(S_k) = \frac{|i_{more\_frequenty}|}{k}, \tag{3.4}$$

where $|i_{more\_frequenty}|$ is the number of occurrences of the most frequent item in S_k. So, $item_{dom}$ can assume values between $\frac{1}{k}$ and 1, where the lower its value, the less description redundancy in top-k subgroups.

3.4 SSDP+: SIMPLE SEARCH DISCRIMINATIVE PATTERNS PLUS

SSDP+ is a top-k mono-objective subgroup mining model focused on high dimensional data sets. SSDP+ improves SSDP by: (1) dealing with redundancy problem in a flexible way; (2) reducing the risk of loss of relevant information; and (3) generating a more informative subgroup set for users.

The main difference between SSDP and SSDP+ is how the top-k subgroups are organized and stored in the search. SSDP stores the top-k best evaluated subgroups that are distinct and non-dominated. A subgroup s is considered dominated by another subgroup s' if $c^+(s) \subseteq c^+(s')$ and $c^-(s') \subseteq c^-(s)$. So, in SSDP the users do not interfere in the diversification process and all subgroups considered redundant are neglected by the algorithm.

In SSDP+, part of subgroups considered redundant is stored to provide more information for the users. So, each subgroup s in top-k has a cache $s.cache$. The size of $s.cache$ is defined for the user by the parameter kc. In this way, $s.cache$ stores the kc best subgroups considered redundant in relation to s. In SSDP+, two subgroups s and $s' \in s.cache$ are considered redundant if $sim(s, s') > min_{similarity}$, where sim is a similarity function measure and $min_{similarity}$ is a threshold defined by the user. The default similarity measure in SSDP+ is the *Jaccard* index (Choi & Cha, 2010), given by Equation (3.5). Finally, the similarity level between s and each $s' \in s.cache$ are showed for the users in order to provide more information.

$$sim_J(s, s') = \frac{|c^+(s) \cap c^+(s')|}{|c^+(s) \cup c^+(s')|}, \tag{3.5}$$

where $|c^+(s) \cap c^+(s')|$ is the number of positive examples covered by s and s' at the same time and $|c^+(s) \cup c^+(s')|$ is the number of positive examples covered by s or s'.

In this way, the cache in SSDP+ is responsible for minimizing the risk of discarding relevant information and, at the same time, providing more information for the user. In practice, a subgroup $s' \in s.cache$ can present an alternative solution in relation to s when they are tightly similar ($sim_J(s,s') \approx 1$). In this way, an alternative solution can represent, for example, other way to impact the same target audience in a company or a different group of features that influence almost the same students to abandon a school. A cache $s.cache$ also can present subgroups weakly similar in relation to s when the $min_{similarity}$ between s and $s.cache$ is small, but they also can represent relevant information that would be discard as consequence of diversification process.

The diversity in SSDP+ is generated as follows. Let be P_k be the vector that stores the top-k subgroups and P be a population of candidates to P_k, where P and P_k are sorted by evaluation metric (0 index as the best one). So, P_k starts with subgroups without items $s_\phi = \{\}$ where $c^+(s_\phi) = D^+$ and $c^-(s_\phi) = D^-$. Then, the best subgroup of P is stored in the first position of P_k. After that, the second position of P_k will only be occupied by a subgroup that is not considered similar to $P_k[0]$. Next, the third position of P_k will only be occupied by a subgroup that is neither similar to $P_k[0]$ or $P_k[1]$, and so on, until P_k is filled up. Thus, the smaller the value of $min_{similarity}$ the greater the diversity between top-k subgroups.

Algorithm 2 describes in details how a candidate subgroup s_c update P_k for a given $min_{similarity}$ value. This algorithm assumes that s_c is better evaluated than the worst subgroup of P_k, that is sorted with $P_k[0]$ as the best evaluated subgroup. So, let $P_k[i]$ be the best evaluated subgroups of P_k where $sim(s_c, P_k[i]) > min_{similarity}$ (lines 1-2). If s_c is better evaluated than $P_k[i]$ (or equal with small size), it subscribes $P_k[i]$ and $P_k[i].cache$ is again considered candidates to P_k (lines 3-13). Otherwise, s_c can be included in $P_k[i].cache$ or discard (lines 14-15). The s_c is discard when $P_k[i].cache$ is complete filled with subgroups better evaluated than s_c or if it is replayed. Finally, if s_c was not similar to any subgroup in P_k, it subscribes the worst evaluated subgroup of P_k (lines 17-20).

The others SSDP+ features are similar to the original SSDP algorithm. Each individual of the genetic algorithm contains only the items that compose a subgroup. The initial population consists of all possible subgroups of one dimension and the selection is made by binary tournament.

The mutation have three possibilities with the same probability: (1) a random item is added to the subgroup (e.g. $s = \{a,b,c\} \rightarrow s' = \{a,b,c,d\}$); (2) a random item is replaced by another (e.g. $s = \{a,b,c\} \rightarrow s' = \{a,b,d\}$); and (3) a random item is removed from the subgroup (e.g. s=$\{a,b,c\} \rightarrow s' = \{a,b\}$). In crossover two individuals generate two new by uniform crossover with 50% mixing ratio (e.g. $s_1 = \{a,b\}$ and $s_2 = \{c,d\} \rightarrow s'_1 = \{a,d\}$ and $s'_2 = \{b,c\}$). However, since the initial population consists of individuals with only one item, the crossover in the first generation is done by joining the parent items in a single child (e.g. $s_1 = \{a\}$ and $s_2 = \{b\} \rightarrow s' = \{a,b\}$).

The mutation and crossover rates are defined dynamically. The mutation rate increases and crossover rate decreases when there are no improvements in top-k subgroups. Otherwise the

Algorithm 2 $updateTopK(s_c, P_k, min_{similarity})$

```
for $i \leftarrow 0$ to $k-1$ do
    if $sim(s_c, P_k[i]) > min_{similarity}$ then
        if $[evaluation(s_c) > evaluation(P_k[i])] \vee [evaluation(s_c) == evaluation(P_k[i]) \wedge size(s_c) < size(P_k[i])]$ then
            $cache_{temporary} \leftarrow P_k[i].cache$
            $s_{new} \leftarrow s_c$
            $s_{new}.cache.ADD(P_k[i])$
            $P_k[i] \leftarrow s_{new}$
            $sort(P_k)$, where $P_k[0]$ is the best one
            for all $s \in cache_{temorary}$ do
                $updateTopK(s, P_k, min_{similarity})$
            end for
            $break$
        else
            $P_k[i].cache.ADD(s_c)$
            $break$
        end if
    else if $i == k-1$ then
        $P_k[k-1] \leftarrow s_c$
        $P_k[k-1].cache \leftarrow \{\}$
        $sort(P_k)$, where $P_k[0]$ is the best one
    end if
end for
return $P_k$
```

crossover increases and mutation decreases. The objective is to deepen the search (by crossover) when the algorithm is in a promising region (improvement in the top-k subgroups) and widen it (by mutation) otherwise. The changes in the rates are 0.2. The initial values are 0.6 for crossover and 0.4 for mutation. The sum of mutation and crossover rates is 1 and both can assume values between 0 and 1. This methodology is an adaptation of the methodology proposed by (Luna *et al.*, 2014).

The SSDP+ uses as stopping criterion the stabilization of the group of top-k subgroup after the population has been reset twice. A population is reset when there is no change in the caches of top-k subgroups for three consecutive generations and the mutation rate is equal to one. In this process the algorithm randomly generates individuals of fixed size between two and the average size of the top-k subgroup. Moreover, 10% of individuals are generated using exclusively items present in top-k subgroup.

Algorithm 3 contains the pseudocode of SSDP+. The algorithm starts by generating the initial population with all possible subgroups $s \in I | size(s) == 1$ (line 1). Then, the top-k subgroups and respective caches are filled with empty subgroups (lines 2-7). In lines 8 to 10 the candidate subgroups of P are assigned to P_k using the function $updateTopK(P_k, P[i], min_{simirarity})$, described in Algorithm 2. Lines 11 to 13 initialize the variable that controls the number of restarts and mutation and crossover rates. Next, between lines 15 and 22, the genetic algorithm performs the search by generating new subgroups candidates to P_k and updating P_k in each generation. The genetic algorithm converges when there is no change in P_k and the respective caches for three consecutive generations having $mutationRate == 1.0$. Population P restarts twice (lines 14,23-25). Finally, in the line 26 the top-k subgroups and respective caches are returned to the user.

The algorithm was implemented in *Java* and is available at `https://github.com/tarcisiodpl/ssdp_plus`. The implementation allows the user to filter attributes, values, or items in a simple way. It allows users, for example, discard non-determined values (*NA*) or some irrelevant attributes or items without modifying the data set.

3.5 EXPERIMENTS

We use in our experiments real-world high dimensional data sets that come from three different domains, described in Table 18. The first group started from 20 *bioinformatics* data sets, available in the package *datamicroarray* (Ramey, 2016) from R software. In this group, the majority class was considered the target and 50% of attributes with lowest variance were removed. Table 18 describes the *bioinformatics* data sets after this process. Finally, each data set was discretized in relation to frequency and width with 2, 4 and 8 bins, and with respect to quartiles, with three bins (less than Q_2, between Q_2 and Q_3 and greater than Q_3). So, seven versions of each *bioinformatics* data were generated, totaling 140 data sets.

The second group consists of six *text mining* data sets, three about sentiment analy-

Algorithm 3 SSDP+ pseudocode

```
Require: $k$, $metricEvaluation$, $kc$, $min_{similarity}$
$P \leftarrow \{\{i_1\},\{i_2\},...,\{i_{|I|}\}\}$
for $i \leftarrow 0$ to $k$ do
    $P_k[i] \leftarrow \{\}$
    for $j \leftarrow 0$ to $kc$ do
        $P_k[i].cache.ADD(\{\})$
    end for
end for
for all $P[i] \in P|evaluation(P[i]) > evaluation(P_k[k-1])$ do
    $updateTopK(P_k, P[i], min_{similarity})$
end for
$reinializationCount \leftarrow 0$
$mutationRate \leftarrow 0.4$
$crossoverRate \leftarrow 0.6$
while $reinializationCount < 2$ do
    while $P_k$ not improve three consecutive generations keeping $mutationRate == 1.0$ do
        $P_{new} \leftarrow evolutionaryOperator\ (P, mutationRate, crossoverRate)$
        $P \leftarrow best(P, P_{new})$
        for all $P[i] \in P|evaluation(P_i) > evaluation(P_k[k-1])$ do
            $updateTopK(P_k, P[i], min_{similarity})$
        end for
        $update(mutationRate, crossoverRate)$
    end while
    $reinializationCount++$
    $P \leftarrow restart(P)$
end while
return $P_k$
```

sis (Kotzias *et al.*, 2015) (target as positive sentiment) and three about scientific communities(Gomes *et al.*, 2018) (target as majority class). In this group of data, each line represents a text and each column a word, when attribute values are 1 when the word is part of the text and 0 otherwise. Besides that, these data sets are sparse (most of the values are zero) and have many attributes and examples.

Finally, the third group consists of eight *humanities/social sciences* data sets, two about education and six about health. These data sets have many examples and attributes, but they are not sparse as *text mining* data sets. All of *humanities/social sciences* data sets represent important real problems in Brazil, such as school dropout and lifestyle of people with chronic diseases. The *Dropout* data was generated in (Bezerra *et al.*, 2016). The others are available in the *Brazilian Open Data Portal* (de Tecnologia da Informação *et al.*, 2018).

The experiments were distributed as follows. In Section 3.5.1 we compare the results of SSDP+ with different $min_{similarity}$ in order to evaluated the efficiency of $min_{similarity}$ parameter for diversity of top-k subgroups. Also in Section 3.5.1, we analyzed a top-k returned by SSDP+ in order to show how it can reduce the risk of despise relevant information and generating a more informative top-k. Finally, in Section 3.5.2 we assess the performance of SSDP+ and other algorithms in three groups of real-world high dimensional data sets in order to evaluated the competitive of proposed method in different kind of problems.

Some contents like SSDP+ implementation, some high dimensionality databases used in the tests, tables with the results of each experiment of this paper, including other evaluation metrics such as support, confidence level, TP (true positive), FP (false positive) and p-value are available on this website (`https://github.com/tarcisiodpl/ssdp_plus`).

3.5.1 Diversity and top-k in SSDP+

Figure 5 shows the average values of Qg, $SUPP^+$ and $item_{dom}$ obtained by SSDP+ with three similarity values ($min_{similarity} = \{0.9, 0.5, 0.1\}$) and SSDP for all data sets (Table 18). In this way, the Figure 5 shows that as the $min_{similarity}$ parameter decreases (0.9 to 0.1), the diversity metrics improve, with the increase of global positive support $SUPP^+$ and decrease of description redundancy $item_{dom}$. Thus, it shows that the user can control the level of diversity in SSDP+ by $min_{similarity}$ parameter. In the original SSDP the user does not have control of diverse process. Thus, in some application, the original SSDP can returned poor information for user without the possibility of improving the results.

Figure 5 also shows that the improvement of diversity ($SUPP^+$ and $item_{dom}$) usually are associated with the decrease of quality in a subgroup set (Qg). Thus, it shows that the diversification process usually remove from top-k well evaluation subgroups. However, subgroups considered redundant can represent, for example, an easier way to solve a problem or a new knowledge. So, it is important to provide some way to the algorithms to deal with the diversification problem reducing the risk of loss of relevant information.

Table 18: Summary of the three groups of data sets utilized in the experiments, where the columns $|D|$, $|D^+|$ and $|D^-|$ contains the total number of examples, and the number of positive and negative examples and the columns $\frac{|D+|}{|D|}$ and $|A|$ contains the percentage of positive examples and the number of attributes in each data set.

