Finding an appropriate generalization for a fuzzy thematic set in taxonomy

Dmitry Frolov Dept. of Data Analysis and AI, National Research University "Higher School of Economics" Moscow, Russian Federation dmitsf@gmail.com

Susana Nascimento Dept. of CS and NOVA LINCS Universidade Nova de Lisboa Caparica, Portugal snt@fct.unl.pt Boris Mirkin

Dept. of Data Analysis and AI, National Research University "Higher School of Economics" Moscow, Russian Federation Dept. of Computer Science, Birkbeck University of London UK bmirkin@hse.ru

Trevor Fenner Dept. of Computer Science Birkbeck University of London London, UK trevor@dcs.bbk.ac.uk

Abstract

This paper proposes a novel method, referred to as ParGenFS, for finding a most specific generalization of a query set, represented by a fuzzy set of topics assigned to leaves of the rooted tree of a taxonomy. This generalization lifts the query set to one or several "head subjects" in the higher ranks of the taxonomy. The head subject is supposed to "tightly" cover the query set, however dispersed that can be over branches of the tree, possibly bringing in some "gaps", that are taxonomy nodes covered by the head subject but irrelevant to the set. To balance that, we admit some "offshoots", that are nodes belonging to the query set but not covered by the head subject. The method globally minimizes the total number of head subjects and gaps and offshoots, differently weighted. Our algorithm is applied to the structural analysis and description of a collection of 17685 abstracts of research papers published in 17 Springer journals on data science for the 20-years period 1998-2017. Our taxonomy of Data Science (DST) is extracted from the Association of Computing Machinery Classification of Computing Subjects 2012 (ACM-CCS), a six-layer hierarchical taxonomy manually developed by a team of ACM experts. The DST also involves a number of additions detailing the leaves of the ACM-CCS taxonomy and added by ourselves. We find fuzzy clusters of leaf topics over the text collection, with a specially developed machinery. Three of the clusters are thematic indeed, relating to Data Science sub-areas: (a) learning, (b) information retrieval, and (c) clustering. These three clusters are lifted with ParGenFS in the DST, which allows us to make some conclusions of the tendencies of the developments in these areas.

Keywords: hierarchical taxonomy, parsimony, generalization, additive fuzzy cluster, spectral clustering, annotated suffix tree

Contents

1	Inti	roduction	4
2		simoniously lifting a fuzzy thematic ster in a taxonomy: model and method	6
	2.1	Statement of the problem	6
	$\frac{2.1}{2.2}$	Algorithm ParGenFS for finding	0
	2.2	a most specific generalization	11
	2.3	Illustrative examples	14
	2.3	2.3.1 Computing a parsimonious lift (illustrative case)	14
		2.3.1 Computing a parsimonious lift (mustrative case) 2.3.2 Computing a parsimonious lift (real-world case)	14
		2.3.2 Computing a parsimonious int (real-world case) 2.3.3 Effects of changing weights: modeling prejudices	10
		2.5.5 Effects of changing weights: modeling prejudices	19
3	Ap	olying ParGenFS to structuring	
Ŭ		conceptualizing a collection of research papers	20
	3.1	Scholarly text collection	21
	3.2	DST Taxonomy	
	3.3	Evaluation of relevance between texts	
		and key phrases	23
	3.4	Defining and computing fuzzy clusters of taxonomy topics	28
		3.4.1 Co-relevance topic-to-topic similarity score	29
		3.4.2 Additive fuzzy spectral clustering	30
		3.4.3 FADDIS thematic clusters	33
	3.5	Results of lifting clusters L, R, and C within DST	34
	3.6	Making conclusions	39
4	Cor	nclusion	40
5	Ref	erences	43
6		PENDIX: Taxonomy of Data Science	
	acc	ording to CCM-CCS 2012	48

1 Introduction

The issue of automation of structurization and interpretation of digital text collections is of ever-growing importance because of both practical needs and theoretical necessity. This paper concerns an aspect of this, the issue of generalization as a unique feature of human cognitive abilities. We use this term in the meaning pointed out by the Merriam-Webster dictionary, which defines "to generalize" as "to give a general form to" (1) or "to derive or induce (a general conception or principle) from particulars" (2a) (see https://www.merriam-webster.com/, last visited 28 November 2018).

The existing approaches to computational analysis of structure of text collections usually involve no generalization as a specific aim. The most popular tools for structuring text collections are cluster analysis and topic modelling. Both involve features of the same level of granularity as individual words or short phrases in the texts, thus no generalization as an explicitly stated goal.

Nevertheless, the hierarchical nature of the universe of meanings is reflected in the flow of publications on text analysis. We can distinguish between at least three directions at which the matter of generalization is addressed. First of all, one should mention activities related to developing taxonomies, especially those involving hyponymic/hypernymic relations (see, for example, [38, 39, 45], and references therein). A recent paper [42] should be mentioned here too, as that devoted to supplementing a taxonomy with newly emerging research topics.

Another direction is part of conventional activities in text summarization. Usually, summaries are created using a rather mechanistic approach of sentence extraction. There is, however, also an approach for building summaries as abstractions of texts by combining some templates such as subject-verb-object (SVO) triplets (see, for example, [17, 27]).

Yet one more field of activities is what can be referred to as operational generalization. In this direction, the authors use generalized case descriptions involving taxonomic relations between generalized states and their parts to achieve a tangible goal such as improving characteristics of text retrieval (see, for example, [26] and [44].)

This paper falls in neither of these approaches, as we do not attempt to change any taxonomy. We rather try to use a taxonomy for straightforwardly implementing the idea of generalization. According to the Merriam-Webster dictionary cited above, the term "generalization" refers to deriving a general conception from particulars. We assume that a most straightfor-

ward medium for such a derivation, a taxonomy of the field, is given to us. A taxonomy, in this setting, is a rooted tree over key concepts of a domain tagging the tree nodes. The tree bears a main hyponymic/hypernymic relation in the domain, so that an A-tagged node is the parent of a B-tagged node if relation "B is an A" is true. The situation of our concern is a case at which we are to generalize a fuzzy set of taxonomy leaves representing the essence of some empirically observed phenomenon. Specifically, one may be interested in patterns of novel research in a domain like Data Science. Data Science is a newly emerging area of Computer Science. The most popular Computer Science taxonomy is manually developed by the world-wide Association for Computing Machinery, a most representative body in the domain; the latest release of the taxonomy has been published in 2012 as the ACM Computing Classification System (ACM-CCS) [1]. We take its part related to Data Science, as presented in a slightly modified form by adding a few leaves in [24]. We add a few more leaves to better reflect the research papers being analyzed, and present it in the Appendix.

As an evidence of the research being conducted in Data Science, we take a collection of research papers published for a recent period in a set of journals representative of the domain of our concern. We are going to extract tight clusters of ACM-CCS topics according to the collection, each representing core tendencies of the development of the domain as reflected in the collection. It should be expected that the clusters are fuzzy, in accordance with the fuzzy nature of semantics. A major issue emerging then would be of interpretation of such a fuzzy cluster. Among various ways for addressing the challenge, we choose a most straightforward approach of generalization, prompted by the case illustrated in Figure 1 (a).



Figure 1: A taxonomy fragment with black boxes corresponding to a query set; a straightforward case (a) and a more complex case (b).

This Figure represents a taxonomy fragment with a leaf cluster fully covering all the children of the parental node A. Of course, the taxonomy concept assigned to A is a most natural generalization of the cluster in this case suggesting an interpretation of the cluster as all those topics that fall in the concept A. We are going to extend this approach to generalize less obvious cases such as that in Figure 1 (b) and see how this approach can be applied to real-world scenarios.

The rest of the paper is organized accordingly. Section 2 presents a mathematical formalization of the generalization problem as of parsimoniously lifting of a given query fuzzy leaf set to higher ranks of the taxonomy and provides a recursive algorithm leading to a globally optimal solution to the problem. Section 3 describes an application of this approach to deriving tendencies in development of the data science, that can be discerned from a set of about 18000 research papers published by the Springer Publishers in 17 journals related to data science for the past 20 years. Its subsections describe our approach to finding and generalizing fuzzy clusters of research topics. Specifically, the taxonomy of data science (DST) used in this paper is presented in Section 3a (see also Appendix); a method for developing a matrix of topic-to-paper relevance values based on automated processing of the text collection is described in Section 3b; Section 3c describes a spectral method for finding fuzzy clusters of research topics from DST using the relevance data; Section 3d presents results of lifting of three most homogeneous clusters out of six found in the DST taxonomy; and Section 3e presents our conclusions on the tendencies in the development of the corresponding parts of Data Science drawn from the lifting results. Section 4 concludes the paper.

2 Parsimoniously lifting a fuzzy thematic cluster in a taxonomy: model and method

2.1 Statement of the problem

Mathematically, a taxonomy is a rooted tree whose nodes are annotated by taxonomy topics. We consider the following problem. Given a fuzzy set S of taxonomy leaves, find a node t(S) of higher rank in the taxonomy, that covers the set S as tight as possible. Such a "lifting" problem is a mathematical explication of the human facility for generalization, that is, "the process of forming aconceptualform" of a phenomenon represented, in this case, by a fuzzy leaf subset.

The problem is not as simple as it may seem to be. Consider, for the sake of simplicity, a hard set S shown with five black leaf boxes on a fragment of a tree in Figure 1 (b). Figure 2 illustrates the situation at which the set of black boxes is lifted to the root, which is shown by blackening the root box, and its offspring, too. If we accept that set S may be generalized by the root, this would lead to a number, four, white boxes to be covered by the root and, thus, in this way, falling in the same concept as S even as they do not belong in S. Such a situation will be referred to as a gap. Lifting with gaps should be penalized. Altogether, the number of conceptual elements introduced to generalize S here is 1 head subject, that is, the root to which we have assigned S, and the 4 gaps occurred just because of the topology of the tree, which imposes this penalty. Another lifting decision is illustrated in Figure 3: here the set is lifted just to the root of the left branch of the tree. We can see that the number of gaps has drastically decreased, to just 1. However, another oddity emerged: a black box on the right, belonging to S but not covered by the root of the left branch at which the set S is mapped. This type of error will be referred to as an offshoot. At this lifting, three new items emerge: one head subject, one offshoot, and one gap. This is less than the number of items emerged at lifting the set to the root (one head subject and four gaps, that is, five), which makes it more preferable. Of course, this conclusion holds only if the relative weight of an offshoot is less than the total relative weight of three gaps.