Name	$\|D\|$	$\|D^+\|$	$\|D^-\|$	$\frac{\|D^+\|}{\|D\|}$	$\|A\|$
		Bioinformatics			
alon	62	40	22	0.65	1000
burczynski	127	59	68	0.46	11142
chiaretti	128	74	54	0.58	6313
chin	118	75	43	0.64	11108
chowdary	104	62	42	0.60	11142
christensen	217	113	104	0.52	707
golub	72	47	25	0.65	3565
gordon	181	150	31	0.83	6267
gravier	168	111	57	0.66	1453
khan	63	23	40	0.37	1154
nakayama	105	21	84	0.20	11142
pomeory	60	39	21	0.65	3564
shipp	77	58	19	0.75	3565
singh	102	52	50	0.51	6300
sorlie	85	32	53	0.38	228
subramanian	50	33	17	0.66	5050
sun	180	81	99	0.45	27307
tian	173	137	36	0.26	6313
west	49	25	24	0.51	3565
yeoh	248	79	169	0.32	6313
		Text mining			
yelp	1000	500	500	0.5	1923
imdb	1000	500	500	0.5	2973
amazon	1000	500	500	0.5	1712
sc_100	372	68	304	0.18	521
sc_1000	372	68	304	0.18	7794
sc_ALL	372	68	304	0.18	59730
		Humanities/social sciences			
ENEM	100000	25104	74896	0.25	114
Dropout	118755	18528	100227	0.16	177
Life exp.	11619	4969	6650	0.43	908
Depression	34704	2800	31904	0.08	112
Diabetes	31023	1666	29357	0.05	112
Cancer	34704	452	34252	0.01	112
Heart	34704	988	33716	0.03	112
AVC	34704	380	34324	0.01	112

SSDP+ was applied in *Life exp.* data set (Table 18) to show how cache subgroups can be useful in real applications. The data set *Life exp.* confront the best and the worst Brazilian state in relation to life expectancy using the year 2013 as reference. So, the SSDP+ was applied for $k = 10$, $WRAcc$ as evaluation metric, $min_{similarity} = 0.1$ and $ks = 5$. Following there are three subgroups returned by the model and the first subgroup of the respective caches:

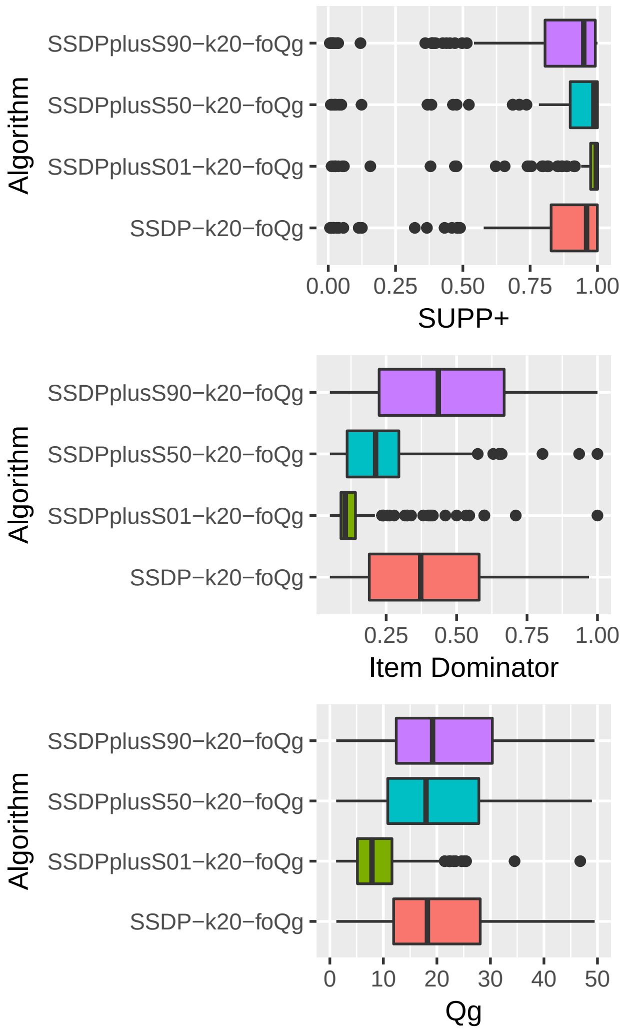

Figure 5: Distribution of Qg, $SUPP^+$ and $item_{dom}$ obtained by the SSDP+ ($min_{similarity} = \{0.9, 0.5, 0.1\}$) and SSDP in all data sets (Table 18).

- s_1: people who live at home with less than five people, can read, never got sick of Dengue fever, usually use the same health service.
  - $s_{1.1}(sim = 0.89)$: people who $s_1 \wedge$ and did not receive any application interest or unemployment insurance.
- s_2 : people who earn more than one minimum salary and usually use the same health service.
  - $s_{2.1}(sim = 0.10)$: people who earn more than one minimum salary and never driven a motorcycle.
- s_3 : people who eat salad more than 5 times a week and usually use the same health service.
  - $s_{3.1}(sim = 0.58)$: people who eat salad more than 5 times a week and never driven a motorcycle.

So, this example shows some interesting information stored in the caches. The subgroup $s_{1.1}$, for example, represents an alternative for subgroup s_1 with 89% of similarity. Following another way, the subgroups $s_{2.1}$ and $s_{3.1}$ are not tightly similar to s_2 and s_3, but they could represent relevant information for the problem. Deaths from motorcycle accidents in Brazil, for example, tripled between 2001 and 2011 (da Saúde do Brasil, 2018). All this cache information would be discarded in a traditional top-k. Besides that, the knowledge about the similarity between groups aggregates more information that can provide more insights about the problem.

3.5.2 Comparing SSDP+ to other competitive approaches

We compared the SSDP+ performance with the following algorithms: SSDP (Lucas *et al.*, 2017), SD (Gamberger & Lavrac, 2002), SD-RSS (Gamberger & Lavrac, 2002) and DSSD (Van Leeuwen & Knobbe, 2012). SD-RSS is a variation of SD approach that uses the diversity operator *RSS* (Gamberger & Lavrac, 2002). The algorithms SSDP and SD showed to be competitive on high dimensionality data sets (Lucas *et al.*, 2017). The DSSD is an approach focused on returning a diverse top-k subgroup. Others algorithms such as NMEEF (Carmona *et al.*, 2010), SDIGA (del Jesus *et al.*, 2007b) and MESDIF (del Jesus *et al.*, 2007a) were not considered because they were inefficient in previous experiments with high dimensionality data sets (Lucas *et al.*, 2017).

The confrontation was made considering the trade-off between quality and diversity, besides time cost. Diversity was evaluated in relation to coverage, through positive global support ($SUPP^+$), and description, through the presence of the dominator item ($Item_{dom}$). The experiments were done for $k = 20$ and 10 repetitions with non-deterministic algorithms.

SSDP+ was compared with SSDP, SD and SD-RSS algorithms for all data sets (Table 18) with one hour as time limit. So, after that time, the algorithm was interrupted and the top-k subgroups were collected. The evaluation metric used was Qg. SSDP+ was tested for three similarity values $min_{similarity} = \{0.1, 0.5, 0.9\}$. SD and SD-RSS used default parameters (Gamberger & Lavrac, 2002): $beamWidth = 2 * k$ and $minSupport = \sqrt{\frac{|D+|}{|D|}}$.

Table 19 shows the average values of Qg, $SUPP^+$, $item_{dom}$ and $time$ of SSDP+, SSDP, SD and SD-RSS for each group of data sets tested. The results show that it is common an algorithm to be better in quality (Qg) and worse in diversity ($SUPP^+$ and $item_{dom}$). In relation to processing time, SSDP+ was more expensive than the SSDP. This was expected, since the generation of top-k in SSDP+ is more elaborate than in SSDP. Other observation is that the difference between SD and SD-RSS could indicate that the operator *RSS* may not generate to much diversity.

The algorithms SD and SD-RSS are similar in relation to the search method and the same with SSDP and SSDP+. Thus, the hypothesis tests were focused on comparing SSDP+ and SD-RSS. Table 20 summarizes the results of the *Wilcoxon tests* ($\alpha = 0.05$) between SSDP+ and SD-RSS in relation to the metrics Qg, $SUPP^+$, $item_{dom}$ and $time$, where the values in bold

Table 19: Average Qg, $item_{dom}$, $SUPP^{+}$ and *time* for algorithm SSDP+ ($min_{similarity} = \{0.1, 0.5, 0.9\}$), SSDP, SD and SD-RSS for each group of data sets tested (Table 18).

Algorithm	**Qg**	$item_{dom}$	$SUPP^{+}$	**time**
		Bioinformatics		
SSDP+s10	9.63	0.11	0.99	8.80
SSDP+s50	25.92	0.19	0.97	8.79
SSDP+s90	32.29	0.42	0.92	9.52
SSDP	32.07	0.38	0.93	6.61
SD	27.33	0.38	0.90	9.16
SD-RSS	27.06	0.29	0.92	9.04
		Text mining		
SSDP+s10	7.09	0.10	0.65	26.24
SSDP+s50	8.55	0.16	0.61	25.19
SSDP+s90	12.42	0.46	0.39	30.06
SSDP	11.10	0.27	0.31	12.36
SD	18.02	0.65	0.36	55.77
SD-RSS	16.70	0.51	0.42	56.71
		Humanities/social sciences		
SSDP+s10	15.463	0.37	0.05	536.62
SSDP+s50	21.355	0.46	0.04	541.76
SSDP+s90	29.268	0.59	0.02	516.16
SSDP	28.286	0.48	0.03	381.82
SD	17.842	0.99	0.03	716.78
SD-RSS	17.131	0.96	0.04	752.98

represent the tests where the null hypothesis was rejected. For each data set group, we utilized the $min_{similarity}$ value that promote competition in quality and diversity at the same time. In this way, one algorithm was considered better than another when it was statistically better in one criterion (quality or diversity) and not worse in the other. The processing time was analyzed separately.

Table 20 shows that SSDP+ was better than SD-RSS in *bioinformatics* and *humanities/social sciences* data sets groups. In both, SSDP+ and SD-RSS were equivalent in relation to Qg and SSDP+ was statistically better than SD-RSS in diversity metrics ($SUPP^{+}$ or $item_{dom}$). In *text mining* data sets there were no statistical superiority between the models. This may indicate that, in sparse high dimensionality data sets, SSDP+ is less competitive. Finally, in relation to time cost, SSDP+ was statistically better than SD-RSS for all groups of data sets.

Then, SSDP+ was confronted with DSSD in 19 *bioinformatics* data sets, with three hours as maximum simulation time, $WRAcc$ as evaluation metric and $k = 20$. SSDP+ was tested for $min_{similarity} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and DSSD with the following parameters: $beamWidth = 20$, $b = 10000$, $coverBeamMultiplier = 0.90$, $beamStrategy = cover$, $maxDepth = 4$ and $minCoverage = 0$. Table 21 shows the mean values of $WRAcc$, $SUPP^{+}$, $item_{dom}$ and $time$ of SSDP+ and DSSD. So, there is not an apparent superiority between SSDP+ and DSSD in relation to trade-off between quality and diversity, but in processing time, DSSD spent much more than SSDP+.

Table 20: The *Wilcoxon* hypothesis test summary comparing the SSDP + and SD-RSS for the metrics Qg, $SUPP^{+}$, $item_{dom}$ and $time$, where bold values represent the tests where the null hypothesis were rejected.

Metric	Best mean	p-value
Bioinformatics: SSDP+s50 vs. SD-RSS		
Qg	SD-RSS	0.0565
$item_{dom}$	SSDP+s50	**0.0000**
$SUPP^{+}$	SSDP+s50	**0.0000**
time	SSDP+s50	**0.0031**
Text mining: SSDP+s90 vs. SD-RSS		
Qg	SD-RSS	0.1563
$item_{dom}$	SSDP+s90	0.8438
$SUPP^{+}$	SD-RSS	0.4375
time	SSDP+s90	**0.0312**
Humanities/social sciences: SSDP+s50 vs. SD-RSS		
Qg	SSDP+s50	0.1484
$item_{dom}$	SSDP+s50	**0.0141**
$SUPP^{+}$	the same	0.1953
time	SSDP+s50	**0.0078**

Table 21: Average Qg, $item_{dom}$, $SUPP^{+}$ and *time* for algorithm SSDP+ ($min_{similarity} = \{0.1,...,0.9\}$)and DSSD for 19 *bioinformatics* data sets from Table 18.

Algorithm	WRAcc	$item_{dom}$	$SUPP^{+}$	time
SSDP+s10	0.0333	0.1025	0.9988	8.04
SSDP+s20	0.0557	0.1026	0.9996	8.06
SSDP+s30	0.0809	0.1284	0.9995	8.09
SSDP+s40	0.1024	0.1700	0.9994	8.38
SSDP+s50	0.1216	0.2378	0.9965	8.66
SSDP+s60	0.1417	0.3510	0.9918	9.03
SSDP+s70	0.1611	0.4863	0.9913	9.68
SSDP+s80	0.1752	0.6515	0.9907	10.80
SSDP+s90	0.1856	0.8034	0.9827	11.82
DSSD	0.1725	0.6131	1	3921.94

Table 22 shows the *Wilcoxon test* ($\alpha = 0.05$) confronting SSDP+ ($min_{similarity} = \{0.7, 0.8, 0.9\}$) and DSSD. So, Table 22 shows that when one algorithm dominates the other in relation to quality it is dominated in diversity. Thus, there was no superiority between SSDP+ and DSSD, in relation to trade-off between quality and diversity.

The experiments showed that the SSDP was statistically better or equivalent to the other algorithms tested in relation to trade-off between quality and diversity. In relation to processing time, SSDP+ was statistically better than all other approaches.

Table 22: Summary of *Wilcoxon test* comparing SSDP+ and DSSD for the metric $WRAcc$, $SUPP^{+}$, $item_{dom}$ and $time$, where bold values represent the tests where the null hypothesis was rejected.

Confrontation	**Metric**	**Best mean**	**p-value**
SSDP+s70 vs. DSSD	WRAcc	DSSD	**0.0071**
	$item_{dom}$	SSDP+s70	**0.0463**
	$SUPP^{+}$	DSSD	0,1814
SSDP+s80 vs. DSSD	WRAcc	SSDP+s80	0.3320
	$item_{dom}$	DSSD	0.2683
	$SUPP^{+}$	DSSD	0.0590
SSDP+s90 vs. DSSD	WRAcc	SSDP+s90	**0.0005**
	$item_{dom}$	DSSD	**0.0045**
	$SUPP^{+}$	DSSD	**0.0224**

3.6 CONCLUSION

This paper presents the SSDP+, an evolutionary approach for mining subgroups focused on high dimensional data sets. SSDP+ was an evolution of SSDP model, that have limitations in relation to diversity in top-k subgroups.

In this way, SSDP+ solves the limitations of the original model with respect to diversity in top-k subgroups. The proposed model allows the user to choose the degree of diversity of the subgroups through the $min_{similariry}$ parameter. Besides that, SSDP+ reduces the risk of loss of relevant information as consequence of diversification process, through the use of caches. Finally, the proposed model still uses the similarity values between subgroups to generate more information for user.

The proposed model was also confronted with other competitive approaches in three groups of high dimensionality data sets. In this way, the experiments showed that in all group of data set, the proposed model returned top-k subgroups with quality and diversity statistical better or equivalent to the other algorithms using less processing time.

One of the main limitation of SSDP+ is not to work with numerical attributes. The configuration of minimum similarity parameter and cache size are also not trivial, it is depending on some manual testing. Finally, SSDP+ did not show superiority to other approach in sparse data sets (*text mining* data). Thus, an investigation in this way can points some improvements in SSDP+ performance.

4 MGP-SD: MULTIVARIATE METHOD FOR GROUP PROFILING USING SUBGROUP DISCOVERY

Este capítulo é uma reprodução completa do artigo *A Multivariate Method for Group Profiling Using Subgroup Discovery*, a ser submetido para revista ou congresso.

4.1 ABSTRACT

We propose in this paper a new method for *Group Profiling* based on *Subgroup Discovery*. *Group Profiling* is the process of constructing descriptive profiles for communities in social networks. Traditional methods for *Group Profiling* often return a set of univariate descriptors. By searching for the best univariate descriptors, these methods neglect possible interactions between them that could enhance the overall community description. Moreover, these methods do not control for coverage of descriptions. This imposes a severe limitation on the significance of results since a description may represent only a small fraction of a community. Here we investigate how the problem of *Group Profiling* can be modeled as a *Subgroup Discovery* (SD) task. We propose a new method based on SD for finding multivariate community descriptions with high coverage. We assess the quality of our proposal by finding descriptions for communities found in a real-world co-authorship network of scientific articles from Arxiv. Our experiments highlight that there is a compromise between the quality and coverage of descriptions. The experiments also show that our method improves on traditional univariate approaches, returning better descriptions both in terms of quality and coverage.

4.2 INTRODUCTION

Recent advances in information and communication technologies have caused significant changes in society, especially in the way people interact with each other. In this context, virtual social networks have become a global phenomenon with great influence on human social life. The increasing adoption of these networks enabled a continuous production of networked data that can be analyzed and mined for different purposes (Getoor & Diehl, 2005). This paper is focused on the challenging task of *Group Profiling* in social networks (Tang *et al.*, 2008), which aims to extract descriptive features from a group of people organized in communities. The derived descriptions can reveal personal values and interests that are shared by the members in a community (Tang *et al.*, 2011; Gomes *et al.*, 2013, 2016, 2018). There are several applications of *Group Profiling*, such as understanding social structures, visualization and navigation of networks, identification of changes in group themes and direct marketing (Tang *et al.*, 2011).

Traditional univariate methods for *Group Profiling* usually rely on a relevance function, which is applied to score the importance of each feature to distinguish the members of the community (Tang *et al.*, 2011; Gomes *et al.*, 2013, 2016, 2018). The features are then ranked according to the relevance function and the k highest ranked (*top-k*) are returned as the community description. As features are independently evaluated, univariate methods neglect possible interesting interactions among features that would enhance the overall description of a community. We thus postulate our first hypothesis in this work that *Group Profiling* would benefit from a multivariate approach, which accounts for such interactions and returns the best subsets of features describing a community.

Another issue currently not addressed by the state of the art methods for *Group Profiling* concerns the coverage of descriptions. The coverage of a description is the fraction of members of a given community that satisfies it. This issue is directly related to the significance of the descriptions. Descriptions with low coverage represent only a small subset of the members of a community. In the worst case, the community might be described by a pattern that occurred only by chance, or in the best case, the description would be incomplete and would not describe the entire community. This is the second research problem that we consider in this work.

This context motivates us to pose the following research question: *would Subgroup Discovery yield more expressive and comprehensive descriptions that allow for a better understanding of groups as a whole?* By that, we mean descriptions that enable analysts to grasp the characteristics of the entire group. To address this question, we propose the method MGP-SD (*Multivariate Group Profiling - Subgroup Discovery*). Our method searches for multivariate descriptions accounting for their coverage to retrieve the most relevant. For that, we model the problem of *Group Profiling* as a *Subgroup Discovery* task and, thus, make full use of the vast ecosystem for solving this task. To the best of our knowledge, this is the first time *Group Profiling* is modeled as a Subgroup Discovery task.

The suitability of such an approach lies on the fact that *Subgroup Discovery* may be presented as the task of identifying sets of features that distinguish a target group from the others in a data set (Liu *et al.*, 2015; Helal, 2016; Atzmueller, 2015; Herrera *et al.*, 2011). It has been extensively used for describing labeled data in different domains, such as medicine (Carmona *et al.*, 2013), bioinformatics (Park *et al.*, 2019), marketing (Carmona *et al.*, 2012) and e-learning (Romero *et al.*, 2009). In terms of algorithmic solutions for finding subgroups we notice different approaches in the literature (Helal, 2016). We employ in our method a recently proposed algorithm, SSDP+(Lucas *et al.*, 2018), developed for mining non-redundant subgroups in high dimensional data sets. Given the excessive amount of features in text data, we strongly believe this the best candidate for the task.

We assess the performance of our proposal on a real-world data set of scientific articles from Arxiv. In our experiments we employ our method to obtain descriptions for communities of authors in the co-authorship network of the articles. We evaluate its performance compared to the traditional univariate strategy for *Group Profiling* and verify the compromise between quality and coverage of descriptions.

The remaining of the manuscript is organized as follows. First, we introduce the *Subgroup Discovery* problem (section 4.3) and review the related literature in section 4.4. Next, we present the proposed method in section 4.5. Then, the results and discussions are presented in section 4.6. We conclude the article presenting our final remarks in section 4.7.

4.3 SUBGROUP DISCOVERY

Subgroup Discovery (SD) is a data mining task that aims to identify subgroups where the presence of a target label is exaggerated in relation to others (e.g., best clients vs. others, best school vs. others, cancer vs. healthy cells). In SD, each subgroup can be described by a rule $cond \rightarrow label_{target}$, where $cond$ is a set of conditions on the attributes, and $label_{target}$ is the target of investigation (ex. best clients).

The primary inputs in the SD process is a labeled data set D and a target label $label_{target}$. The target partitions the examples of D into positive D^{+} and negative D^{-} examples, where $D^{+} = \{e \in D \mid label(e) = target\}$, and $D^{-} = \{e \in D \mid label(e) \neq target\}$. The examples in D are described by a set of attributes A. Each attribute $a_i \in A$ has a set of values associated with it, called $dom(a_i)$. We call *features* the set of all pairs $(attribute, value)$ in a data set D, that is $F = \bigcup a_i \times dom(a_i) = \{f_1, f_2, ..., f_{|F|}\}$. Then, we say a feature $f = (a, v) \in F$ *covers* an example $e \in D$ if $a(e) = v$. Similarly, we define the cover of $f = (a, v)$ as $c(f) = \{e \in D \mid a(e) = v\}$, and its positive and negative covers respectively as $c^{+}(f) = c(f) \cap D^{+}$, $c^{-}(f) = c(f) \cap D^{-}$.

The generalization of coverage for sets of features $F' \subseteq F$ might be interpreted in two different ways. The conjunctive cover of F' is defined as the intersection of the individual covers of each of its features, that is $c(F') = \bigcap c(f)$ for $f \in F'$. The second interpretation is the disjunctive cover, which is defined as the union of individual covers, $c(F') = \bigcup c(f)$.

We call the cover of $F' \subseteq F$ a *subgroup* of D. A subgroup is interesting if it has an unusual distribution of the target feature compared to the entire data. Generally, that means a subgroup is interesting if it has many more positive examples than negatives. The goal of SD algorithms is to identify such subgroups along with their descriptions. Since subgroups and set of features are intrinsically related, we abuse the notation and refer to $F' \subseteq F$ both as a subgroup and a set of features for the sake of simplicity. The context will make it clear whether we are referring to the set of features or examples.

The definition of interestingness/relevance of a subgroup is formalized by a metric (Atzmueller, 2015). A well-known metric is the Q_g (Gamberger & Lavrac, 2002) (Equation 4.1), where s is a subgroup and g is a generalization parameter. The value of g represents the tolerance to negative examples in relation to positives in a subgroup. The higher the value of g, the more generic subgroups (descriptions) will be (Gamberger & Lavrac, 2002).

$$Q_g(s) = \frac{|c^+(s)|}{|c^-(s)| + g}, \tag{4.1}$$

There are also global metrics that evaluate the interestingness of the whole set of subgroups. One such global metric is global (positive) support, which is the proportion of positive examples covered by a set of subgroups. Let S_k be a set of k subgroups, the global support $SUPP^+$ is given by Equation 4.2. $SUPP^+$ can assume values between 0 and 1, where 1 means that S_k completely covers D^+ and small values of $SUPP^+$ implies that S_k describes just a tiny fraction of the target examples. In this way, it is essential to take into account the $SUPP^+$ to find a set of subgroups with high coverage. The concept of $SUPP^+$ is similar to *recall* in the *Information Retrieval* area.

$$SUPP^+(S_k) = \frac{|c^+(s_1) \cup \ldots \cup c^+(s_k)|}{|D^+|}, \tag{4.2}$$

Figure 6 summarizes the general idea of the *Subgroup Discovery* task. In this hypothetical example the goal is to identify subgroups that contain an over-representation of documents of a scientific community. In the example's data set each line represents a document, each column a word present in the vocabulary, and the label is the ID of a community that a document belongs to. We observe the following distribution of documents in each community: $C_1 : 40\%$, $C_2 : 30\%$ and $C_3 : 30\%$.

Now, assume that we want to identify subgroups (and their descriptions) where documents from community C_1 are over-represented; that is we want to find subgroups considering $label_{target} = C_1$. We show in Figure 6 two examples of subgroups (s_1 and s_2), where the fraction of documents of community C_1 is greater than the others. The analysis of the subgroup descriptions unveils that community C_1 is mainly composed of two areas: *Fuzzy Systems* and *Neural Networks*.

4.4 A BRIEF REVIEW ON GROUP PROFILING METHODS

The group profiling problem can be formally stated as follows. Consider a network of interest represented as a graph $G = (V, E)$ with vertices $V = \{v_1, v_2, ..., v_n\}$ and edges $E \subseteq V \times V$. Each vertex is associated to a d-dimensional vector of attributes, $\mathbf{a} \in \mathbf{A}^d, \mathbf{a} = (a_1, a_2, ..., a_d)$, where $dom(a_j)$ comprises the attribute domain (e.g., $\{0, 1\}$). A community is represented by a subgraph $P_i = (V_{P_i}, E_{P_i})$, where $V_{P_i} \subseteq V$, $E_{P_i} \subseteq V_{P_i} \times V_{P_i}$, $E_{P_i} \subseteq E$. For simplicity, we assume that communities are disjoint: $(G = \bigcup_i P_i) \wedge (P_i \cap P_j = \emptyset)$. The objective in *Group Profiling* is to select the best k descriptive attributes for each community from the original d candidate attributes. For such, one can define a quality measure $f(a_j, P_i)$ in order to assign the importance

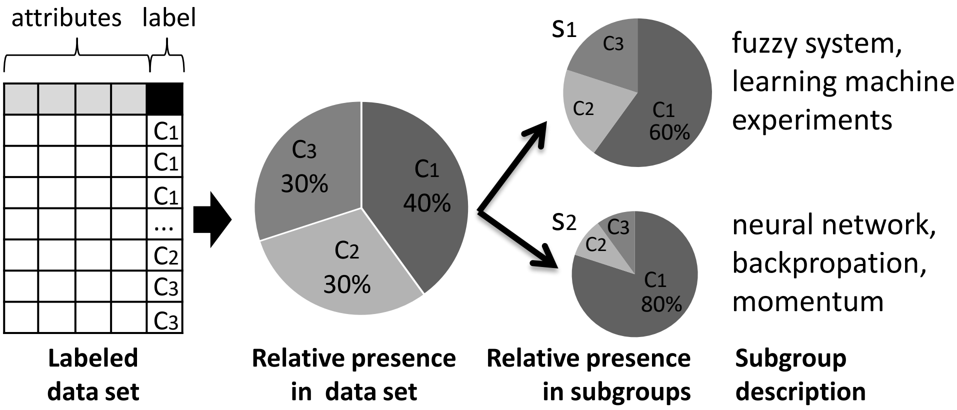
attributes
label
C1
C1
C1
...
C2
C3
C3
Labeled data set
C3 30%
C1 40%
C2 30%
Relative presence in data set
S1
C3
C2
C1 60%
S2
C3
C2
C1 80%
Relative presence in subgroups
fuzzy system, learning machine experiments
neural network, backpropation, momentum
Subgroup description

Figure 6: Example of *Subgroup Discovery* application in order to describe feature sets that concentrate the documents of community C_1 in relation to C_2 and C_3.

(i.e., descriptive score) for each attribute in a given partition, and then select the top-k scored attributes.

There are two main strategies in *Group Profiling* (Tang *et al.*, 2008): Aggregation-based Group Profiling (AGP), and Differentiation-based Group Profiling (DGP). In the former strategy descriptions consist of attributes that are most likely to occur within the community, without taking into account the rest of the network. An example of an AGP quality measure is the *Term Frequency* (TF), which indicates the number of occurrences of an attribute. On the other hand, DGP approaches select labels for communities by comparing the distribution of attributes in a community to their distribution in the remaining communities. Thus, the objective of DGP approaches is to discover the main (top-k) discriminative characteristics that represent the group, differentiating it from the rest of the network. Examples of DGP methods are: the *Term Frequency - Inverse Document Frequency* (TF-IDF) (Treeratpituk & Callan, 2006), *Wilcoxon Rank Sum Test* (WRS)(Gomes *et al.*, 2013), *Bi-standard separation* (BNS) (Tang *et al.*, 2011) and *Chi-Squared Test* (χ^2) (Gomes *et al.*, 2016). Details of the adaptations made on the methods for the group profiling problem in (Gomes *et al.*, 2018).

As already mentioned in section 4.2, traditional methods for *Group Profiling* present some flaws mainly due to their restriction to univariate descriptions (top-k features). Univariate methods do not exploit possible interactions between attributes to generate more comprehensive descriptions. Another relevant aspect neglected by traditional methods is the coverage of descriptions. It can represent a severe limitation, mainly because descriptions with low coverage do not fully represent the community and might have occurred by chance.