Therefore, at lifting a fuzzy leaf set to taxonomic higher ranks there may emerge two types of errors, gaps and offshoots. If one likens assigning a general concept to the process of classification then gaps correspond to false positives, and offshoots, to false negatives. Further on, we are going to formulate an algorithm for producing the most parsimonious lifting to involve arbitrary weights assigned to errors.

We are interested to see whether a fuzzy set S can be generalized by a node t from higher ranks of the taxonomy, so that S can be thought of as falling within the framework covered by the node t. If this is done straightforwardly, for example, by picking up the last common ancestor tof the nodes in S, the coverage of S by t would be unlikely as tight as it might be desirable. To make the general concept more specific, one should admit not only gaps, but also offshoots, that are leaves which do belong in S but are not covered by t. The goal of finding an interpretable pigeonhole for S within the taxonomy can be formalized as that of finding one or more "head subjects" t to cover S with the minimum number of all the elements introduced at the generalization: head subjects, gaps, and offshoots. This goal realizes the principle of Maximum Parsimony (MP) in describing the phenomenon in question, and is popular in some domains such as Bioinformatics [33].

Consider a rooted tree T representing a hierarchical taxonomy so that its nodes are annotated with key phrases signifying various concepts. We denote the set of its *leaves* by I. The relationship between nodes in the hierarchy is conventionally expressed using genealogical terms: each node $t \in T$ is said to be the *parent* of the nodes immediately descending from t in T, its *children*. We use $\chi(t)$ to denote the set of children of t. Each *interior* node $t \in T - I$ is assumed to correspond to a concept that generalizes the topics corresponding to the leaves I(t) descending from t, viz. the leaves of the subtree T(t) rooted at t, which is conventionally referred to as the *leaf cluster of* t.

A fuzzy set on I is a mapping u of I to the non-negative real numbers that assigns a membership value, or support, $u(i) \ge 0$ to each $i \in I$. We refer to the set $S_u \subset I$, where $S_u = \{i \in I : u(i) > 0\}$, as the base of u. In general, no other assumptions are made about the function u, other than, for convenience, commonly limiting it to not exceed unity. Conventional, or *crisp*, sets correspond to binary membership functions u such that u(i) = 1if $i \in S_u$ and u(i) = 0 otherwise.

Given a fuzzy query set u defined on the leaves I of the tree T, one can consider u to be a (possibly noisy) projection of a higher rank concept, u's "head subject", onto the corresponding leaf cluster. Under this assumption, there should exist a head subject node h among the interior nodes of the tree T such that its leaf cluster I(h) more or less coincides (up to small errors) with S_u . This head subject is the generalization of u to be found. The two types of possible errors associated with the head subject if it does not cover the base precisely, are false positives and false negatives, referred to in this paper, as *gaps* and *offshoots*, respectively, are illustrated in Figures 2 and 3. Altogether, the total number of head subjects, gaps, and offshoots has to be as small as possible. To this end, we introduce a penalty for each of these elements. Assuming for the sake of simplicity, that the black box leaves on Figure 1 (b) have membership function values equal to unity, one can easily see that the total penalty at the head subject raised to the root (Figure 2) is equal to $1+4\gamma$ where 1 is the penalty for a head subject and γ , the penalty for a gap, since the lift on Figure 2 involves one head subject, the root, and four gaps, the blank box leaves. Similarly, the penalty for the lift on Figure 3 to the root of the left-side subtree is equal to $1 + \gamma + \lambda$ where λ is the penalty for an offshoot, as there is one copy of each, head subject,

gap, and offshoot, in Figure 3. Therefore, depending on the relationship between γ and λ either lift on Figure 2 or lift on Figure 3 is to be chosen. That will be the former, if $3\gamma < \lambda$, or the latter, if otherwise.

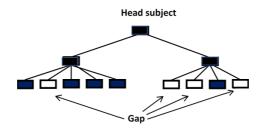


Figure 2: Generalization of the query set from Figure 1 (b) by mapping it to the root, with the price of four gaps emerged at the lift.

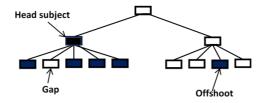


Figure 3: Generalization of the query set from Figure 1 (b) by mapping it to the root of the left branch, with the price of one gap and one offshoot emerged at this lift.

To properly define the concept of gap in general, let us first consider the *u*-irrelevant nodes in the tree T. A node $t \in T$ is referred to as *u*-irrelevant if its leaf-cluster I(t) is disjoint from the base S_u . Obviously, if a node is *u*-irrelevant, all of its descendants are also *u*-irrelevant.

Consider a candidate node h in T and its meaning relative to fuzzy set u. An h-gap is a node g of T(h), other than h, at which a loss of the meaning has occurred, that is, g is a maximal u-irrelevant node in the sense that its parent is not u-irrelevant. Conversely, establishing a node h as a head subject can be considered as a gain of the meaning of u at the node. The set of all h-gaps will be denoted by G(h). An *h*-offshoot is a leaf $i \in S_u$ which is not covered by h, i.e., $i \notin I(h)$. The set of all *h*-offshoots is $S_u - I(h)$.

Since no taxonomy perfectly reflects all the relevant real-world relations, some fuzzy topic sets u may refer to general concepts that are not captured in T(h). In such a case, two or more head subjects may be needed to cover S_u accurately, rather than just one. This motivates the following definition.

Given a fuzzy topic set u over I, a set of nodes H will be referred to as a u-cover if: (a) H covers S_u , that is, $S_u \subseteq \bigcup_{h \in H} I(h)$, and (b) the nodes in H are unrelated, i.e. $I(h) \cap I(h') = \emptyset$ for all $h, h' \in H$ such that $h \neq h'$. The interior nodes of H will be referred to as *head subjects* and the leaf nodes as *offshoots*, so the set of offshoots in H is $H \cap I$. The set of gaps in H is the union of G(h) over all head subjects $h \in H - I$.

We associate a penalty with H, so that only the most parsimonious generalizations are to be considered. A parsimonious generalization should have as small a number of head subjects, gaps and offshoots as possible. To reflect the relative importance of each of these, we use positive *penalty weights*, λ and γ , for gaps and offshoots, respectively. A head subject is assigned weight 1. The introduced concept of parsimonious generalization is a mathematical explication of the idea of most specific generalization.

Of course, the penalty value associated with a node $h \in H$ should take into account the *u*-membership values of all its offspring, not just the the leaf cluster I(h): the smaller they are, the less their effect should be. Therefore, to properly define the overall penalty, we extend the u-membership values from I to all the nodes in T. The algorithm ParGenFS (Parsimonious Generalization of Fuzzy Sets) for a parsimonious generalization of u, described below, does not depend on the way the *u*-values are assigned, so any extension of the membership values to u(t) for $t \in T - I$ is acceptable, provided the value of u(t) is zero for all u-irrelevant nodes t. Although every gap is assigned with a membership value of zero, we may consider some gaps more important than others, depending on the membership values assigned to their parents. A gap is less significant if its parent's membership value is smaller. Therefore, a measure v(q) of "gap importance" should also be defined, to be reflected in the penalty function. We suggest defining the gap importance as v(g) = u(par(g)), where par(g) is the parent of g. An alternative definition would be to scale these values by dividing them by the number of children of par(g). However, we note that the algorithm ParGenFS below works for any definition of gap importance.

We, therefore, define the penalty function p(H) for a *u*-cover *H* as:

$$p(H) = \sum_{h \in H-I} u(h) + \sum_{h \in H-I} \sum_{g \in G(h)} \lambda v(g) + \sum_{h \in H \cap I} \gamma u(h).$$
(1)

2.2 Algorithm ParGenFS for finding a most specific generalization

The problem we address is to find a *u*-cover H that globally minimizes the penalty p(H). Such a *u*-cover will be the parsimonious generalization of the query set u.

Before applying an algorithm to minimize the total penalty, one needs to execute a preliminary transformation of the tree by pruning it from all the non-maximal *u*-irrelevant nodes, i.e. descendants of gaps. Simultaneously, the sets of gaps G(t) and the internal summary gap importance $V(t) = \sum_{g \in G(t)} v(g)$ in Eq. (1) can be computed for each interior node *t*. We note that the elements of S_u are in the leaf set of the pruned tree, and the other leaves of the pruned tree are precisely the gaps.

Assume that the tree T has already been pruned and all its nodes are annotated by the membership values u(t). The sets G(t), and the gap importance values, v(t) and V(t), are assigned as described above.

Now we can apply our lifting algorithm ParGenFS. For each node t, the algorithm ParGenFS computes two sets, H(t) and L(t), containing those nodes in T(t) at which respectively gains and losses of head subjects occur (including offshoots). The associated penalty is computed as p(t) described below.

An assumption of the algorithm is that no gain can happen after a loss. Therefore, H(t) and L(t) are defined assuming that the head subject has not been gained (nor therefore lost) at any of t's ancestors. The algorithm ParGenFS recursively computes H(t), L(t) and p(t) from the corresponding values for the child nodes in $\chi(t)$.

Specifically, for each leaf node that is not in S_u , we set both $L(\cdot)$ and $H(\cdot)$ to be empty and the penalty to be zero. For each leaf node that is in S_u , $L(\cdot)$ is set to be empty, whereas $H(\cdot)$, to contain just the leaf node, and the penalty is defined as its membership value multiplied by the offshoot penalty weight γ .

To compute L(t) and H(t) for any interior node t, we analyze two possible cases: (a) when the head subject has been gained at t and (b) when the head subject has not been gained at t.