We illustrate the severity of the issue regarding coverage in Figure 7. The figure shows two sets of descriptions for communities C_1, C_2 and C_3 On the left, we observe a set of low coverage descriptions. We notice that features are highly redundant and cover almost the same instances of each community, leaving behind many instances that are not described at all. Oppositely, the figure on the right-hand side shows examples of descriptions with high coverage.

In this case, descriptions cover the majority of the members of each community, being, thus, more relevant than their counterpart in the first figure.

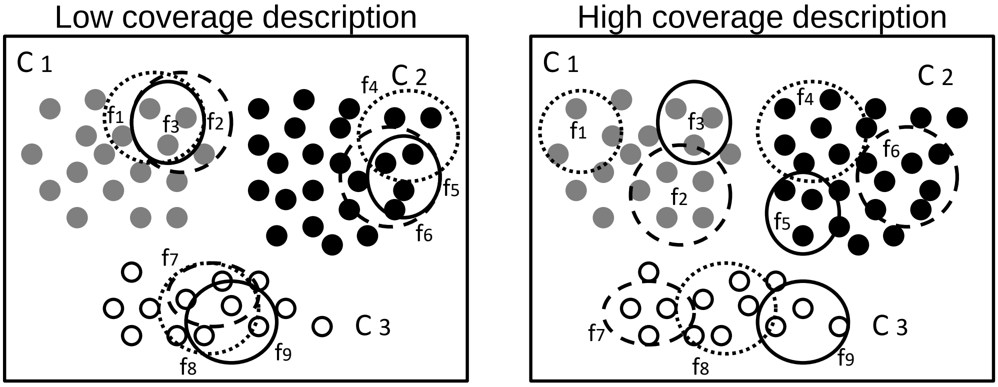

Figure 7: Example of descriptions with low and high coverage. C_1, C_2 and C_3 are the communities and $f_1, f_2, f_3, ..., f_9$ are the features that cover members of the communities.

Computing time is also an important issue in *Group Profiling*, especially for large communities. *Egocentric Differentiation-based Group Profiling* (EDGP) that reduces the computational cost by selecting only the community neighbors in the differentiation process (Tang *et al.*, 2011). More recently, Gomes et al. (Gomes *et al.*, 2018) proposed a *Centrality-based Group Profiling* (CGP) approach. In this approach the most relevant nodes are first filtered out according to their centrality (relative importance) in the observed community before descriptions are found.

4.5 MULTIVARIATE GROUP PROFILING BASED ON SUBGROUP DISCOVERY

We present in this section our method *Multivariate Group Profiling based on Subgroup Discovery* (MGP-SD). MGP-SD is a multivariate DGP approach that uses the framework of Subgroup Discovery for identifying descriptions. Figure 8 schematically presents the general methodology we propose.

We start the process by collecting a repository of documents previously acquired through a crawling process. In step 1, we generate a graph, in which nodes represent authors and two nodes are connected if they have co-authored at least one paper in our corpus. In step 2 we remove all nodes without connection (singletons) and apply the algorithm *Multi-level Aggregation Method* (MAM) (Blondel *et al.*, 2008) for detecting the communities. MAM is one of the best community detection approaches for non-directed and unweighted networks, such as co-authorship networks (Fortunato & Lancichinetti, 2009). In step 3, we remove less relevant communities based on their size and density.

We then proceed to describing communities by analyzing the articles written by their members (step 4). For this, we work with raw texts of the documents. We pre-process the documents to extract the features required for describing communities in step 5. We apply

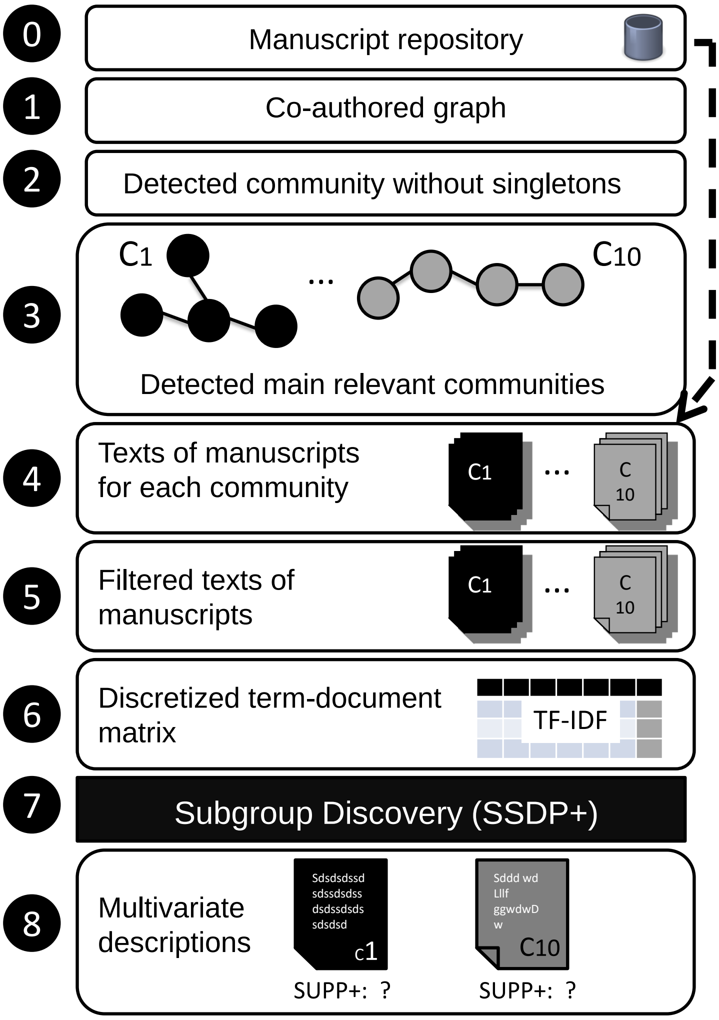
0
Manuscript repository
1
Co-authored graph
2
Detected community without singletons
3
C1
...
C10
Detected main relevant communities
4
Texts of manuscripts for each community
C1
...
C 10
5
Filtered texts of manuscripts
C1
...
C 10
6
Discretized term-document matrix
TF-IDF
7
Subgroup Discovery (SSDP+)
8
Multivariate descriptions
Sdsdsdssd sdssdsdss dsdssdsds sdsdsd
c1
SUPP+: ?
Sddd wd Lllf ggwdwD w
C10
SUPP+: ?

Figure 8: MGP-SD methodology scheme for *Group Profiling* task.

stemming, remove stop words and discard the 10% most and least frequent terms from the texts. The reason for discarding the terms is that they are either non-discriminative or relatively rare to be considered a significant descriptor for a community.

The next step aims at reformatting the data set to be used as input for the SD algorithm. Step 6 involves two sub-tasks. The first involves obtaining a term-document matrix from the pre-processed texts. The second involves the discretization of such matrix to obtain the input for the SD algorithm. The term-document matrix is generated using the *TF-IDFs* of the terms, bigrams, and trigrams. We then discretize the values into *low*, *medium*, and *high* using the first and third quartiles as thresholds.

At last, in step 7, we use an SD algorithm with the results of step 6 to obtain the descriptions. The algorithm we chose for this task is the *Simple Search Discriminative Patterns Plus* (Lucas *et al.*, 2018, 2017). This algorithm is the state of the art method for Subgroup Discovery in high dimensional data. We set the SSDP+ heuristic to use *disjunctive covers* (section 4.3) with the following parameters: $k = 10$, $k_s = 5$, $similarity = 0.1$ and Q_g $(g = 5)$ as the quality measure. The *similarity* parameter controls the coverage of top-k subgroups in SSDP+. It can assume values between 0 and 1; where low values tend to yield descriptions with higher coverage.

Since we are interested in improving the descriptions of the communities, we restrict our attention to features related to frequent terms or n-grams ($value(f) = high$). This is an important choice because, if infrequent terms were left in the data set, the algorithm would focus on these, finding descriptions of what a community is not, rather than on what it is. Nevertheless, this process may be tuned to accommodate variations in other scenarios. This is the default setting of the proposed method, but some variation must be done according to each application.

Table 23 puts MGP-SD in perspective to other works in the literature. MGP-SD uses the DGP strategy and *Subgroup Discovery* to generate descriptions. Our proposal is the first in the literature that provides multivariate descriptions and coverage control.

Table 23: State of the Art Comparative Table.

work	**strategy**	**method**	**description model**	**Coverage Control**
Tang at al. (Tang *et al.*, 2011)	AGP, DGP and EDGP	TF and BNS	Univariate	No
Gomes at al. (Gomes *et al.*, 2013)	DGP	Wilcoxon Rank Sun (WRS)	Univariate	No
Gomes at al. (Gomes *et al.*, 2016)	DGP	WRS Text, BNS, TF-IDF, χ^2	Univariate	No
Gomes at al. (Gomes *et al.*, 2018)	CGP and DGP	WRS Text, BNS, TF-IDF, χ^2	Univariate	No
MGP-SD	DGP	Subgroup Discovery	Multivariate	Yes

4.6 RESULTS AND DISCUSSIONS

This section aims to validate the proposed method MGP-SD as a multivariate approach for *Group Profiling* with coverage control. We also show how MGP-SD generates better community descriptions. We divide our experiments in two parts. First, in subsection 4.6.1, we describe a case study in which we demonstrate how the method can be applied to a real problem. Then, in subsection 4.6.2, we compare MGP-SD to traditional univariate strategies. We also analyze the compromise between quality and coverage of descriptions in *Group Profiling*.

4.6.1 Application of the proposed method

We use a corpus of articles in the *Artificial Intelligence* field to demonstrate how our method shall be applied. The articles were collected from the online repository Arxiv. These articles were first uploaded to the repository between 2012 and 2014 years.

Having the corpus at hand, we follow the methodology described in the previous section. In step 1 we generated a graph with 1850 authors and 2560 relationships. Next, in step 2 we detected 439 communities in the graph. Then, in step 3 we reduced the graph to the 10 most relevant communities, as recommended by previous works (Gomes *et al.*, 2016, 2018).

In step 4 we extract the raw texts from the articles' PDF files. Next, we pre-process the texts and remove less discriminating words. Then, as the result of step 6, we obtained a *term-document matrix* with 568 documents, 4264 terms and 10 labels. Finally, in step 7, we applied SSDP+ heuristic to generate a high coverage multivariate community description.

Table 24 shows three examples of community descriptions returned by the proposed method. We notice that, at first, descriptions were dominated by the name of authors, departments and other affiliation related features. In community 6, for example, the features *freita* and *univers british columbia* refer us to *Prof. Nando de Freitas* at the time, affiliated to the *University of British Columbia*, Canada. We observed in the communities that a few authors stand out from others in the number of published papers. This information is interesting since it corroborates the hypothesis that the principal authors of communities are enough to represent them, utilized in CGP strategy (Gomes *et al.*, 2018).

Although influential authors and their affiliations may be useful to describe communities, this is a trivial information that can be extracted directly from the documents' metadata. Here we are interested in finding descriptions that are rather topic related. So, we excluded from the searches any feature related to names of authors or their affiliations, as well as the name of venues where articles were published (conferences, journals, etc.). Table 25 shows the descriptions of the same communities after this process.

The description of community 6 now contains features related to *belief function* and *neural networks*. The community 116 is characterized by features related to *fuzzy* systems and community 156 by features related to *causal models*. The global positive support of descriptions

Table 24: Examples of MGP-SD community description by top-5 features sets.

subgroup	description
community 6	
1	freita, lang, univers british columbia
2	freita
3	lang, optimist
4	centr, colleg, council, littman
5	belief function, centr, defin new
community 116	
subgroup	description
1	duboi, fuzzi
2	max min, ordin
3	citi, follow proposit, inform retriev
4	note comput
5	point view
community 156	
subgroup	description
1	california los angel, intervent
2	comput scienc engin, verma pearl
3	comput scienc engin, edg node
4	graph structur
5	schein causat predict, weiss

$SUPP^{+}$ were 82.75%, 94.28% and 82.14 for communities 6, 116 and 156, respectively.

We noticed that, in general, descriptions were composed of three types of features. The first and most important in this application were the features that represent a core area of *Artificial Intelligence*, such as *fuzzy set*, *neural networks* and *bayesian network*. The second type were features that are often used together with terms of a core area but do not directly represent it. The feature *belong*, for example, is frequently used in articles related to *fuzzy systems* to indicate the membership of an element to a fuzzy set. Nevertheless, such term does not represent the *fuzzy* area by itself. Finally, the third type of features are ordinary terms that are not directly or indirectly related to *Artificial Intelligence*. Some of them could be, for example, related to the vocabulary of the most influential authors, thus reflecting their personal writing style.

4.6.2 Multivariate vs. univariate strategy and the coverage control in MGP-SD

In this experiment, the univariate strategy is represented by the *top-k features* method, which returns the best *k* features. On the other hand, the multivariate strategy is represented by *top-k features sets* method, that is the descriptions of subgroups returned by the SSDP+ algorithm. Both strategies used $Q_g(g = 5)$ as the quality measure and were applied to the discretized *TF-IDF term-document matrix*.

Table 25: Examples of MGP-SD community description by top-5 subgroups after removing features related to names of authors, university, congress and journals.

community 6	
subgroup	description
1	belief function, optimist,sampler, want comput
2	global optim, gradual, sampler
3	belief function, function sinc, particl
4	global optim, neural network,white
5	belief function, global optim
community 116	
subgroup	description
1	fuzzi, fuzzi set
2	max min, ordin
3	max min
4	capac, follow proposit, order set
5	point view
community 156	
subgroup	description
1	causal model reason, intervent
2	model reason infer, predict search
3	approxim margin, graph structur
4	hypothet, inform theori
5	approxim margin, howev result

Figure 9 shows the mean Q_g of descriptions using *top-k features* and *top-k features sets* for $simularity = 0.9$. Thus, the *top-k features sets* were considerably better in terms of quality measure than *top-k features* in all communities. In relation to the coverage of descriptions, the methods *top-k features* and *top-k features sets* were relatively similar, with mean $SUPP^+$ of 65.96% for the former and 69.04% for the latter. However, in relation to processing time, the *top-k features sets* spent on average $15.07s$ while the *top-k features* spent just $0.18s$.

We now analyze the compromise between quality and coverage of descriptions. Figure 10 shows the mean Q_g (x-axis) and $SUPP^+$ (y-axis) of descriptions returned by *top-k features* and *top-k features sets* for $similarity = \{0.1, 0.2, ..., 0.9\}$. The reader can notice a trend in the results: the lower the support, the higher the quality. This happens because high quality patterns are often redundant and cover the same set of examples, which culminates in lower global support. This trend is fomented here by the similarity parameter in SSDP+. This parameter controls how SSDP+ deals with redundancy in its result. As discussed by Lucas et al. (Lucas *et al.*, 2018), lower values of this parameter increases diversity at the cost of quality; often high-quality subgroups are discarded. Nevertheless, the parameter exposes to the user the possibility to fine tune the trade-off between coverage and quality. Finally, we also notice in the figure that, in spite of the possibility of tuning coverage and quality, the multivariate approach returned better results than the univariate in both aspects in eight of nine settings.

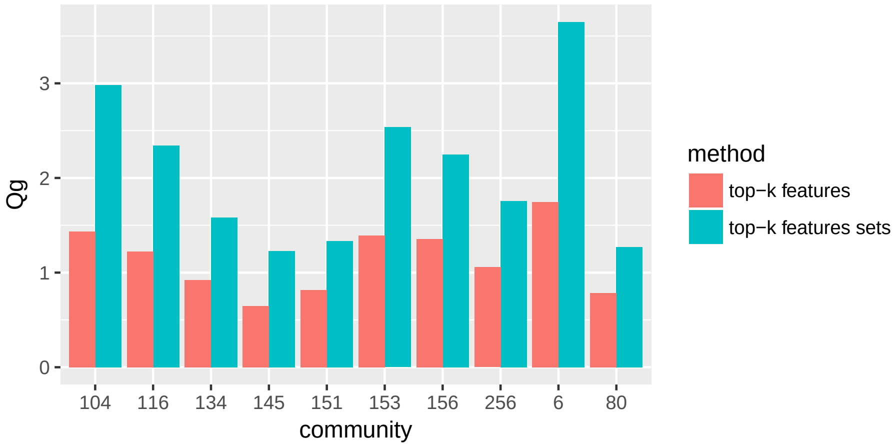

Figure 9: Mean Q_g of descriptions returned by univariate and multivariate strategies, when applied to the discretized *TF-IDF term-document matrix*.

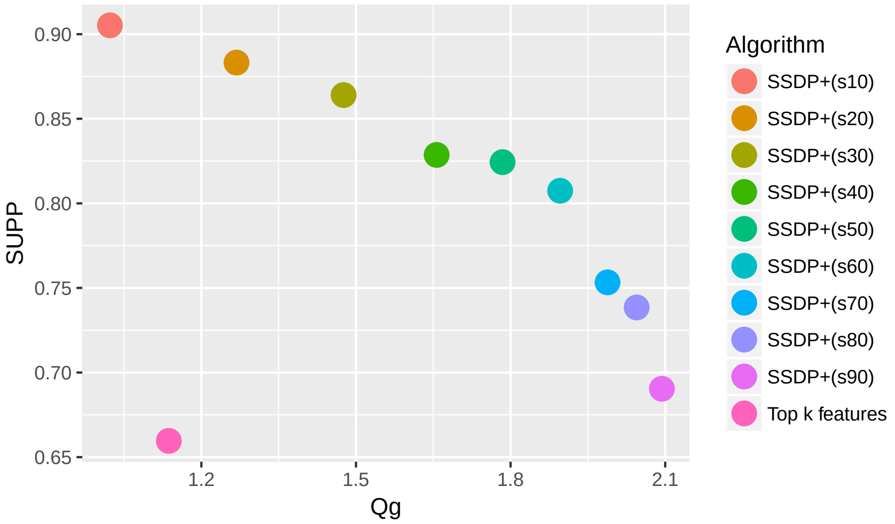

Figure 10: Mean individual quality (Q_g) and the coverage ($SUPP^+$) of descriptions returned by *top-k features sets* for *similarity* $= \{0.1, 0.2, ..., 0.9\}$ and the *top-k features*.

4.7 CONCLUSION

Traditional methods for *Group Profiling* usually return univariate descriptions, which are limited in relation to multivariate ones. Additionally, they also do not control the coverage of descriptions, which may result in either incomplete or insignificant information. So, the primary motivation of this work was to present MGP-SD, a new method for *Group Profiling* that result in multivariate descriptions with coverage control.

We validated the proposed model applying it to a co-authorship network to generate a high coverage multivariate description for 10 scientific communities in *Artificial Intelligence*. In general, even with the presence of terms not related to the purpose of the application, it was possible to clearly identify the content of communities.

We compared the use of multivariate and univariate strategies under the same conditions (data set and quality measure). Multivariate descriptions were better than univariate in relation to mean individual quality and global coverage at the same time in almost all experiments. The univariate method, however, spent less than one second to return the descriptions, while the multivariate spent around 15 seconds. We also showed that the proposed method could control the coverage of the description by an external parameter. This possibility is not presented in traditional *Group Profiling* methods.

Knowing that this work is an initial study, we aim at extending it in future works by: (1) modifying the MGP-SD to use the *Centrality-based Group Profiling* (CGP) approach, to reduce its computational requirements; and (2) conduct a qualitative study to verify the relevance of the obtained descriptions.

5 CONCLUSÃO

Esta tese propôs e aplicou soluções para a mineração de subgrupos com foco em bases de dados de alta dimensionalidade. O primeiro modelo proposto foi o SSDP, um algoritmo mono-objetivo top-k baseado em computação evolucionária que utiliza apenas dois parâmetros. Em seguida, foi proposto o SSDP+, uma extensão do SSDP que lida de uma forma inovadora com o problema de redundância gerando mais informações e minimizando o descarte de subgrupos relevantes de forma prematura. Por fim, este trabalho propôs um novo modelo para o problema de *descrição do perfil de comunidades* baseado no SSDP+.

O algoritmo SSDP foi proposto como o estado da arte entre as heurísticas de mineração de subgrupos no contexto de alta dimensionalidade. O modelo foi estatisticamente melhor ou equivalente a outros da literatura em bases de alta dimensionalidade e competitivo mesmo em bases de dados tradicionais, com relação à qualidade dos subgrupos retornados. Além disso, possui apenas dois parâmetros, sendo também simples de ajustar. No entanto, o SSDP mostrou ser um pouco limitado com relação ao combate à redundância entre os top-k subgrupos, resultando no empobrecimento das informações retornadas em algumas aplicações.

Em seguida, SSDP+ lançou um novo caminho para combate à redundância, reduzindo o risco de descarte prematuro de subgrupos relevantes e gerando mais informações para o usuário. A estratégia básica utilizada foi reter parte dos subgrupos considerados redundantes e os apresentar como soluções alternativas ao final da busca. Os subgrupos retidos nos testes realizados representaram informações relevantes em algumas ocasiões. Além disso, os experimentos mostraram que o SSDP+ permitiu a flexibilização do controle da diversidade entre os top-k subgrupos sem comprometer a competitividade do modelo com relação à qualidade dos subgrupos. O SSDP+ foi testado com diferentes grupos de bases de dados de alta dimensionalidade e foi estatisticamente melhor ou equivalente a outros algoritmos competitivos da literatura.

No entanto, existem limitações relevantes nos algoritmos propostos. A primeira delas é que eles não lidam com dados numéricos. Isso exige que o usuário discretize os valores numéricos da base de dados, o que nem sempre é uma tarefa simples e normalmente limita de forma significativa o espaço de busca do problema. A ausência de implementação dos modelos num ambiente paralelo também representa uma limitação. Isso porque, embora exista um potencial para paralelização nos algoritmos propostos, não foi possível avaliar de forma experimental o quanto isso pode ser revertido em eficiência.

Por fim, o modelo proposto para o problema de *descrição do perfil de comunidades* com base no algoritmo SSDP+ mostrou ser uma alternativa promissora. O modelo conseguiu identificar o conteúdo de comunidades científicas na área de *Inteligência Artificial* durante a aplicação realizada. Já no confronto entre a caracterização de comunidades via *top-k características* (estratégia tradicional) versus *top-k subgrupos* (estratégia proposta), o uso de subgrupos obteve caracterizações com melhor qualidade individual média e maior suporte global. Por outro lado, a caracterização via subgrupos utilizou em média pouco mais de 15 segundos para gerar as descrições enquanto que uso das top-k características utilizou menos de 1 segundo.

5.1 TRABALHOS FUTUROS

Os trabalhos futuros possuem duas frentes: uma com objetivo de propor novos mo-de-los de mineração de subgrupos e outra com o objetivo de aplicar os modelos propostos na descoberta de conhecimento em problemas relevantes, ambos no contexto de bases de dados de alta dimensionalidade.

Com relação ao desenvolvimento de novos modelos, existem três linhas com objetivos diferentes. A primeira delas tem o objetivo de encontrar algoritmos mais eficientes com relação à capacidade de encontrar os melhores subgrupos no menor espaço de tempo. Uma primeira opção é o algoritmo SD+, uma adaptação do algoritmo SD (Gamberger & Lavrac, 2002) com o objetivo de gerar diversidade entre os top-k subgrupos na mesma direção do algoritmo SSDP+. O SD é um algoritmo competitivo em bases de dados de alta dimensionalidade e possui alto poder de convergência. No entanto, tal algoritmo tende a cair em mínimos locais (Lucas *et al.*, 2017; Pontes *et al.*, 2016) e é pouco flexível com relação à diversidade entre os top-k subgrupos. Dessa forma, acreditamos que o método de diversificação do SSDP+ tornará o SD mais robusto ao problema de mínimos locais e mais flexível com relação ao controle de diversidade entre os subgrupos retornados.

Outra opção promissora para gerar algoritmos mais eficientes é o uso de abordagens híbridas. Uma primeira estratégia é combinar os algoritmos SD e SSDP+ na tarefa de busca. A primeira opção é rodar o SSDP+ e utilizar os itens dos top-k subgrupos como entrada do algoritmo SD. Assim, o SSDP+ funcionará como um seletor de itens de qualidade e diversificados para o SD, que tem alto poder de convergência e busca em profundidade. Outra opção é mesclar o uso dos algoritmos SSDP+ e SD durante a busca.

Um segundo caminho de hibridização consiste em combinar o uso de algoritmos exatos e heurísticos. Os algoritmos exatos de mineração de subgrupos garantem o melhor resultado dentro de algumas limitações dadas pelo usuário, como o suporte mínimo e confiança. No entanto, a escolha desse tipo de parâmetro não é uma tarefa simples. Se o suporte mínimo for muito baixo, por exemplo, pode não representar uma limitação útil para o algoritmo. Já se o suporte mínimo for muito alto, pode reduzir de forma exagerada o espaço de busca. Já os algoritmos heurísticos não garantem o melhor resultado, mas viabilizam as buscas, principalmente na alta dimensionalidade. Nesse contexto, a combinação desses dois tipos de algoritmos pode minimizar suas falhas e potencializar suas qualidades.

Os algoritmos heurísticos, por exemplo, podem rodar previamente numa base de dados com o objetivo de prover informações para algoritmos exatos, como valor adequado de suporte mínimo, tamanho máximo de subgrupos ou mesmo restringir o espaço de busca aos itens contidos nos subgrupos retornados pela heurística. Outra opção é que um algoritmo exato realize a busca até uma dimensão d e uma heurístia continue a busca nas dimensões seguintes. Com isso, seria possível garantir uma busca exata para os subgrupos de tamanho menor ou igual a d e as informações obtidas até então poderiam potencializar a busca realizada *a posteriori* pela heurística nas dimensões superiores a d.

Já a segunda linha de proposta de novos algoritmos tem o objetivo de adaptar os modelos propostos para lidar com bases grandes com relação ao número de exemplos. Nesse sentido, vamos estudar tecnologias de paralelização comumente utilizadas em *big data*, como *MapReduce*, bem como algoritmos da área que as utilizam. Em seguida vamos testar algumas possibilidades de adaptação do SSDP+ e outros modelos no caminho de gerar um algoritmo eficiente para bases de alta dimensionalidade e com grande número de exemplos.

Por fim, a terceira linha de algoritmos propostos terá como objetivo a proposição de modelos para mineração de subgrupos não restritos a dados categóricos. Nesse caminho, vamos analisar os algoritmos de mineração de subgrupos que trabalham com dados numéricos, bem como estudar técnicas clássicas de busca em dados numéricos, como o PSO (do inglês *Particle Swarm Optimization*), e combinar tais conceitos de forma a propor modelos com o perfil desejado.

Já a frente de pesquisa de aplicações tem como objetivo a descoberta de conhecimento através da mineração de subgrupos em problemas sociais relevantes, como saúde, violência e educação. Nesse caminho, a primeira aplicação está relacionada ao projeto de pesquisa *O IMPACTO INTERGERACIONAL DE TRANSFERÊNCIAS DE RENDA CONDICIONAIS NA SAÚDE DOS RECÉM-NASCIDOS*. O projeto possui mais três pesquisadoras da área de economia, tem duração de 18 meses e está sendo fincanciado pelo *CNPq*, *Ministério da Saúde* e *Fundação Bill § Melinda Gates*. Entre outras atividades, o projeto prevê a aplicação dos algoritmos propostos nesta tese com o objetivo de identificar os grupos de características que diferenciam os recém-nascidos com bom estado de saúde dos demais a partir de grandes bases de dados que serão disponibilizadas pelo governo brasileiro.

REFERÊNCIAS

Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 94)*, 487–499.

Aguilar-Ruiz, J. S., Ramos, I., Riquelme, J. C., & Toro, M. (2001). An evolutionary approach to estimating software development projects. *Information and Software Technology*, 43(14):875–882.

Alcalá-Fdez, J., Sánchez, L., García, S., & Jesus, M. (2009). KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318.

Atzmueller, M. (2015). Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49.

Atzmueller, M. & Puppe, F. (2006). SD-Map – a fast algorithm for exhaustive subgroup discovery. In *Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 6–17. Springer Berlin Heidelberg, Berlin, Heidelberg.

Azevedo, P. J. (2010). Rules for contrast sets. *Intelligent Data Analysis*, 14(6):623–640.

Bay, S. D. & Pazzani, M. J. (2001). Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246.

Bendimerad, A. A., Plantevit, M., & Robardet, C. (2016). Unsupervised exceptional attributed sub-graph mining in urban data. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM)*, 21–30.