In case (a), the sets $H(\cdot)$ and $L(\cdot)$ at its children are not needed. In this case, H(t), L(t) and p(t) are defined by:

$$H(t) = \{t\}$$

$$L(t) = G(t)$$

$$p(t) = u(t) + \lambda V(t).$$
(2)

In case (b), the sets H(t) and L(t) are just the unions of those of its children, and p(t) is the sum of their penalties:

$$H(t) = \bigcup_{w \in \chi(t)} H(w)$$

$$L(t) = \bigcup_{w \in \chi(t)} L(w)$$

$$p(t) = \sum_{w \in \chi(t)} p(w).$$
(3)

To obtain a parsimonious lift, whichever case gives the smaller value of p(t) is chosen.

When both cases give the same values for p(t), we may choose arbitrarily – in the formulation of the algorithm below, we have chosen (a). The output of the algorithm consists of the values at the root, namely, H – the set of head subjects and offshoots, L – the set of gaps, and p – the associated penalty.

Algorithm ParGenFS

- INPUT: u, T
- **OUTPUT:** H = H(root), L = L(root), p = p(root)

I Base Case

for each leaf
$$i \in I$$

if $u(i) > 0$
 $H(i) = \{i\}$
 $L(i) = \oslash$
 $p(i) = \gamma u(i)$

else

$$H(i) = \emptyset$$
$$L(i) = \emptyset$$
$$p(i) = 0$$

II Recursion

$$\begin{aligned} &\text{if } u(t) + \lambda V(t) \leq \sum_{w \in \chi(t)} p(w) \\ &H(t) = \{t\} \\ &L(t) = G(t) \\ &p(t) = u(t) + \lambda V(t) \\ &\text{else} \\ &H(t) = \bigcup_{w \in \chi(t)} H(w) \\ &L(t) = \bigcup_{w \in \chi(t)} L(w) \end{aligned}$$

$$p(t) = \sum_{w \in \chi(t)} p(w)$$

It is not difficult to see that the algorithm ParGenFS leads to an optimal lifting indeed, as stated in the following proposition.

Theorem 1. Any u-cover H found by the algorithm ParGenFS is a (global) minimizer of the penalty p.

Proof. We prove this result by induction over the number of nodes n in the tree. If n = 1, there is only one node i and, in the Base Case of ParGenFS, the definition of the sets H(i) and L(i) is such that the only possible nonempty set is $H(i) = \{i\}$, when $i \in S_u$. The penalty in this case is $\gamma u(i)$, which is clearly the correct, and minimum, penalty. When $i \notin S_u$, the penalty is obviously zero.

Let us now assume that the statement is true for all rooted trees with fewer than n nodes. Consider a rooted tree T(t) with n nodes, where n > 1. Each child w of the root t is itself the root of a subtree T(w) with fewer than n nodes.

If the head subject is not gained at t, then the optimal H- and L-sets at t are clearly the unions of the corresponding sets for the subtrees T(w); this follows from the additive structure of the penalty function in (1). Clearly, the minimum penalty for the subtree T(t) must be the smaller of the penalty values $p(t) = u(t) + \lambda V(t)$ and $p(t) = \sum_{w \in \chi(t)} p(w)$, as it is in the algorithm. The result now follows by induction on n.

2.3 Illustrative examples

m 11 4

2.3.1 Computing a parsimonious lift (illustrative case)

Let us apply ParGenFS algorithm to the case presented in Fig. 4(a). This shows a three-layer tree whose nodes are labeled by letters A and B with extensions, and a fuzzy query set u having support $S_u = \{A1, A2, B1, B2\}$, with membership values shown in Fig. 4(b). The membership values are given in thousandths for convenience of reading and are normalized according to the quadratic condition defined further on in (15).

Fig. 4(b) shows the pruned tree, with the membership values extended to all nodes and the associated gap importance values $V(\cdot)$ shown at the higher rank nodes.

Tables 1 and 2 illustrate successive steps of the algorithm ParGenFS at the interior nodes A, B (Table 1) and the root (Table 2), with penalty weights $\lambda = 0.2$ and $\gamma = 0.9$. The computations leading to the smaller penalty values are highlighted in bold face.

Table 1: Computational results of ParGenFS at nodes A and B of the pruned
tree for the fuzzy query set u in Fig. 4; the gap and offshoot penalty weights
are $\lambda = 0.2$ and $\gamma = 0.9$.

 $\mathbf{D}\mathbf{Q}$

c D

1.

.

1 D C 1

1

1

i	H(i)	L(i)	p(i)	t		H(t)	L(t)	p(t)
A1	$\{A1\}$	\oslash	0.9×0.105		Gain	$\{A\}$	$\{A3, A4\}$	$0.106 + 0.2 \times 0.212$
			= 0.095					= 0.148
A2	$\{A2\}$	\oslash	0.9×0.0105	A				
			= 0.009	l l				
A3	\oslash	\oslash	0		No Gain	{A1,A2}	{A3,A4}	0.095 + 0.009
								= 0.104
A4	\oslash	\oslash	0					
B1	$\{B1\}$	\oslash	0.9×0.843		Gain	{B}	{B3}	0.994 + 0.2 imes 0.994
			= 0.759					= 1.193
B2	$\{B2\}$	\oslash	0.9×0.527	B				
			= 0.474	1				
B3	\oslash	\oslash	0		No Gain	$\{B1, B2\}$	$\{B3\}$	0.759 + 0.474
								= 1.233
C	\oslash	\oslash	0					

Table 2 shows that the "No Gain" solution at the root corresponds to the smaller penalty value 1.297. The query set u then leads to a u-cover with just node B being a head subject, leaving A1 and A2 as offshoots.

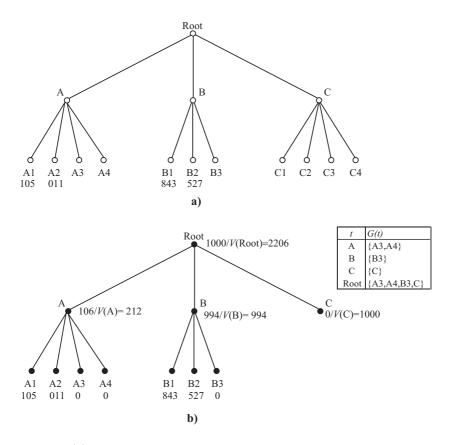


Figure 4: (a) Illustrative taxonomy T and a fuzzy query set with membership values assigned to leaves in thousandths; (b) T after pruning and annotating with membership values and gap importance values $V(\cdot)$.

Table 2: Computational results at the root for the fuzzy query set u in Fig. 4, following the computations shown in Table 1; the gap and offshoot penalty weights are $\lambda = 0.2$ and $\gamma = 0.9$.

t	H(t)	L(t)	p(t)	t		H(Root)	L(Root)	p(Root)
A	$\{A1, A2\}$	$\{A3, A4\}$	0.104		Gain	$\{Root\}$	$\{A3, A4, B3, C\}$	
B	$\{B\}$	$\{B3\}$	1.193	Root	No Gain	{A1,A2,B}	{B3}	= 1.441 0.104 + 1.193 = 1.297
C	\oslash	\oslash	0.00					

2.3.2 Computing a parsimonious lift (real-world case)

Consider the following fragment of fuzzy cluster R, described later in Section 3.5, related to Computer Vision in DST taxonomy:

- image representations: 0.262,
- shape representations: 0.246,
- appearance and text representations: 0.177,
- hierarchical representations: 0.144,
- video segmentation: 0.209,
- image segmentation: 0.196.

At penalty values $\lambda = 0.1$ and $\gamma = 0.9$, the lifting results are presented in Figure 5; the head subject is Computer vision. The penalty values involved are presented in Table 3.

The cumulative penalty at the head subject is $p(H) = 0.514 + 0.1 * (8 \cdot 0.287) + 0.9 \cdot |\emptyset| = 0.743$. If we do not lift the query set, the penalty would be greater: $p(H) = |\emptyset| + 0.1 \cdot |\emptyset| + 0.9 \cdot (0.262 + 0.246 + 0.177 + 0.144 + 0.209 + 0.196) = 1.214$.

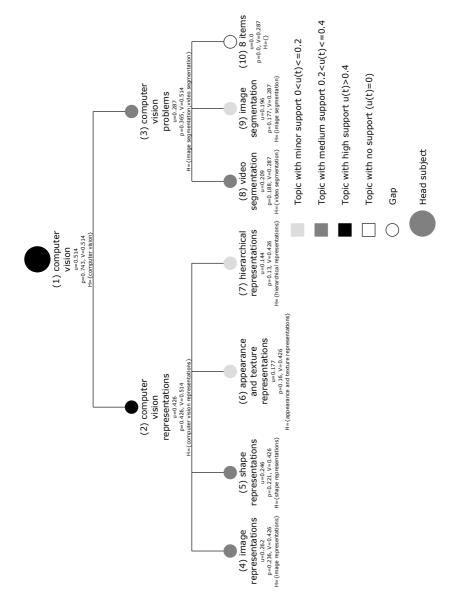


Figure 5: Lifting of the Computer Vision sub-cluster. Nodes are numbered as in Table 3.

Table 3: Nodes and variable values for lifting the Computer Vision fuzzy subcluster.

	Node name	u(t)	p(t)	v(t)	H(t)	V(t)	L(t)
1	computer vision	0.514	0.743	0.514	computer vision	interest point	interest point
						and	and
						salient region	salient region
						detections,	detections,
						shape inference,	shape inference,
						object detection,	object detection,
						object recogni-	object recogni-
						tion, object	tion, object
						identification,	identification,
						tracking,	tracking,
						reconstruction,	reconstruction,
						matching	matching
2	computer vision	0.426	0.426	0.514	computer vision		
	representations				representations		
3		0.287	0.365	0.514	image segmenta-		
	problems				tion,	and	
					video segmenta-		
					tion	tections,	
						shape inference,	
						object detection,	
						object recogni-	
						tion,	
						object identifica- tion.	
						tracking,	
						reconstruction,	
						matching	
4	image representa-	0.262	0.236	0 426	image represen-	matching	
T	tions	0.202	0.250	0.420	tations		
5	shape representa-	0 246	0.221	0 4 2 6			
Ŭ	tions	0.210	0.221	0.120	tations		
6	appearance and	0.177	0.160	0.426	appearance and		
	texture represen-				texture represen-		
	tations				tations		
7		0.144	0.130	0.426	hierarchical rep-		
	resentations				resentations		
8	video segmenta-	0.209	0.188	0.287	video segmenta-		
	tion				tion		
9	image segmenta-	0.196	0.177	0.287			
	tion				tion		
10	interest point		0.000				
	and salient region	0.000	0.000	0.287			
	detections,						
	shape inference,						
	object detection						
	object recogni-						
	tion, object						
	identification,						
	tracking,						
	reconstruction,						
	matching						

2.3.3 Effects of changing weights: modeling prejudices

It is well known that cognitive generalizations can be much biased because a cognitive system may involve prejudices preventing that from an adequate assessment of facts. Our model of generalization can imitate this phenomenon.

To illustrate this, let us return to the tree in Fig. 4 and the query set u presented in its part (b). The output of ParGenFS depends not on the membership values only, but also on the interrelation between the topology of the taxonomy tree and the penalty weights, too.

Table 4 presents the resulting u-covers for a number of different combinations of membership values and penalty weights, leading to different interpretations for the taxonomy T presented in Fig. 4(a). For the convenience of reading, the membership values in the table are not normalized. The membership values are extended to the interior nodes using the quadratic normalization condition in (15) further on.

Table 4 exhibits all possible *u*-cover outcomes at the very same fuzzy set *u*. Case 1 demonstrates a situation in which all the elements of the optimal *u*-cover are offshoots, rarely a desirable outcome. By increasing the membership value u(B2) from 0.01 to 0.1, we arrive at Case 2. The optimal *u*-cover now includes B as a head subject. This case clearly demonstrates how much the structure of the tree may affect the result. Indeed, the bulk of the membership values is clearly in the subtree rooted at A, yet A is not a head subject in this case! The slight asymmetry of the pruned tree and the fact that B has fewer children than A are the causes of this change.

Table 4: Different membership values and penalty weights that lead to all
possible descriptions in rows 1 to 5 for the illustrative taxonomy of Fig.
4(a). The algorithm is run using the membership values normalized and
extended according to the quadratic condition in (15) further on.

#	u(A1)	u(A2)	u(B1)	u(B2)	γ	λ	H(Root)	L(Root)	P(Root)
1	0.8	0.5	0.1	0.01	0.9	0.2	$\{A1, A2, B1, B2\}$	{}	1.34
2	0.8	0.5	0.1	0.1	0.9	0.2	$\{A1, A2, B\}$	<i>{B3}</i>	1.40
3	0.8	0.5	0.1	0.01	0.9	0.1	$\{A, B1, B2\}$	$\{A3, A4\}$	1.30
4	0.8	0.5	0.1	0.1	1.1	0.2	$\{A, B\}$	$\{A3, A4, B3\}$	1.56
5	0.8	0.5	0.1	0.1	0.9	0.1	$\{Root\}$	$\{A3, A4, B3, C\}$	1.31
6	0.1	0.01	0.8	0.5	0.9	0.1	$\{A1, A2, B\}$	<i>{B3}</i>	1.20
7	0.1	0.1	0.8	0.5	0.9	0.1	$\{Root\}$	$\{A3, A4, B3, C\}$	1.23
8	0.1	0.01	0.8	0.5	0.9	0.2	$\{A1, A2, B\}$	<i>{B3}</i>	1.30

Case 2 can be transformed into either Case 4 or Case 5 by changing just one or other of the weights. By changing γ from 0.9 to 1.1 (Case 4), we obtain a *u*-cover that includes both A and B as head subjects. This seems somewhat odd – because A and B are the only relevant children of the root, one might think that here the root should be a head subject in an optimal *u*-cover. Indeed, to achieve this "unnatural" result, we had to make γ greater than unity, the penalty weight for a head subject. By reducing λ from 0.2 in Case 2 to 0.1, we obtain Case 5, for which the root is the sole head subject. A different change to the parameters in Case 1, now changing the weight λ from 0.2 to 0.1 rather than changing the membership values, leads to Case 3, in which A is now the sole head subject (i.e. Case 3 is like Case 5, but reverting to the original membership values in Case 1).

Cases 6, 7 and 8 illustrate results that can be obtained by rearranging the membership values so that the high values are moved from A1 and A2 to B1 and B2, and vice versa. Case 6 leads to a *u*-cover in which B is the only head subject. Small changes to Case 6, by increasing either the membership value of A2 (Case 7) or the weight λ (Case 8), may or may not lead to changes in the result. The optimal *u*-cover makes the root a head subject in Case 7, but there is no change from Case 6 to Case 8. Using the quadratic normalization condition in (15) further on. Case 8 is just the familiar fuzzy query set *u* illustrated in Fig. 4 and Tables 1 and 2.

3 Applying ParGenFS to structuring and conceptualizing a collection of research papers

Being confronted with the problem of structuring and interpreting a set of research publications in a domain, one can think of either of the following two pathways to take. One is so-to-speak empirical and the other theoretical. The first pathway tries to discern main categories from the texts, the other, from knowledge of the domain. The first approach is exemplified by the LDA-based topic modeling (see, for example, [6, 43]); the second approach, by using an expert-driven taxonomy such as ACM-CCS [1] (see, for example, [14, 28]).

This paper follows the second pathway by moving, in sequence, through the following stages:

- preparing a scholarly text collection;
- preparing a taxonomy of the domain under consideration;

- developing a matrix of relevance values between taxonomy leaf topics and research publications from the collection;
- finding fuzzy clusters according to the structure of relevance values;
- lifting the clusters over the taxonomy to conceptualize them via generalization;
- making conclusions from the generalizations.

Each of the items is covered in a separate subsection further on.

3.1 Scholarly text collection

Because of a generous offer from the Springer Publisher, we were able to download a collection of 17685 research papers together with their abstracts published in 17 journals related to Data Science, in our opinion, for 20 years from 1998-2017. We take the abstracts to these papers as a representative collection. The list of the journals is in Table 5.

Table 5: List of Springer journals related to Data Science used as the source for our text collection. Some journals start not in 1998, but later because of unrelated issues.

#	Title	Volumes	Years
1	Pattern Analysis and Applications	1-20	1998 - 2017
2	Journal of Mathematical Imaging and Vision	14 - 29	2001 - 2017
3	World Wide Web	1 - 20	1998 - 2017
4	Artificial Intelligence Review	22 - 48	2004 - 2017
5	Annals of Mathematics and Artificial Intelligence	23 - 80	1998 - 2017
6	Pattern Analysis and Applications	1 - 20	1998 - 2017
7	Knowledge and Information Systems	1 - 52	1999 - 2017
8	Machine Learning	30 - 106	1998 - 2017
9	Swarm Intelligence	1-11	2007 - 2017
10	Applied Intelligence	14-47	1998 - 2017
11	Neural Processing Letters	7 - 45	1998 - 2017
12	Data Mining and Knowledge Discovery	2-31	1998 - 2017
13	Machine Vision and Applications	15 - 28	2004 - 2017
14	Social Network Analysis and Mining	1 - 7	2011 - 2017
15	International Journal on Document Analysis and Recognition	1 - 20	1998 - 2017
16	International Journal of Multimedia Information Retrieval	1-6	2012 - 2017
17	Pattern Recognition and Image Analysis	16 - 27	2006 - 2017

3.2 DST Taxonomy

Taxonomy is a form of knowledge engineering which is getting more and more popular. Most known are taxonomies within the bioinformatics Genome Ontology project (GO) [12], health and medicine SNOMED CT project [16] and the like.

Mathematically, a taxonomy is a rooted tree, a hierarchy, whose all nodes are labeled by main concepts of a domain. The hierarchy corresponds to a relation of inclusion: the fact that node A is the parent of B means that B is part, or a special case, of A. An important characteristic of a rooted tree is that each node has only one parent.

There are two major approaches for developing a domain taxonomy: automatic and manual. The latter currently is by far more mature and developed. However, even this approach suffers of deficiences summarized as follows: "Taxonomy design decisions regarding the used classification structures, procedures and descriptive bases are usually not well described and motivated." [41]. The automatic approach can exploit a multitude of digital resources and methods for semantic analysis. A rather comprehensive attempt is described in [41]. Our preferences, at this moment, are with taxonomies manually established by a representative body of specialists. Indeed, such a taxonomy does not depend on purely empirical data utilized by automatic methods.

Moreover, a manual taxonomy usually balances the theoretical insight and practical experience accumulated in the community represented by the body issuing it. Such is the Computing Classification System (ACM-CCS 2012) by the world-wide Association for Computing Machinery [1]. The subdomain of our choice is Data Science, comprising such areas as machine learning, data mining, data analysis, etc. We take that part of the ACM-CCS 2012 taxonomy, which is related to Data Science, and add a few leaves related to more recent Data Science developments. A major extract from the taxonomy of Data Science is published in [24].

The higher ranks of the taxonomy are presented in Table 6 and its full version, in Table 6 in the Appendix. The leaves added by the authors are labeled with a star "*".

Table 6: ACM Computing Classification System (ACM-CCS) 2012 higher rank subjects related to Data Science.

Subject index	Subject name	
1.	Theory of computation	
1.1.	Theory and algorithms for application domains	
2.	Mathematics of computing	
2.1.	Probability and statistics	
3.	Information systems	
3.1.	Data management systems	
3.2.	Information systems applications	
3.3.	World Wide Web	
3.4.	Information retrieval	
4.	Human-centered computing	
4.1.	Visualization	
5.	Computing methodologies	
5.1.	Artificial intelligence	
5.2.	Machine learning	

3.3 Evaluation of relevance between texts and key phrases

After ground-breaking discoveries of methods for automatically developing sets of topics relevant to collections of documents [5,6], it has become popular to concentrate on key-words taken from the documents being analyzed. We, however, prefer using topics produced manually by committees of experts because of the obvious advantages: comprehensiveness and stability. Therefore, our list of key-phrases comprises lower-rank ACM-CCS taxonomy topics related to data science. To take into account specifics of the structure of the collection of texts being analyzed, one should focus on the relevance between taxonomy key-phrases and the texts. Most popular and well established approaches to scoring keyphrase-to-document relevance include the so-called vector-space approach [34] and probabilistic text model approach [33]. These, however, rely on individual words and text pre-processing.