Bezerra, C., Scholz, R., Adeodato, P., Lucas, T., & Ataide, I. (2016). School evasion: Applying data mining to identify relevant variables (in portuguese). In *V Congresso Brasileiro de Informática na Educação*.

Blinova, V. G., Dobrynin, D. A., Finn, V. K., Kuznetsov, S. O., & Pankratova, E. S. (2003). Toxicology analysis by means of the JSM-method. *Bioinformatics*, 19(10):1201.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:8.

Boley, M., Lucchese, C., Paurat, D., & Gärtner, T. (2011). Direct local pattern sampling by efficient two-step random procedures. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 582–590.

Bosc, G., Boulicaut, J.-F., Raïssi, C., & Kaytoue, M. (2017). Anytime discovery of a diverse set of patterns with monte carlo tree search. *Data Mining and Knowledge Discovery*.

Carmona, C. J., Chrysostomou, C., Seker, H., & del Jesus, M. (2013). Fuzzy rules for describing subgroups from influenza A virus using a multi-objective evolutionary algorithm. *Applied Soft Computing*, 13(8):3439–3448.

Carmona, C. J., González, P., Del Jesus, M., Navío-Acosta, M., & Jiménez-Trevino, L. (2011). Evolutionary fuzzy rule extraction for subgroup discovery in a psychiatric emergency department. *Soft Computing*, 15(12):2435–2448.

Carmona, C. J., González, P., del Jesus, M. J., & Herrera, F. (2010). NMEEF-SD: non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Transactions on Fuzzy Systems*, 18(5):958–970.

Carmona, C. J., González, P., del Jesus, M. J., & Herrera, F. (2014). Overview on evolutionary subgroup discovery: analysis of the suitability and potential of the search performed by evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):87–103.

Carmona, C. J., Ramírez-Gallego, S., Torres, F., Bernal, E., del Jesús, M. J., & García, S. (2012). Web usage mining to improve the design of an e-commerce website: OrOliveSur.com. *Expert Systems with Applications*, 39(12):11243–11249.

Carmona, C. J., Ruiz-Rodado, V., del Jesús, M. J., Weber, A., Grootveld, M., González, P., & Elizondo, D. (2015). A fuzzy genetic programming-based algorithm for subgroup discovery and the application to one problem of pathogenesis of acute sore throat conditions in humans. *Information Sciences*, 298:180–197.

Chin, K., DeVries, S., Fridlyand, J., Spellman, P. T., Roydasgupta, R., Kuo, W.-L., Lapuk, A., Neve, R. M., Qian, Z., Ryder, T., Chen, F., Feiler, H., Tokuyasu, T., Kingsley, C., Dairkee, S., Meng, Z., Chew, K., Pinkel, D., Jain, A., Ljung, B. M., Esserman, L., Albertson, D. G., Waldman, F. M., & Gray, J. W. (2006). Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer Cell*, 10(6):529–541.

Choi, S.-S. & Cha, S.-H. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 43–48.

da Saúde do Brasil, M. (2018). Site oficial. URL: http://portalms.saude.gov.br/.

de Tecnologia da Informação, S., do Planejamento, M., & e Gestão, D. (2018). Portal brasileiro de dados abertos. URL: http://dados.gov.br/.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197.

del Jesus, M. J., González, P., & Herrera, F. (2007a). Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules. In *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, 50–57.

del Jesus, M. J., Gonzalez, P., Herrera, F., & Mesonero, M. (2007b). Evolutionary fuzzy rule induction process for subgroup discovery: a case study in marketing. *IEEE Transactions on Fuzzy Systems*, 15(4):578–592.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

Dong, G. & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 43–52.

Fang, G., Wang, W., Oatley, B., Ness, B. V., Steinbach, M., & Kumar, V. (2011). Characterizing discriminative patterns. *CoRR*, abs/1102.4104.

Flach, P., Peter, N. L., & Zupan, B. (1999). Rule evaluation measures: a unifying view. In *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99*, 174–185.

Flach, P. A. (2003). The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, 194–201.

Fortunato, S. & Lancichinetti, A. (2009). Community detection algorithms: a comparative analysis. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, 27:1–27:2.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.

Gamberger, D. & Lavrac, N. (2002). Expert-guided subgroup discovery: methodology and application. *J. Artif. Int. Res.*, 17(1):501–527.

Gao, C. & Wang, J. (2010). Direct mining of discriminative patterns for classifying uncertain data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 861–870.

Garriga, G., Kralj, P., & Lavrač, N. (2008). Closed sets for labeled data. *The Journal of Machine Learning Research*, 9:559–580.

Getoor, L. & Diehl, C. P. (2005). Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12.

Gomes, J. E. A., Prudêncio, R., & Nascimento, A. (2016). A comparative study of group profiling techniques in co-authorship networks. *Brazilian Conference on Inteligent Systems (BRACIS 2016).*

Gomes, J. E. A., Prudêncio, R. B. C., Meira, L., Azevedo Filho, A., Nascimento, A. C. A., & Oliveira, H. (2013). Profiling for understanding educational social networking. *Software Engineering and Knowledge Engineering (SEKE 2013).*

Gomes, J. E. A., Prudêncio, R. B. C., & Nascimento, A. C. A. (2018). Centrality-based group profiling: a comparative study in co-authorship networks. *New Generation Computing*, 36(1):59–89.

Gravier, Eleonore, Pierron, G., Vincent-Salomon, A., gruel, N., Raynal, V., Savignoni, A., De Rycke, Y., Pierga, J.-Y., Lucchesi, C., Reyal, F., Fourquet, A., Roman-Roman, S., Radvanyi, F., Sastre-Garau, X., Asselain, B., & Delattre, O. (2010). A prognostic DNA signature for T1T2 node-negative breast cancer patients. *Genes, Chromosomes and Cancer*, 49(12):1125–1125.

Han, J., Wang, J., Lu, Y., & Tzvetkov, P. (2002). Mining top-k frequent closed patterns without minimum support. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, 211–218.

Helal, S. (2016). Subgroup discovery algorithms: a survey and empirical evaluation. *Journal of Computer Science and Technology*, 31(3):561–576.

Herrera, F. (2008). Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1(1):27–46.

Herrera, F., Carmona, C. J., González, P., & Del Jesus, M. J. (2011). An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 65–70.

Kavšek, B. & Lavrac, N. (2004). Analysis of example weighting in subgroup discovery by comparison of three algorithms on a real-life data set. *Advances in Inductive Rule Learning*, 64.

Kavšek, B., Lavrac, N., & Bullas, J. C. (2002). Rule induction for subgroup discovery: a case study in mining UK traffic accident data. In *Proceedings of the international multi-conference on information society*, 127–130.

Kavšek, B., Lavrač, N., & Jovanoski, V. (2006). APRIORI-SD: adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583.

Kaytoue, M., Plantevit, M., Zimmermann, A., Bendimerad, A., & Robardet, C. (2017). Exceptional contextual subgraph mining. *Machine Learning*, 1–41.

Kotzias, D., Denil, M., de Freitas, N., & Smyth, P. (2015). From group to individual labels using deep features. In *ACM SIGKDD*.

Koza, J. R. (1992). *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Lavrač, N., Kavšek, B., Flach, P., & Todorovski, L. (2004). Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188.

Li, J. & Wong, L. (2002). Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, 18(5):725–734.

Lichman, M. (2013). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Liu, X., Wu, J., Gu, F., Wang, J., & He, Z. (2015). Discriminative pattern mining and its applications in bioinformatics. *Briefings in bioinformatics*, 16(5):884–900.

Lucas, T., Silva, T. C., Vimieiro, R., & Ludermir, T. B. (2017). A new evolutionary algorithm for mining top-k discriminative patterns in high dimensional data. *Applied Soft Computing*, 59:487–499.

Lucas, T., Vimieiro, R., & Ludermir, T. (2018). SSDP+: a diverse and more informative subgroup discovery approach for high dimensional data. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8.

Luna, J. M., Romero, J. R., Romero, C., & Ventura, S. (2014). On the use of genetic programming for mining comprehensible rules in subgroup discovery. *IEEE Transactions on Cybernetics*, 44(12):2329–2341.

Madani, O., Georg, M., & Ross, D. A. (2013). On using nearly-independent feature families for high precision and confidence. *Machine Learning*, 92:457–477.

Martín, D., Alcalá-Fdez, J., Rosete, A., & Herrera, F. (2016). NICGAR: a niching genetic algorithm to mine a diverse set of interesting quantitative association rules. *Information Sciences*, 355–356":208–228.

Moens, S. & Boley, M. (2014). *Instant Exceptional Model Mining Using Weighted Controlled Pattern Sampling*, 203–214. Springer International Publishing, Cham.

Mueller, M., Rosales, R., Steck, H., Krishnan, S., Rao, B., & Kramer, S. (2009). Subgroup discovery for test selection: a novel approach and its application to breast cancer diagnosis. In *Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis (IDA 2009)*, 119–130. Springer Berlin Heidelberg, Berlin, Heidelberg.

Nakayama, R., Nemoto, T., Takahashi, H., Ohta, T., Kawai, A., Seki, K., Yoshida, T., Toyama, Y., Ichikawa, H., & Hasegawa, T. (2007). Gene expression analysis of soft tissue sarcomas: characterization and reclassification of malignant fibrous histiocytoma. *Nature*, 20(7):749–759.

Novak, P. K., Lavrač, N., & Webb, G. I. (2009). Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *The Journal of Machine Learning Research*, 10:377–403.

of Granada Research Group, U. (2016). Statistical inference in computational intelligence and data mining. URL: http://sci2s.ugr.es/sicidm.

Pachón, V., Mata, J., Domínguez, J. L., & Maña, M. J. (2011). Multi-objective evolutionary approach for subgroup discovery. In *Hybrid Artificial Intelligent Systems*, 271–278. Springer, Berlin, Heidelberg.

Pandey, G., Wang, W., Gupta, M., Fang, G., Kumar, V., & Steinbach, M. (2010). Mining low-support discriminative patterns from dense and high-dimensional data. *IEEE Transactions on Knowledge & Data Engineering*, 24:279–294.

Park, J. V., Park, S. J., & Yoo, J. S. (2019). Finding characteristics of exceptional breast cancer subpopulations using subgroup mining and statistical test. *Expert Systems with Applications*, 118:553–562.

Pontes, T., Vimieiro, R., & Ludermir, T. B. (2016). SSDP: A simple evolutionary approach for top-k discriminative patterns in high dimensional databases. In *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, 361–366.

Pulgar-Rubio, F., Rivera-Rivas, A., Pérez-Godoy, M., González, P., Carmona, C., & del Jesus, M. (2016). MEFASD-BD: Multi-objective evolutionary fuzzy algorithm for subgroup discovery in big data environments-a mapreduce solution. *Knowledge-Based Systems*.

Quackenbush, J. (2001). Computational analysis of microarray data. *Nature reviews genetics*, 2(6):418–427.

Ramey, J. (2016). URL: https://github.com/ramhiser/datamicroarray.

Rodríguez, D., Ruiz, R., Riquelme, J. C., & Aguilar-Ruiz, J. S. (2012). Searching for rules to detect defective modules: a subgroup discovery approach. *Information Sciences*, 191:14–30.

Romero, C., González, P., Ventura, S., Del Jesús, M. J., & Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: a practical application using Moodle data. *Expert Systems with Applications*, 36(2):1632–1644.

Scholz, M. (2005). Sampling-based sequential subgroup mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 265–274.

Tang, L., Liu, H., Zhang, J., Agarwal, N., & Salerno, J. J. (2008). Topic taxonomy adaptation for group profiling. *ACM Trans. Knowl. Discov. Data*, 1(4):1:1–1:28.

Tang, L., Wang, X., & Liu, H. (2011). Group profiling for understanding social structures. *ACM Trans. Intell. Syst. Technol.*, 3:15:1–15:25.

Treeratpituk, P. & Callan, J. (2006). Automatically labeling hierarchical clusters. In *Proceedings of the 2006 International Conference on Digital Government Research*, 167–176.

Van Leeuwen, M. & Knobbe, A. (2012). Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25(2):208–242.

Vimieiro, R. (2012). *Mining disjunctive patterns in biomedical data sets*. PhD thesis, The University of Newcastle, NSW, Australia.

Vimieiro, R. & Moscato, P. (2014). A new method for mining disjunctive emerging patterns in high-dimensional datasets using hypergraphs. *Information Systems*, 40:1–10.

Yu, L. T., Chung, F.-l., Chan, S. C., & Yuen, S. (2004). Using emerging pattern based projected clustering and gene expression data for cancer detection. In *Proceedings of the second conference on Asia-Pacific bioinformatics-Volume 29*, 29:75–84.

Zitzler, E., Laumanns, M., Thiele, L., *et al.* (2001). SPEA2: improving the strength pareto evolutionary algorithm. In *Eurogen*, 3242(103):95–100.

APÊNDICE A - SSDP: A SIMPLE EVOLUTIONARY APPROACH FOR TOP-K DISCRIMINATIVE PATTERNS IN HIGH DIMENSIONAL DATABASES

SSDP: A Simple Evolutionary Approach for Top-K Discriminative Patterns in High Dimensional Databases

Tarcísio Lucas, Renato Vimieiro and Teresa Ludermir
Centro de Informática
Universidade Federal de Pernambuco
Recife-PE, Brasil,
Caixa Postal 7851 – 50732-970
Email: {tdpl,rv2,tbl}@cin.ufpe.br

Abstract—**It is a great challenge to companies, governments and researchers to extract knowledge in high dimensional databases. Discriminative Patterns (DPs) is an area of data mining that aims to extract relevant and readable information in databases with target attribute. Among the algorithms developed for search DPs, it has highlighted the use of evolutionary computing. However, the evolutionary approaches typically (1) are not adapted for high dimensional problems and (2) have many nontrivial parameters. This paper presents SSDP (Simple Search Discriminative Patterns), an evolutionary approach to search the top-k DPs adapted to high dimensional databases that use only two easily adjustable external parameters.**

I. Introduction

Knowledge discovery in high dimensional databases is a challenge for companies, governments and researchers. Microarray databases are an important example of high dimensional problem. Microarray is a technology that allows measuring the expression thousands of genes in one experiment. Finding some combination of genes whose expression levels can distinguish some groups of patients (cancer vs. healthy, for example). Since microarray technology has developed databases for several important studies in Bioinformatics [1]–[5]. Microarray databases is having a revolutionary impact on molecular biology [2].