We utilize a method, first developed by R. Pampapathi et al [30] and further advanced by Chernyak and Mirkin [9,10], the AST method for evaluating keyphrase-to-text relevance score using purely string frequency information. An advantage of the method is that it requires no manual work, but works rather reliably, as claimed by these authors.

An Annotated Suffix Tree (AST) is a weighted rooted tree used for storing text fragments and their frequencies. To build an AST for a text string, all suffixes from this string are extracted.

A k-suffix of a string $x = x_1x_2...x_N$ of length N is a continuous end fragment $x^k = x_{N-k+1}x_{N-k+2}...x_N$. For example, a 3-suffix of string *INFORMATION* is substring *ION*, and a 5-suffix, *ATION*. Each AST node is assigned a symbol and the so-called annotation (frequency of the substring corresponding to the path from the root to the node including the symbol at the node). The root node of AST has no symbol or annotation (see Figure 3.3). An algorithm for building an AST for any given string $x = x_1x_2...x_N$ is described below.

- 1. Initialize an AST to consist of a single node, the root: T.
- 2. Find all the suffixes of the given string:

$$\{x^k = x_{N-k+1}x_{N-k+2}\dots x_N | k = 1, 2, \dots, N\}.$$

3. For each suffix x^k find its maximal overlap, that is, a path from the root in T coinciding with its beginning fragment $x^{k_{max}}$. At each node of the path for $x^{k_{max}}$ add 1 to the annotation. If the length of the overlap $x^{k_{max}}$ is less than k, the path is extended by adding new nodes corresponding to symbols from the remaining part of this suffix. Annotations of all the new nodes are set to be 1.

To accelerate the working of the method, one should use efficient versions of algorithms utilising suffix trees and suffix arrays (see, for example, [13]).

Having an AST T built, we can score the string-to-document relevance over the AST. To do this, we follow [21] by computing the conditional probability of node u in T:

$$p(u) = \frac{f(u)}{f(parent(u))}.$$
(4)

For all the immediate offspring of the root (R), formula has the following form:

$$p(u) = \frac{f(u)}{\sum_{v \in T: parent(v) = R} f(v)},$$
(5)

where f(u) is the frequency annotation of the node u. Using the formula above, one can calculate the probability of node u relative to all its siblings.

For each suffix x^k of string x the relevance score $s(x^k, T)$ is defined as:

$$s(x^k, T) = \frac{1}{k_{max}} \sum_{i=1}^{k_{max}} p(x_i^k).$$
 (6)

The AST relevance score of string x and text T is defined as the sum of all the suffix scores:

$$S(x,T) = \frac{1}{N} \sum_{k=1}^{N} s(x^k,T).$$
 (7)

In practical computations, we split any document into a set of strings (usually consisting of 2-3 consecutive words), create an empty AST for the document and add these strings in the AST in sequence, by using the algorithm above.

To lessen the effects of frequently occurring general terms, the scoring function is modified by five-fold decreasing the weight of stop-words. The list of stop-words includes: "learning, analysis, data, method" and a few postfixes: "s/es, ing, tion". After an AST for a document has been built, the time complexity of calculating the string-to-document relevance score is $O(m^2)$ where m is the length of the query string. This does not depend on the document length, in contrast to the popular Levenstein-distance based approaches.

Let us consider, for example, a "document" consisting of a string 'inference'. We start from an AST consisting of a single node (Root), split the document into short strings and sequentially add suffixes of these strings to the AST. In the example, we should add all the suffixes of the only string to the AST: from 'inference' to 'nference' to 'ference', etc., and to 'e' (see Figures 6 and 7).

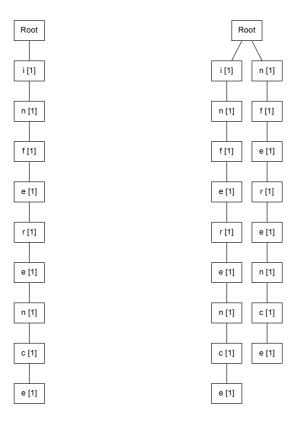


Figure 6: AST for string 'inference': step 1.

.....

Figure 7: AST for string 'inference': step 2.

The final AST for string 'inference' is shown in Figure 8.

Let us calculate the relevance score of a string 'fence' according to the AST found above. The string has five suffixes: 'fence', 'ence', 'nce', 'ce', 'e'. First of all, one needs to calculate relevance scores for these suffixes according to formula 6. These scores are presented in Table 7. After that, according to formula 7, we calculate score for the whole string:

$$S(fence, T) = \frac{1}{5} \cdot (0.5555 + 0.7083 + 0.5740 + 0.5555 + 0.3333) \approx 0.545$$

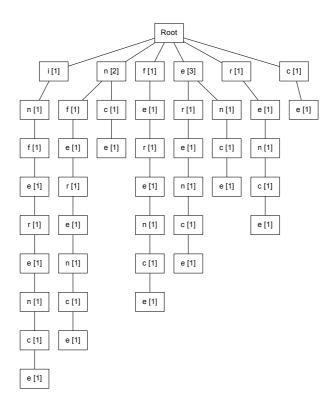


Figure 8: Final AST for string 'inference'.

Consider another example, relevance score calculations for string 'since' (see Table 8). For the whole string, we have:

$$S(since, T) = \frac{1}{5} \cdot (0 + 0.5555 + 0.5740 + 0.5555 + 0.3333) \approx 0.404$$

Suffix	Match	Score
'fence'	'fe'	$\frac{1}{2} \cdot \left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
'ence'	'ence'	$\frac{1}{4} \cdot \left(\frac{3}{9} + \frac{1}{2} + \frac{1}{1} + \frac{1}{1}\right) \approx 0.7083$
'nce'	'nce'	$\frac{1}{3} \cdot \left(\frac{2}{9} + \frac{1}{2} + \frac{1}{1}\right) \approx 0.5740$
'ce'	'ce'	$\frac{1}{2} \cdot \left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
'e'	'e'	$\frac{1}{1} \cdot \left(\frac{3}{9}\right) \approx 0.3333$

Table 7: Computing the relevance scores for suffixes of a string 'fence'.

Table 8: Computing the scores for suffixes of string 'since'.

Suffix	Match	Score
'since'	۰,	0
'ince'	ʻin'	$\frac{1}{2}\left(\frac{1}{9} + \frac{1}{1}\right) \approx 0.5555$
'nce'	'nce'	$\frac{1}{3}\left(\frac{2}{9} + \frac{1}{2} + \frac{1}{1}\right) \approx 0.5740$
'ce'	'ce'	$\frac{1}{2}\left(\frac{1}{9}+\frac{1}{1}\right)\approx 0.5555$
'e'	'e'	$\frac{1}{1}\left(\frac{3}{9}\right) \approx 0.3333$

3.4 Defining and computing fuzzy clusters of taxonomy topics

Clusters of topics should reflect co-occurrence of topics: the greater the number of texts to which both topics t and t' are relevant, the greater the interrelation between t and t', the greater the chance for topics t and t' to fall in the same cluster.

We have tried several popular clustering algorithms. Unfortunately, no satisfactory results have been found. Therefore, we present here results obtained with the FADDIS algorithm from [21] developed specifically for finding thematic clusters.

This algorithm implements assumptions that are relevant to the task:

- LN Laplacian Normalization: Similarity data transformation modeling to an extent heat distribution and, in this way, making the cluster structure sharper.
- AA Additivity: Thematic clusters behind the texts are additive so that similarity values are sums of contributions by different hidden themes.
- AN Non-Completeness: Clusters do not necessarily cover all the key phrases available as the text collection under consideration may be irrelevant to some of them.

3.4.1 Co-relevance topic-to-topic similarity score

Given a keyphrase-to-document matrix R of relevance scores, it is converted to a keyphrase-to-keyphrase similarity matrix A or scoring the "corelevance" of keyphrases according to the text collection structure. The similarity score $a_{tt'}$ between topics t and t' can be computed as the inner product of vectors of scores $r_t = (r_{tv})$ and $r_{t'} = (r_{t'v})$ where $v = 1, 2, \ldots, V = 17685$. The inner product is moderated by a natural weighting factor assigned to texts in the collection. The weight of text v is defined as the ratio of the number of topics n_v relevant to it and n_{max} , the maximum n_v over all $v = 1, 2, \ldots, V$. A topic is considered relevant to v if its relevance score is greater than 0.2 (a threshold found experimentally, see [9]). The numbers of texts in our collection, that are relevant to 0, 1, or more topics, are presented in Table 9.

N	Number of texts	Number of relevant topics
	1237	0
	2353	1
	7114	2,3, or 4
	6124	5-11
	857	12 or more

Table 9: The numbers of relevant topics in our text collection.

The accepted topic-to-topic similarity measure has the following properties [21]:

- 1. The similarity matrix is positive semi-definite.
- 2. The similarity between two topics can be positive if and only if there is at least one text in which both are engaged.
- 3. The greater the individual relevance scores, the greater the similarity.
- 4. Given a pair of topics, the greater the set of texts relevant to both of them, the greater the similarity.

3.4.2 Additive fuzzy spectral clustering

Let us denote the total set of leaf topics by T and assume that a fuzzy cluster over T is represented by a fuzzy membership vector $\vec{u} = (u_t), t \in T$, such that $0 \leq u_t \leq 1$ for all $t \in T$, and an intensity $\mu > 0$, a scale coefficient to relate the membership scores to the similarity scores. For T being a set of research topics and $\vec{u} = (u_t), t \in T$, a membership values vector representing the a semantic substructure of a corpus of research papers under consideration, the product $(\mu u_t)(\mu u_{t'}) = \mu^2 u_t u_{t'}$ can be considered as the contribution by the research direction represented by the cluster under consideration to the total similarity score $a_{tt'}$ between topics t and t'. The additive fuzzy clustering model in [21] states that the entries in the topicto-topic similarity matrix A can be considered as resulting from additive contributions of K fuzzy clusters, up to small errors to be minimized:

$$a_{tt'} = \sum_{k=1}^{K} \mu_k^2 u_{kt} u_{kt'} + e_{tt'}, \qquad (8)$$

where $\vec{u}_k = (u_{kt})$ is the membership vector of cluster k, and μ_k its intensity. These assumptions require that clusters are extracted according to an additive model. A method developed in [21], FADDIS, finds clusters in (8) one-by-one, which accords with the assumptions above. Paper [21] provides some theoretical and experimental computation results to demonstrate that FADDIS is competitive over other fuzzy clustering approaches.