Discriminative Patterns (DPs) aims to find humanly interpretable subgroups where the presence of a label vs. others is exaggerated. From this, is possible to generate insights about a problem or just explain it in a simple way [5]. DPs have evolved rapidly with different terminologies (Subgroups Discovery [6], [7], Emerging Patterns [8] and Contrast Sets [9]).

However, mining best DPs in high dimensional databases is often computationally infeasible. In this way it highlights the development of heuristic algorithms based on *Evolutionary Computing* [10]–[16] and *Beam Search* [17]–[20]. But none of evolutionary approach has been developed with focus on very high dimensional problem. Besides that, they use complex parameters and the user has no control over the amount of returned DPs.

This paper presents the SSDP (Simple Search Discriminative Patterns), a DPs mining approach focused on high dimensional problem based on *Evolutionary Computing* and *Beam Search.* SSDP uses simple parameters and returns the top-k DPs, where k is chosen by user. SSDP was developed in special for microarray databases, but it is a general solution for high dimensional problem.

Thus, we hope this work contributes to knowledge discovery task in Bioinformatics and other high dimensional problems. This paper is organized as follows. Section II summarizes the main DPs concepts. The Section III presents some related work, followed by Section IV, where the SSDP approach is described in detail. Section V shows the experiments and Section VI the results. Finally, Section VII presents the conclusion.

II. Discriminative Patterns (DPs)

The DPs problem can be defined as follows. Let D be a database where D^+ are positive examples (research target) and D^- the negative (other examples). DPs aim to find groups where the presence of positive examples is disproportionate in relation to negative. A DP is formed by one or more items (features). Each item consists of a pair $(attribute, value)$. The universe of all possible items of D is given by $I = \{i_1, i_2, ..., i_{|I|}\}$. A three dimensionality DP, for example, can be represented as follows: $dp_3 = \{i_a, i_b, i_c\}$, where $dp_3 \subseteq I$.

The analysis of all possible DPs for a given problem is usually an infeasible task. Thus, during the search process, the DPs are evaluated automatically (using one or more evaluation metrics). There are several types of evaluation metrics, but there is no consensus about the best one. This choice often depends on the problem or specialist convictions. In this way, it is important that the DPs search algorithms accept different options of evaluation metrics to meet user needs.

The metrics used to evaluate this work are described in Table I, where TP and FP are true positives and false positives DPs, k is the number of returned DPs and $|D|$, $|D^+|$ and $|D^-|$ are number of the total, positive and negative examples. Several other evaluation metrics can be found in [5] and [7].

TABLE I
Discriminative Patterns evaluate measures.

Equation	Description
$Q_g = \frac{TP}{FP-g}$, default $g = 1$	*Trade off* between TP and FP [18]
$WRAcc = \frac{TP+FP}{\|D\|}(\frac{TP}{TP+FP} - \frac{\|D^+\|}{\|D\|})$	Relative DP accuracy [21]
$DiffSup = \|\frac{TP}{\|D^+\|} - \frac{FP}{\|D^-\|}\|$	Difference between positive and negative support [9]
$supp = \frac{TP}{\|D^+\|}$	Average positive support [12]
$conf = \frac{TP}{TP+FP}$	Confidence [7]
$SUPP = \frac{1}{k}\sum_{i=1}^{k} supp^*$	Positive support by set of DPs (D^+ covered percentage) [12]
$size$	Average size of top-k DPs

The DPs search algorithm usually return the best DPs in one of two ways: (1) based on constraints, where it returned DPs with some constraint, as minimum support and minimum confidence and (2) based on top-k, where it returned the k best DPs determined according to a given quality function. Both options have their relevance depending on the analysis goals, but the top-k approach provides more flexibility for users [6].

There are several algorithms for DPs mining [5] [7]. The use of thresholds parameters are often in these approaches. However, setting values as minimum support and confidence is not a simple task. If it is too large, the algorithm can not return any results, if it is small can not represent a useful constraint.

III. Related work

There are several DPs mining approaches based on Evolutionary Computing [10]–[16]. However, most of the performance tests on evolutionary approaches were directed to problems with less than 40 attributes and none of them was validated to thousand dimensionality order.

Some important features, as initial population, show that some evolutionary approaches would have difficulty in high dimensional databases. In [10]–[12] 75% of individuals are generated up to 25% of items $i \in I$. Already [16] uses between 1% to 50% of the attributes. This type of initialization can be problematic in high dimensional databases. A problem where $|I| = 10000$, for example, an individual using 5% of I possibilities represent a DP with 500 dimensions. This hardly represents a valid solution and may hinder the algorithm convergence.

The individual representation is another example. In evolutionary approach it is often the use of fixed size individuals equal to $|I|$ [10]–[12], [14]. But in high dimension problems the items that are not used by best DPs is often more than 99%, the most genes is zero. Other approaches using dynamic size tree generated by grammars [15], [16], but to build grammars can not be a simple process.

Another feature present in some evolutionary approaches is the number and complexity of the parameters. Table II summarizes some of the parameters required by six evolutionary approach. The definition of such parameters is not a trivial task and may hinder the use of these algorithms. It is also common in current evolutionary approaches the user has no control over the amount of DPs returned.

TABLE II
Summary of parameters used by 6 evolutionary techniques to search Discriminative Patterns.

Parameter	SDIGA [10]	MESDF [11]	NMEEF [12]	EDER [14]	GP3 [15]	FuGeP [16]
Fitness	X	X	X	X	X	X
Linguistic labels	X	X	X			X
Crossover		X	X			X
Mutation	X	X	X			X
Population	X	X	X	X	X	X
Elite size		X				
Evaluations	X	X	X			
Generations				X	X	X
Confidence	X		X		X	X
Support				X		
Sensitivity						X
Total	6	7	7	4	4	8

Finally, few studies have considered the efficiency of evolutionary methods with respect to processing time. In high dimensional databases context, time is often critical. In the next section is explained in detail SSDP algorithm, an evolutionary approach that has as main features: (1) focused on high dimensional problems, (2) uses only k and the metric evaluation as external parameters and (3) it allows the user to choose the number of DPs want to receive.

IV. SSDP: Simple Search Discriminative Patterns

SSDP uses important concepts of different search algorithms, they are:

- In [22] was presented an evolutionary algorithm to search Diverse-Frequent Pattern (a type of patterns similar to DPs) in high dimensional databases. The algorithm includes to the next generation the best individuals from old population P_{old} and others newly created by genetic operators ($P_c \leftarrow crossOver(P_{old})$ and $P_m \leftarrow mutation(P_{old})$), where the size of populations are equal ($|P_{old}| = |P_c| = |P_m|$). That is, P_{new} $best(P_{old}, P_c, P_m)$. SSDP uses this process to generate new populations.
- *Beam Search* is an efficient search strategy used in some DPs algorithms, like Subgroup Miner [17], SD [18], CN2-SD [19] and RSD [20]. There are two important features in *Beam Search* algorithm. One is to initialize the search from all one dimension DPs. This ensures that all items $i \in I$ are considered in the search. The other feature is that the searches in the dimension d are made from the best DPs smaller than d. In SSDP the initial population is formed by all one dimension DPs and the genetic operators expand the search to other dimensions.
- SD [18] is an algorithm that ensures that all DPs stored along the search are relevant. A solution dp_a is considered

irrelevant to a set DP if there is $dp_b \in DP$ that dp_a covers a subset of the positive samples and all the negative examples of dp_b. With this concept the algorithm eliminate redundancies among the top-k DPs. SSDP algorithm uses this concept only for k best DPs.

The most important parts of the SSDP algorithm are described below:

A. Representation

The individuals have variable size and represent only items used by DP. Thus, each individual is represented by integers (or index) that is the item position i in I. For example, $dp = \{2043, 213\}$ is a dp_2 composed by items at position 2043 and 203 from I.

B. Initialization and population size

The initial population is composed of all one dimentional possible DPs. That is, for each $i \in I$ an individual is created (dp_1), where I is all possible items (attribute value pairs) in the database. It represents a new way for initial population in evolutionary approach for DP problem.

C. Genetic operators

- Crossover: there are two possibilities: (1) *crossOverAND*, when two individuals unite their genes creating a new individual (used only in the first generation) or (2) *crossOverUniform*, where two individuals generate two new by uniform crossover with 50% mixing ratio.
- Mutation: there are two possibilities: (1) a new item is selected and added to the individual or (2) an old gene is replaced by new item. Both options with 50% probability.
- Selection: by binary tournament.

In each generation n new individuos are generated exclusively by crossover and other n exclusively by mutation. That is, SSDP considers the same importance to mutation and crossover operators. This is because, besides providing diversity, mutation is used to find unlikely DPs.

D. Stopping criterion

The algorithm stops when there are no changes in the top-k DPs for three consecutive generation.

E. Parameters and fitness

SSDP does not use some common parameters of other evolutionary DPs mining approaches, as mutation and crossover rate, population size and minimal support. It uses only two easily adjustable external parameters, they are:

- k: number of DPs returned to the end of the process. The k allows the user to have control over the amount of information that he wants to receive. It is also an intuitive parameter and does not require technical knowledge.
- Evaluating measure: function to evaluate DPs quality. The more functions, the more the algorithm becomes flexible for the user. SSDP theoretically allows the use of any objective function. Currently SSDP implementation has the following possibilities: Q_g, $WRAcc$ and $DiffSup$. The genetic algorithm uses the evaluating measure as fitness.

F. Algorithm

SSDP works with five population, where P, P_c, P_m and P_* size are $|I|$ and P_k size is k. They are:

- P: current population.
- P_c: generated from P by crossover.
- P_m: generated from P by mutation.
- P_*: generated by best individuals of P, P_m e P_c. It does not require that individuals are unique.
- P_k: keeps the best k individuals that are relevant. An individual is considered irrelevant in relation to P_k if it is a subset of positive and superset of negative examples for any $dp \in P_k$.

SSDP algorithm starts for all dp_1 possibilities and the genetic operators expand the search to larger dimensions. Thus, at first, the searches tend to be directed to larger dimension as best fitness individuals are found. In a second moment the individuals are becoming very specific, then, the fitness tends to worsen and the algorithm can return the searches for smaller dimension or converge.

The Algorithm 1 describes the SSDP approach. In it, the *kBestRelevants* function returns the best relevant individuals. Already the *best* function accepts repeated and not relevant individuals as a way to reduce the computational cost.

Algorithm 1 SSDP pseudocode

```
Require: $k$, $ObjectiveFunction$
  $P$     all dp1 possibilits ($i \in I$)
  $P_k$     $kBestRelevants(P)$
  while $P_k$ not improve three times in a row do
    if generation == 1 then
      $P_c$     $crossOverAND(P)$
      $P*$     $best(P, P_c)$
    else {generation > 1}
      $P_c$     $crossOverUniform(P)$
      $P_m$     $mutation(P)$
      $P*$     $best(P, P_c, P_m)$
    end if
    $update(P_k, P_*)$
    $P$     $P*$
  end while
  return $P_k$
```

V. Experiments

The experiments start from 21 original microarray databases, described in Table III. Such databases are available in the package *datamicroarray* [4] from R software [23]. For each database the majority class was considered the target of searches (p) and other examples were labeled as negative (n). The attributes of databases are all numeric. They have been discretized using methods based on frequency and width by 2, 4 and 8, totaling 126 discretized databases.

Each experiment was repeated 30 times, with the objective function Q_g ($g = 1$) and $K = \{5, 10, 20, 50\}$. SSDP performance was compared to the following algorithms:

TABLE III
MICROARRAY DATABASES DESCRIPTION

Name	N° Examples	N° Attributes	N° Labels
alon	62	2,000	2
borovecki	31	22,283	2
burczynski	127	22,283	3
chiaretti	111	12,625	2
chin	118	22,215	2
chowdary	104	22,283	2
christensen	217	1,413	3
golub	72	7,129	3
gordon	181	12,533	2
gravier	168	2,905	2
khan	63	2,308	4
nakayama	105	22,283	10
pomeroy	60	7,128	2
shipp	58	6,817	2
singh	102	12,600	2
sorlie	85	456	5
subramanian	50	10,100	2
sun	180	54,613	4
tian	173	12,625	2
west	49	7,129	2
yeoh	248	12,625	6

- Random1M e Random2M: one and two million DPs randomly generated. The purpose of this comparison is to validate SSDP heuristic.
- ExaustiveK: DPs with highest fitness among all combinations of up to four dimensions, but using only the k best items. The purpose of this comparison is to validate the SSDP ability to find non-trivial DPs.
- SD-adapted: SD algorithm was adapted to search the same types of rules of SSDP approach. The SD is based on *Beam Search*. The aim is to confront SSDP with a competitive classical SD mining approach. SD used the following parameters: $beamWidth = 2 * k$ and $minimumSupport = \frac{\sqrt{|Dp|}}{|D|}$, as indicate by author [18].

VI. RESULTS

The results were divided into two parts. In first part the aim is to evaluate the SSDP search strategy. In the second the aim is to evaluate SSDP performance.

A. Validation SSDP search strategy

SSDP starts the search considering all items possibilities $i \in I$. Table IV shows the average size frequency of top-50 DPs from 126 databases. In 18 of them the top-50 DPs were found exclusively in the first dimension. At the same time, in 15 of them the average size was above 3. This shows that is unpredictable to know what dimensions are the best DPs. In this context, boot searches by the size of $d = 1$ and evolve into other dimensions d prioritizing the well evaluated DPs seems to be an effective strategy.

Figure 1 shows the evolution of DPs average size in populations P and P_k, for $k = 50$ from *West* database. So, in the first generation P and P_k are just dp_1. After that poorer quality dp_1 are replaced by higher best quality DPs. The P behavior shows that SSDP tends to evolve searches for larger dimensions but it can change the direction to smaller dimensions when required.

TABLE IV
AVERAGE SIZE FREQUENCY OF TOP-50 DPS IN 126 *microarray* DATABASES

Average size	**Frequency**
[1;1]	18
(1;2]	54
(2;3]	39
(3;4]	14
(4;5]	1

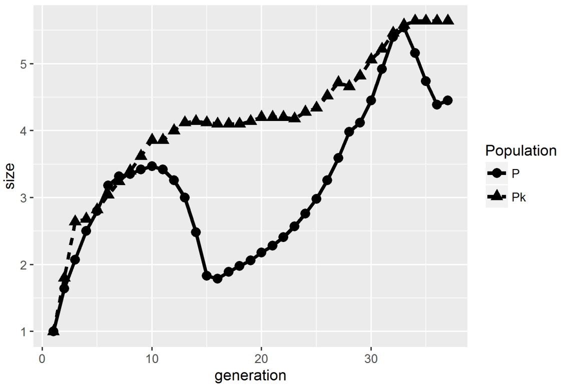

Fig. 1. DPs average size evolution in populations P and P_k, for $k = 50$ from *West* database.

B. Performance Analysis

Figure 2 and Figure 3 show respectively the average Q_g and time from SSDP, SD, *Random1M*, *Random2M* and *ExaustiveK* for $K = \{5, 10, 20, 50\}$ in 126 microarray databases. SSDP and SD obtained better average Q_g then random approach for all k values. The SSDP processing time is close to *Ramdom2M* for all k value, while SD used more time them *Ramdom2M* for $k = \{20, 50\}$. So, at first analysis it is possible to validate the heuristic SSDP. SSDP obtained better results than random approaches with closed time processing.

At second analysis it is possible to validate the SSDP regarding the ability to find nontrivial relevant DPs. Figure 2 shows that SSDP obtained better average Q_g then *ExaustiveK* for all k value. This feature also applies to the SD algorithm.

Finally, the comparison with the SD approach shows that SSDP is a promising approach in the context of top-k DPs for high dimensional databases. This is because the SSDP got considerably better DPs for all k values with time process slightly higher to $k = \{5, 10\}$ and a bit less for $k = \{20, 50\}$.

It is still applied the *Wilcoxon test* to evaluate if the performance between SSDP and SD was statistically significant. The *Wilcoxon* is a non-parametric test that has been indicated and used for performance analysis between two algorithm [24] [16]. Table V shows the result. In this way the null-hypothesis that SSDP perform equally well as SD are rejected for all k values for level of significance $\alpha = 0.01$.

An important differential of heuristics in DPs mining problem is the search capability in larger dimensions. Figure 4 shows the average size of top-k DPs for $k = \{5, 10, 20, 50\}$ from all algorithms. It shows the more successful of SSDP in

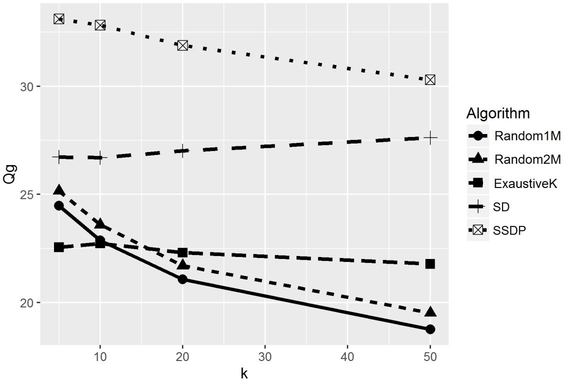

Fig. 2. Qg average for SSDP, SD, *ExaustiveK*, *Random1M* and *Random2M* in 126 microarray databases for different k values.

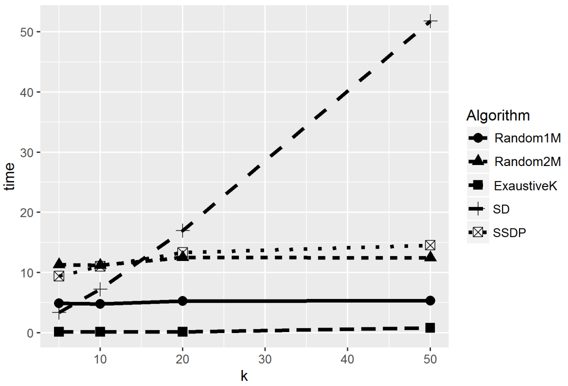

Fig. 3. Time average for SSDP, SD, *ExaustiveK*, *Random1M* and *Random2M* in 126 microarray databases for different k values.

larger dimension search for all k values. That is the probably explanation for more SSDP superiority over other algorithms.

Finally, Figure 5 shows the percentage of samples covered by top-k DPs for different values k. The tested approaches is not intended to cover all the positive examples, four of them obtained $SUPP > 80\%$ for $k = 5$ and $SUPP > 90\%$ for $k \geq 10$.

The exact values of Q_g, $time$, $size$, $SUPP$ average and other metrics as support and confidence average in all databases for $K = \{5, 10, 20, 50\}$ can be seen in Tables VI,VII,VIII and IX, respectively. It can be seen that SSDP also obtained DPs with greater confidence and support for all k values.

TABLE V
RESULTS OF THE WILCOXON TEST BETWEEN SSDP AND SD

K	p-value	Hypothesis
5	1.67E-13	Rejected by SSDP
10	5.42E-12	Rejected by SSDP
20	4.29E-10	Rejected by SSDP
50	0.0005933	Rejected by SSDP

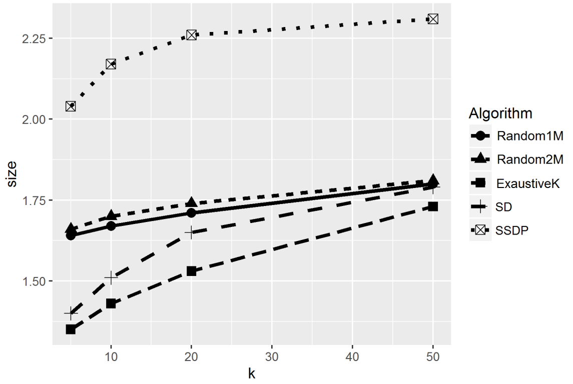

Fig. 4. Average size for SSDP, SD, *ExaustiveK*, *Random1M* and *Random2M* DPs in 126 microarray databases for different k values.

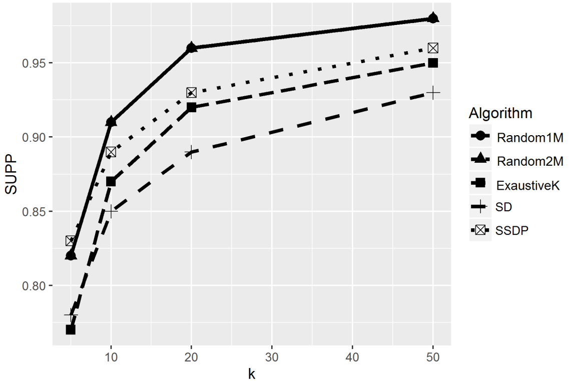

Fig. 5. Positive support by set of top-k DPs returned by *ExaustiveK*, *Random1M* and *Random2M* DPs in 126 microarray databases for different k values.

TABLE VI
$Q_g, time, size, supp, conf$ AND $SUPP$ AVERAGE FOR 126 MICROARRAY DATABASES FOR $k = 5$.

K = 5						
Algorithm	**Qg**	**time**	**size**	**conf**	**supp**	**SUPP**
SSDP	**33.12**	9.40	2.04	1.00	0.53	0.83
SD	26.73	3.34	1.40	1.00	0.45	0.79
ExaustiveK	22.57	0.14	1.35	0.99	0.42	0.78
Random1M	24.49	4.87	1.64	1.00	0.42	0.82
Random2M	25.16	11.32	1.67	1.00	0.43	0.83

TABLE VII
$Q_g, time, size, supp, conf$ AND $SUPP$ AVERAGE FOR 126 MICROARRAY DATABASES FOR $k = 10$.

K = 10						
Algorithm	**Qg**	**time**	**size**	**conf**	**supp**	**SUPP**
SSDP	**32.82**	11.06	2.18	1.00	0.53	0.90
SD	26.71	7.25	1.51	1.00	0.44	0.86
ExaustiveK	22.74	0.14	1.44	1.00	0.41	0.87
Random1M	22.88	4.81	1.68	1.00	0.39	0.91
Random2M	23.61	11.23	1.71	1.00	0.41	0.92

VII. CONCLUSION

Microarray databases are having a revolutionary impact on molecular biology. But microarray databases are an high

TABLE VIII
$Q_g, time, size, supp, conf$ AND $SUPP$ AVERAGE FOR 126 MICROARRAY DATABASES FOR $k = 20$.

K = 20						
Algorithm	**Qg**	**time**	**size**	**conf**	**supp**	**SUPP**
SSDP	**31.89**	13.33	2.27	1.00	0.52	0.94
SD	27.03	16.97	1.65	1.00	0.45	0.90
ExaustiveK	22.31	0.17	1.54	1.00	0.39	0.92
Random1M	21.07	5.29	1.72	0.99	0.37	0.96
Random2M	21.71	12.53	1.74	0.99	0.38	0.97

TABLE IX
$Q_g, time, size, supp, conf$ AND $SUPP$ AVERAGE FOR 126 MICROARRAY DATABASES FOR $k = 50$.

K = 50						
Algorithm	**Qg**	**time**	**size**	**conf**	**supp**	**SUPP**
SSDP	**30.30**	14.55	2.31	1.00	0.49	0.96
SD	27.62	51.81	1.79	1.00	0.45	0.93
ExaustiveK	21.79	0.81	1.73	1.00	0.38	0.95
Random1M	18.77	5.36	1.80	0.99	0.33	0.99
Random2M	19.52	12.46	1.81	0.99	0.35	0.99

dimension problem. Discriminative Patterns (DPs) aims to find humanly interpretable subgroups where the presence of a label vs. others is exaggerated. However, mining best DPs in high dimensional databases is often computationally infeasible. In this context, several evolutionary approaches were developed, but with little focus on high dimensional databases. They also often use many complex parameters and the user has no control over the amount of returned DPs.

This paper presented the SSDP, an evolutionary approach to search the top-k DPs adapted to high dimensional databases that use only two easily adjustable external parameters and the user can control the number of DPs returned. SSDP has as main concepts features: (1) the evolutionary strategy using concepts of *Beam Search* and (2) the simple and efficient way to represent individuals.

SSDP was validated as heuristic and the ability to find nontrivial DPs. The proposed approach also is superior to SD, a classical and competitive algorithm based on *Beam Search*. This work also showed the SSDP ability to change the focus of the search for larger or smaller as needed.

Finally, this study is being expanded to: (1) evaluate SSDP in other types of problems, (2) compare performance with newer approaches and (3) further experiments with statistical tests.

Acknowledgment

The authors would like to thank FACEPE (project IBPG-0666-1.03/13) and CNPQ (project APQ-0585-1.03/14) for supporting this research.

References

[1] J. Quackenbush, "Computational analysis of microarray data," *Nature reviews genetics*, vol. 2, no. 6, pp. 418–427, 2001.

[2] M. Molla, M. Waddell, D. Page, and J. Shavlik, "Using machine learning to design and interpret gene-expression microarrays," *AI Magazine*, vol. 25, no. 1, p. 23, 2004.

[3] M. de Souto, A. Lorena, A. Delbem, and A. de Carvalho, "Técnicas de aprendizado de máquina para problemas de biologia molecular," *Sociedade Brasileira de Computaçao*, 2003.

[4] J. Ramey. (2016) The datamicroarray r package. [Online]. Available: https://github.com/ramhiser/datamicroarray

[5] X. Liu, J. Wu, F. Gu, J. Wang, and Z. He, "Discriminative pattern mining and its applications in bioinformatics," *Briefings in bioinformatics*, p. bbu042, 2014.

[6] M. Atzmueller, "Subgroup discovery," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 35–49, 2015.

[7] F. Herrera, C. J. Carmona, P. González, and M. J. Del Jesus, "An overview on subgroup discovery: foundations and applications," *Knowledge and information systems*, vol. 29, no. 3, pp. 495–525, 2011.

[8] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 43–52.

[9] S. D. Bay and M. J. Pazzani, "Detecting group differences: Mining contrast sets," *Data Mining and Knowledge Discovery*, vol. 5, no. 3, pp. 213–246, 2001.

[10] M. J. del Jesus, P. Gonzalez, F. Herrera, and M. Mesonero, "Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 578–592, Aug 2007.

[11] M. J. del Jesus, P. González, and F. Herrera, "Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules." in *MCDM*. Citeseer, 2007, pp. 50–57.

[12] C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera, "Nmeef-sd: non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 5, pp. 958–970, 2010.

[13] V. Pachón, J. Mata, J. L. Domínguez, and M. J. Maña, "Multi-objective evolutionary approach for subgroup discovery," in *Hybrid Artificial Intelligent Systems*. Springer, 2011, pp. 271–278.

[14] D. Rodríguez, R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Searching for rules to detect defective modules: a subgroup discovery approach," *Information Sciences*, vol. 191, pp. 14–30, 2012.

[15] J. M. Luna, J. R. Romero, C. Romero, and S. Ventura, *Discovering subgroups by means of genetic programming*. Springer, 2013.

[16] C. J. Carmona, V. Ruiz-Rodado, M. J. del Jesús, A. Weber, M. Grootveld, P. González, and D. Elizondo, "A fuzzy genetic programming-based algorithm for subgroup discovery and the application to one problem of pathogenesis of acute sore throat conditions in humans," *Information Sciences*, vol. 298, pp. 180–197, 2015.

[17] W. Klosgen and M. May, "Census data mining - an application." Proceedings of the 6th European conference on principles of data mining and knowledge discovery, 2002, pp. 65–79.

[18] D. Gamberger and N. Lavrac, "Expert-guided subgroup discovery: Methodology and application," in *J. Artif. Int. Res.* AI Access Foundation, 2002, pp. 501–527.

[19] N. Lavrac, B. Kavsek, P. Flach, and L. Todorovski, "Subgroup discovery with cn2-sd," in *Journal of Machine Learning Research*, S. Wrobe, Ed., 2004, pp. 153–188.

[20] F. Zelezny and N. Lavrac, "Propositionalization-based relational subgroup discovery with rsd," in *Machine Learning*, H. Blockeel, D. Jensen, and S. Kramer, Eds. Springer, 2006, pp. 33–63.

[21] N. L. Peter, P. Flach, and B. Zupan, "Rule evaluation measures: A unifying view," in *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99*. Citeseer, 1999.

[22] S. Khatun, H. U. Alam, and S. Shatabda, "An efficient genetic algorithm for discovering diverse-frequent patterns," 2015, vol. abs/1507.05275.

[23] J. Chambers. (2016) The r project for statistical computing. [Online]. Available: https://www.r-project.org/

[24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.