A fuzzy cluster (\vec{u}, μ) is extracted to minimize the one-cluster least-squares criterion

$$E = \sum_{t,t' \in T} (w_{tt'} - \xi u_t u_{t'})^2 \tag{9}$$

with respect to unknown positive $\xi = \mu^2$ and fuzzy membership vector $\vec{u} = (u_t)$ for a given similarity matrix $W = (w_{tt'})$.

Initially, W = A. Then the matrix W is changed by subtracting from it that part of the similarities that is accounted for by the found cluster, $W \leftarrow W - \mu^2 \vec{u} \vec{u}'$, where μ and \vec{u} are the intensity and membership vector of the cluster.

In this way, A is step-by-step additively decomposed in accordance with the model (8), so that the number of clusters K can be determined during the process. The first-order optimality condition for the criterion E in (9) states that an optimal ξ satisfies equation:

$$\xi = \frac{\sum_{t,t' \in T} w_{tt'} u_t u_{t'}}{\sum_{t \in T} u_t^2 \sum_{t' \in T} u_{t'}^2},$$

or, in matrix terms,

$$\xi = \frac{\vec{u}^T W \vec{u}}{\left(\vec{u}^T \vec{u}\right)^2}.\tag{10}$$

By substituting this value of ξ in (9), we obtain

$$E = S(W) - \xi^2 \left(\vec{u}^T \vec{u} \right)^2,$$

where $S(W) = \sum_{t,t' \in T} w_{tt'}^2$ is the similarity data scatter.

Denoting the last term by $G(\vec{u})$, we have

$$G(\vec{u}) = \xi^2 \left(\vec{u}^T \vec{u} \right)^2 = \left(\frac{\vec{u}^T W \vec{u}}{\vec{u}^T \vec{u}} \right)^2, \tag{11}$$

which provides for a decomposition of the similarity data scatter into explained and unexplained parts, according to the equation above:

$$S(W) = G(\vec{u}) + E. \tag{12}$$

Therefore, to minimize E, one must maximize $G(\vec{u})$ in (11) or its square root,

$$g(\vec{u}) = \xi \vec{u}^T \vec{u} = \frac{\vec{u}^T W \vec{u}}{\vec{u}^T \vec{u}},\tag{13}$$

The value $g(\vec{u})$ in (13) is the so-called Rayleigh quotient whose maximum is the maximum eigenvalue of W reached at the corresponding eigenvector. This brings forward the celebrated spectral clustering approach [37]. According to this approach, one first solves the unconstrained problem of maximizing $g(\vec{u})$ with respect to any u, to obtain the normalized eigenvector \vec{z} corresponding to the maximum eigenvalue of W. Then its projection \vec{u} onto the set of nonnegative fuzzy membership vectors is found:

$$u_t = \begin{cases} 0, & \text{if } z_t \le 0; \\ z_t, & \text{if } 0 < z_t \le 1. \end{cases}$$
(14)

As the criterion in (13) involves division over the squared Euclidean norm of vector u, the Euclidean normalization condition is most natural for the solution, so that the extracted cluster u is normalized accordingly:

$$\vec{u}^T \vec{u} = 1. \tag{15}$$

The process of one-by-one extracting fuzzy clusters stops when any of the conditions below is satisfied:

- 1. The value of ξ given by (10) at the current step is negative.
- 2. The contribution of a single extracted cluster is too low. For example, a cluster should contribute at least as much as an average entity, so that the threshold 1/N should be considered a well-defined conservative value.
- 3. The residual scatter E becomes smaller than, say, 5% of the original similarity data scatter.
- 4. A pre-specified number K_{max} of clusters is reached.

To make the hidden cluster structure in similarity data sharper, we apply the so-called Laplacian normalization [18, 37]. This normalization usually applies with the so-called minimum normalized cut criterion, and, thus, brings forward the minimum eigenvalue, whereas FADDIS uses the maximum one. This is why this normalization is modified in [21] to involve the inverse eigen-values. Specifically, the so-called Laplacian pseudo-inverse transformation (Lapin, in short) is applied. Given a similarity matrix W, the $N \times N$ diagonal matrix D is computed, with (t, t)-th entry equal to $d_t = \sum_{t' \in T} w_{tt'}$, the sum of row t of W, and the Laplacian is defined as

 $L_n = I - D^{-1/2} W D^{-1/2}$. Then the Laplace Pseudo IN verse transformation (Lapin) is defined as

$$L_n^+ = Z\tilde{\Lambda}^{-1}Z^T,$$

where Z is the matrix of eigenvectors corresponding to non-zero eigenvalues, from the spectral decomposition $L_n = Z\Lambda Z^T$, and the diagonal matrix $\tilde{\Lambda}$ is obtained from Λ by removing zero eigenvalues and replacing each non-zero eigenvalue λ by $1/\lambda$.

Lastly, the eigenvector corresponding to the maximum eigenvalue of L_n^+ is taken to generate the fuzzy cluster for the model (9).

3.4.3 FADDIS thematic clusters

After computing the 317×317 topic-to-topic co-relevance matrix, converting in to a topic-to-topic Lapin transformed similarity matrix, and applying FADDIS clustering, we sequentially obtained 6 clusters, of which three clusters seem especially homogeneous. We denote them using letters L, for 'Learning'; R, for 'Retrieval'; and C, for 'Clustering'. These clusters are presented in Tables 10, 11, and 12, respectively.

Table 10: Cluster L "Learning": topics with membership values greater than 0.15.

u(t)	Code	Topic
0.300	5.2.3.8.	rule learning
0.282	5.2.2.1.	batch learning
0.276	5.2.1.1.2.	learning to rank
0.217	1.1.1.11.	query learning
0.216	5.2.1.3.3.	apprenticeship learning
0.213	1.1.1.10.	models of learning
0.203	5.2.1.3.5.	adversarial learning
0.202	1.1.1.14.	active learning
0.192	5.2.1.4.1.	transfer learning
0.192	5.2.1.4.2.	lifelong machine learning
0.189	1.1.1.8.	online learning theory
0.166	5.2.2.2.	online learning settings
0.159	1.1.1.3.	unsupervised learning and clustering

u(t)	Code	Topic
0.211	3.4.2.1.	query representation
0.207	5.1.3.2.1.	image representations
0.194	5.1.3.2.2.	shape representations
0.194	5.2.3.6.2.1	tensor representation
0.191	5.2.3.3.3.2	fuzzy representation
0.187	3.1.1.5.3.	data provenance
0.173	2.1.1.5.	equational models
0.173	3.4.6.5.	presentation of retrieval results
0.165	5.1.3.1.3.	video segmentation
0.155	5.1.3.1.2.	image segmentation
0.154	3.4.5.5.	sentiment analysis

 Table 11: Cluster R "Retrieval": topics with membership values greater than 0.15.

Table 12: Cluster C "Clustering": topics with membership values greater than 0.15.

u(t)	Code	Topic
0.327	3.2.1.4.7	biclustering
0.286	3.2.1.4.3	fuzzy clustering
0.248	3.2.1.4.2	consensus clustering
0.220	3.2.1.4.6	conceptual clustering
0.192	5.2.4.3.1	spectral clustering
0.187	3.2.1.4.1	massive data clustering
0.159	3.2.1.7.3	graph based conceptual clustering
0.151	3.2.1.9.2.	trajectory clustering

3.5 Results of lifting clusters L, R, and C within DST

All obtained clusters are lifted in the DST taxonomy using ParGenFS algorithm with the gap penalty $\lambda = 0.1$ and off-shoot penalty $\gamma = 0.9$.

The results of lifting Cluster L are shown in Figure 9. The cluster has received three head subjects: machine learning, machine learning theory, and learning to rank. These represent the structure of the general concept "Learning" according to our text collection. The list of gaps obtained is less instructive, reflecting probably a relatively modest coverage of the domain by the publications in the collection (see in Table 13).

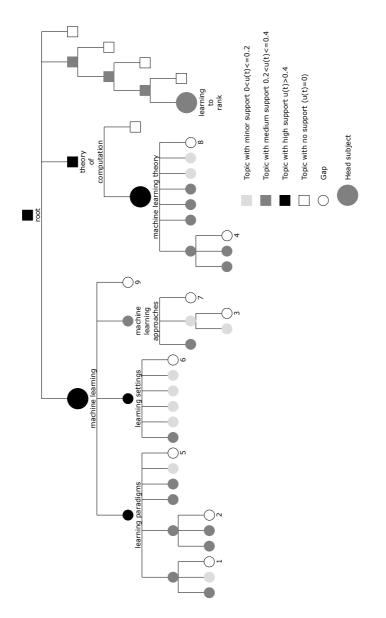


Figure 9: Lifting results for Cluster L: Learning. Gaps are numbered, see Table 13.

Table 13: Gaps at the lifting of Cluster L.

Number	Topics	
1	ranking, supervised learning by classification,	
	structured outputs	
2	sequential decision making in practice,	
	inverse reinforcement learning in practice	
3	statistical relational learning	
4	sequential decision making, inverse reinforcement learning	
5	unsupervised learning	
6	learning from demonstrations, kernel approach	
7	classification and regression trees, kernel methods,	
	neural networks, learning in probabilistic graphical models,	
	learning linear models, factorization methods,	
	markov decision processes, stochastic games,	
	learning latent representations, multiresolution,	
	support vector machines	
8	sample complexity and generalization bounds,	
	boolean function learning, kernel methods, boosting,	
	bayesian analysis, inductive inference,	
	structured prediction, markov decision processes,	
	regret bounds	
9	machine learning algorithms	

Similar comments can be made with respect to results of lifting of Cluster R: Retrieval (see Figure 10). The obtained head subjects: Information Systems and Computer Vision show the structure of "Retrieval" in the set of publications under considerations. Gaps obtained for Cluster R are listed in Table 14.

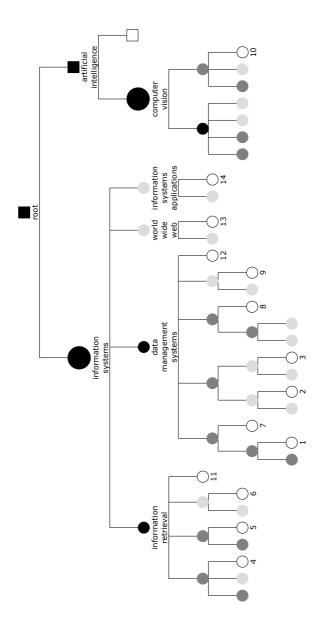


Figure 10: Lifting results for Cluster R. Gaps are numbered, see Table 14. The meaning of elements in the Figure is explained in the legend to Fig. 9.

Number Topics 1 data streams, incomplete data, temporal data, uncertainty, inconsistent data 2 multidimensional range search, point lookups, unidimensional range search, proximity search 3 data compression, record and block layout 4 query intent, query suggestion, query reformulation test collections, relevance assessment, retrieval effectiveness, 5 retrieval efficiency 6 question answering, document filtering, recommender systems, information extraction, expert search, clustering and classification, summarization, business intelligence 7 relational database model, entity relationship models, hierarchical data models, network data models, physical data models 8 relational database query languages, mapreduce languages, call level interfaces 9 data exchange, data cleaning, mediators and data integration, entity resolution, data warehouses 10 interest point and salient region detections, shape inference, object detection, object recognition, object identification, tracking, reconstruction, matching document representation, users and interactive retrieval, 11 retrieval models and ranking, specialized information retrieval 12database management system engines 13 site wrapping, data extraction and integration, traffic analysis, knowledge discovery 14 data cleaning, collaborative filtering, association rules, clustering, data stream mining, graph mining, process mining, text mining, data mining tools, sequence mining

Table 14: Gaps at the lifting of Cluster R.

We decided to show no drawing for the results of lifting of Cluster C, because the corresponding taxonomy fragment is too large, whereas the lifting results are too fragmentary. There are 16 (!) head subjects here:

- clustering,
- graph based conceptual clustering, item trajectory clustering,
- clustering and classification,
- unsupervised learning and clustering,
- spectral methods,
- document filtering,
- language models,

- music retrieval,
- collaborative search,
- database views,
- stream management,
- database recovery,
- mapreduce languages,
- logic and databases,
- language resources.

As one can see, the core clustering subjects are supplemented by methods and environments in the cluster – this shows that the ever increasing role of clustering activities perhaps should be better reflected in the taxonomy.

3.6 Making conclusions

We can see that the topic clusters found with the text collection do highlight areas of soon-to-be developments. Three clusters under consideration closely relate, in respect, to the following processes:

- theoretical and methodical research in learning, as well as merging the subject of learning to rank within the mainstream;
- representation of various types of data for information retrieval, and merging that with visual data and their semantics; and
- various types of clustering in different branches of the taxonomy related to various applications and instruments.

In particular, one can see from the "Learning" head subjects (see Figure 9 and comments to it) that main work here still concentrates on theory and method rather than applications. A good news is that the field of learning, formerly focused mostly on tasks of learning subsets and partitions, is expanding currently towards learning of ranks and rankings. Of course, there remain many subareas to be covered: these can be seen in and around the list of gaps in Table 13.

Moving to the lifting results for the information retrieval cluster R (see Figure 10 and Table 11), we can clearly see the tendencies of the contemporary stage of the process. Rather than relating the term "information" to

texts only, as it was in the previous stages of the process of digitalization, visuals are becoming parts of the concept of information. There is a catch, however. Unlike the multilevel granularity of meanings in texts, developed during millennia of the process of communication via languages in the humankind, there is no comparable hierarchy of meanings for images. One may only guess that the elements of the R cluster related to segmentation of images and videos, as well as those related to data management systems, are those that are going to be put in the base of a future multilevel system of meanings for images and videos. This is a direction for future developments clearly seen from the picture in Figure 10.

Regarding the "clustering" cluster C with its 16 (!) head subjects, one may conclude that, perhaps, a time moment has come or is to come real soon, when the subject of clustering must be raised to a higher level in the taxonomy to embrace all these "heads". At the beginning of the Data Science era, a few decades ago, clustering was usually considered a more-or-less auxiliary part of machine learning, the unsupervised learning. Perhaps, soon we are going to see a new taxonomy of Data Science, in which clustering is not just an auxiliary instrument but rather a model of empirical classification, a big part of the knowledge engineering. When discussing the role of classification as a knowledge engineering phenomenon, one encounters three conventional aspects of classification:

- structuring the phenomena;
- relating different aspects of phenomena to each other;
- shaping and keeping knowledge of phenomena.

Each of them can make a separate direction of research in knowledge engineering.

4 Conclusion

This paper presents a formalization of the concept of generalization, an important part of the human ability for conceptualization. According to Collins Dictionary, conceptualization is "formation (of a concept or concepts) out of observations, experience, data, etc." We assume that such an operation may require a coarser granularity of the domain structuring. This is captured by the idea of lifting a query set to higher ranks in a hierarchical taxonomy of the domain.

The hierarchical structure of taxonomy brings in possible inconsistencies between a query set and the taxonomy structure. These inconsistencies can be of either of two types, gaps or offshoots, potentially emerging at the coarser "head subject" to cover the query set. A gap is such a node of the taxonomy, that is covered by the head subject but does not belong in the query set. An offshoot is a node of the taxonomy, that does belong in the query set but is not covered by the head subject. The higher the rank of a candidate for the conceptual head subject, the larger the number of gaps. The lower is the rank of the head subject, the larger the number of offshoots. Our algorithm ParGenFS allows to find a globally optimal lifting to balance the numbers of head subjects, gaps, and offshoots depending on relative penalties for these types of inconsistencies.

We illustrate usefulness of this approach on the set of 17685 abstracts of research papers published by the Springer Publishers for 20 years, 1998– 2017, in 17 journals related to Data Science. We use this set to obtain six fuzzy clusters of taxonomy topics according to their co-relevance. We can easily interpret only three out of the found clusters, which probably reflects the inherent randomness of research processes and presence in it of an agenda not clearly manifested so far. It should be pointed out that finding interpretable clusters of taxonomy topics over the textual data requires using rather sophisticated methods involving spectral clustering, weighting publications, Laplacian transformation and the like. This complexity perhaps comes, at least partly, from the way we estimate the similarity between concepts – by considering them as just strings and, thus, by removing any imposed pre-structuring coming from pre-selected keywords or NLP constructions. Lifting of these clusters brings in several general conclusions over the current research in Data Science. These conclusions, even if not entirely unexpected, give as a glimpse into the hidden research processes, as captured by the authorship of the Springer journals.

The proposed approach to generalization can be used in a number of similar tasks such as positioning of a research project, interpretation of a concept which is not present in the taxonomy, annotation of a set of research articles. These all are parts of the processes of long-term research analysis and planning at which our approach should be positioned.

Among major issues requiring further development in this direction, two of the most relevant are:

- Taxonomy developments
- Specifying penalty weights

The former needs more attention both from research communities and planning committees. Specifically, most urgent directions for development here are:

- developing better methods to automate the process of taxonomy making (an interesting approach to this is described in [15]); and

- open discussion of the taxonomies at conferences and meetings of research communities and committees (the need in this is pointed out in a recent paper [25].

As to the latter, a reasonable computational progress over penalty weights can be achieved, in our view, by replacing the criterion of maximum parsimony by the criterion of maximum likelihood if each node of the taxonomy can be assigned probabilities of "gain" and "loss" of topic events. We intend pursuing this approach for the DST taxonomy in the near future.

Acknowledgment. Dmitry Frolov and Boris Mirkin acknowledge support by the Basic Research Program at the National Research University Higher School of Economics (HSE) and by the International Decision Choice and Analysis Laboratory (DECAN) NRU HSE in the framework of a subsidy granted to the HSE by the Government of the Russian Federation for the implementation of the Global Competitiveness Program.

5 References

- [1] The 2012 ACM Computing Classification System. [Online]. Available: http://www.acm.org/about/class/2012 (Accessed 2018, 30 April).
- [2] J. Ashraf, E. Chang, O. K. Hussain, F. K. Hussain, "Ontology usage analysis in the ontology lifecycle: A state-of-the-art review," *Knowledge-Based Systems*, vol. 80, pp. 34-47, 2015.
- [3] D. Beneventano, N. Dahlem, S. El Haoum, A. Hahn, D. Montanari, M. Reinelt, "Ontology-driven semantic mapping," *Enterprise Interop*erability III, Part IV, Springer, pp. 329-341, 2008.
- [4] J.C. Bezdek, R.J. Hathaway, M.P. Windham, "Numerical comparisons of the RFCM and AP algorithms for clustering relational data," *Pattern Recognition*, 24, pp. 783-791, 1991.
- [5] Blei D.M., Ng D.M., Jordan M.I. "Latent Dirichlet allocation," The Journal of Machine Learning Research, 3, pp. 993–1022, 2003.
- [6] D. Blei, "Probabilistic topic models," Communications of the ACM, 55 (4), pp. 77–84, 2012.
- [7] R.K. Brouwer, "A method of relational fuzzy clustering based on producing feature vectors using FastMap," *Information Sciences*, 179, pp. 3561-3582, 2009.
- [8] P. Buche, J. Dibie-Barthelemy, L. Ibanescu, "Ontology mapping using fuzzy conceptual graphs and rules," *ICCS Supplement*, pp. 17-24, 2008.
- [9] Chernyak, Ekaterina. "An approach to the problem of annotation of research publications." Proceedings of the eighth ACM international conference on web search and data mining, ACM, 429-434, 2015.
- [10] E. Chernyak, B. Mirkin, "Refining a Taxonomy by Using Annotated Suffix Trees and Wikipedia Resources," Annals of Data Science, 2(1), pp. 61-82, 2015.
- [11] M.J. Gacto, R. Alcalá, F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Information Sciences*, vol. 181, pp. 4340-4360, 2011.

- [12] Gene Ontology Consortium, "Gene ontology consortium: going forward," Nucleic Acids Research, vol. 43 D1049-D1056, 2015.
- [13] R. Grossi and J.S. Vitter, "Compressed suffix arrays and suffix trees with applications to text indexing and string matching," *SIAM Journal* on Computing, 35, 2 pp.378-407, 2005.
- [14] Kapur, S., Ayman O. F., and R. E. Chatwin. "Method and apparatus for representing text using search engine, document collection, and hierarchal taxonomy." U.S. Patent No. 7,580,926. 2009.
- [15] Klavans, Richard, and Kevin W. Boyack, "Which type of citation analysis generates the most accurate taxonomy of scientific and technical knowledge?", Journal of the Association for Information Science and Technology, 68(4), pp. 984-998, 2017.
- [16] D. Lee, R. Cornet, F. Lau, N. De Keizer, "A survey of SNOMED CT implementations," *Journal of Biomedical Informatics*, vol. 46, no. 1, pp. 87-96, 2013.
- [17] Lloret, E., Boldrini, E., Vodolazova, T., Martnez-Barco, P., Munoz, R., & Palomar, M. "A novel concept-level approach for ultra-concise opinion summarization", *Expert Systems with Applications*, 42(20), pp. 7148-7156, 2015.
- [18] U. von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, pp. 395-416, 2007.
- [19] J.P. Mei, Y. Wang, L. Chen, and C. Miao, "Large scale document categorization with fuzzy clustering," *IEEE Transactions on Fuzzy Sys*tems, 25(5), 1239-1251, 2017.
- [20] B. Mirkin, *Clustering: A Data Recovery Approach*, Chapman and Hall/CRC Press, 2012.
- [21] Mirkin B. G., Chernjak E. L., Chugunova O. N. Metod annotirovannogo suffiksnogo dereva dlja ocenki stepeni vhozhdenija strok v tekstovye dokumenty. [Annotated Suffix Tree as a Way of String-To-Document Score Evaluating]. Business Informatics, 3 (21), pp. 31–41, 2012 (in Russian).

- [22] B. Mirkin, S. Nascimento, "Analysis of Community Structure, Affinity Data and Research Activities using Additive Fuzzy Spectral Clustering," Technical Report 6, School of Computer Science, Birkbeck University of London, 2009.
- [23] B. Mirkin, S. Nascimento, "Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices," *Information Sciences*, vol. 183, no. 1, pp. 16-34, 2012.
- [24] B. Mirkin, M. Orlov, "Three aspects of the research impact by a scientist: measurement methods and an empirical evaluation." In Optimization, Control, and Applications in the Information Age, Springer, Cham, pp. 233-259, 2015.
- [25] F. Murtagh, M. Orlov, & B. Mirkin, "Qualitative judgement of research impact: Domain taxonomy as a fundamental framework for judgement of the quality of research." *Journal of Classification*, 35(1), pp. 5-28, 2018.
- [26] Mueller G, Bergmann R. "Generalization of Workflows in Process-Oriented Case-Based Reasoning,", In FLAIRS Conference, pp. 391-396, 2015.
- [27] D. Nallaperuma & D. De Silva, "A participatory model for multidocument health information summarisation," Australasian Journal of Information Systems, 21.
- [28] S. Nascimento, T. Fenner, B. Mirkin, "Representing research activities in a hierarchical ontology," in *Procs. of 3rd International Workshop on Combinations of Intelligent Methods and Applications* (CIMA 2012), Montpellier, France, August, 28, pp. 23-29, 2012.
- [29] F. Osborne, A. Salatino, A. Birukou, E. Motta, "Automatic Classification of Springer Nature Proceedings with Smart Topic Miner," In: Groth P. et al. (Eds) *The Semantic Web* – ISWC 2016. *Lecture Notes in Computer Science*, vol 9982, pp. 383-399. Springer, Cham, 2016.
- [30] Pampapathi R., Mirkin B., Levene M. "A suffix tree approach to antispam email filtering," *Machine Learning*, 65(1), pp. 309–338, 2006.
- [31] A. M. Rinaldi, "An ontology-driven approach for semantic information retrieval on the Web," ACM Trans. on Internet Technologies, vol. 9, no.3, article 10, 2009.

- [32] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Journal Foundations and Trends in Information Retrieval*, 3(4), pp. 333–389, 2009.
- [33] P. Robinson and S. Bauer, Introduction to Bio-Ontologies, Chapman and Hall/CRC Press, 2011.
- [34] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 25, no 5, pp. 513–523, 1998.
- [35] A.P. Santos and F. Rodrigues, "Multi-Label Hierarchical Text Classification Using the ACM Taxonomy," Proceedings of 14th Portuguese Conference on Artificial Intelligence, (Aveiro, Portugal, October 12– 15), pp. 553-564, 2010.
- [36] F. Sebastiani, "Machine learning in automated text categorization," Journal of ACM Computing Surveys, 1, 34, pp. 1-42, 2002.
- [37] R.N. Shepard and P. Arabie, "Additive clustering: representation of similarities as combinations of discrete overlapping properties," *Psychological Review*, vol. 86, pp. 87-123, 1979.
- [38] R. Snow, D. Jurafsky, and A.Y. Ng, Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 801-808, 2006.
- [39] Song, Y., Liu, S., Wang, H., Wang, Z., & Li, H. "Automatic taxonomy construction from keywords," U.S. Patent No. 9,501,569. Washington, DC: U.S. Patent and Trademark Office, 2016.
- [40] A.R. de Soto, "A hierarchical model of a linguistic variable," Information Sciences, vol. 181, pp. 4394-4408, 2011.
- [41] M. Usman, R. Britto, J. Boerstler, and E. Mendes, "Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method," *Information and Software Technology*, 85, pp. 43-59, 2017.

- [42] N. Vedula, P.K. Nicholson, D. Ajwani, S. Dutta, A. Sala, and S. Parthasarathy, "Enriching Taxonomies With Functional Domain Knowledge," In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ACM, pp. 745-754, 2018.
- [43] C. Wang, D.M. Blei, "Collaborative topic modeling for recommending scientific articles," InProceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 448-456, 2011.
- [44] J. Waitelonis, C. Exeler, and H. Sack, "Linked data enabled generalized vector space model to improve document retrieval," In *Proceedings of NLP & DBpedia 2015 workshop in conjunction with 14th International Semantic Web Conference (ISWC)*, CEUR-WS, vol. 1486, 2015.
- [45] Wang, C., He, X., & Zhou, A. "A Short Survey on Taxonomy Learning from Text Corpora: Issues, Resources and Recent Advances," In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1190-1203, 2017.

6 APPENDIX: Taxonomy of Data Science according to CCM-CCS 2012

The leaves added by the authors are labeled with a star ``*".

Index	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
1.	Theory of					
	computa-					
	tion					
1.1.		Theory				
		and al-				
		gorithms				
		for ap-				
		plication				
		domains				
1.1.1.			Machine			
			learning			
			theory			
1.1.1.1				Sample complexity and		
				generalization bounds		
1.1.1.2.				Boolean function		
1.1.1.2.				learning		
1.1.1.3.				Unsupervised learning		
1.1.1.3.				and clustering		
1.1.1.4.				Kernel methods		
1.1.1.4. 1.1.1.4.1.				Kernel methods	Support votor	
1.1.1.4.1.					Support vector ma-	
1.1.1.4.2.					chines	
1.1.1.4.2. 1.1.1.4.3.*					Gaussian processes	
1.1.1.4.3.* 1.1.1.5.				Bti	Modelling	
				Boosting		
1.1.1.6.				Bayesian analysis		
1.1.1.7.				Inductive inference		
1.1.1.8.				Online learning theory		
1.1.1.9.				Multi-agent learning		
1.1.1.10.				Models of learning		
1.1.1.11.				Query learning		
1.1.1.12.				Structured prediction		
1.1.1.13.				Reinforcement learning		
1.1.1.13.1.					Sequential decision	
					making	
1.1.1.13.2.					Inverse reinforce-	
					ment learning	
1.1.1.13.3.					Apprenticeship	
					learning	
1.1.1.13.4.					Multi-agent rein-	
					forcement learning	
1.1.1.13.5.					Adversarial learning	
1.1.1.14.				Active learning	l	
1.1.1.15.				Semi-supervised learn-		
				ing		
1.1.1.16.				Markov decision pro-		
				cesses		
1.1.1.17.				Regret bounds		
1.1.2.			Database			
			theory			
1.1.2.1.				Data exchange		
1.1.2.2.				Data provenance		
1.1.2.3.				Data modeling		
1.1.2.3.				Database query lan-		
1.1.2.4.				guages (principles)		
1.1.2.5.						
1.1.2.5.				Database constraints		
1100				theory		
1.1.2.6.				Database interoper-		
				ability		
1.1.2.7.				Data structures and al-		
				gorithms for data man-		
1				agement		

1.1.2.8. Database query pro- cessing and optimiza- tota databases 1.1.2.0. Incomplete inconsi- tics of comput- ing Probability and statistics 2.1. Mathema- tics of comput- ing Probability and statistics 2.1.1. Probability tics Probability and statistics 2.1.1. Probability tics Probability and statistics 2.1.1. Probability tics Probability and statistics 2.1.1. Probability tics Probability and statistics 2.1.1.6. Probability tics Probability and statistics 2.1.1.7. Probability tics Probability and statistics 2.1.1.8.1 Probability tics Probability and statistics 2.1.2.1. Probability tics Probability and statistics 2.1.2.1. Probability tics Probability tics 2.1.2.1. Probability tics Probabil						
1.1.2.0 1.1.2.10. 1.1.2.10. 1.1.2.11Mathematics of comput- tics o	1.1.2.8.				Database query pro-	
1.1.2.0 1.1.2.10. 1.1.2.10. 1.1.2.11Mathematics of comput- tics o					cessing and optimiza-	
1.1.2.9. 1.1.2.10. 1.1.2.10. I.1.2.10. 1.1.2.11. I.1.2.10. 2. Mathema- tics of comput- ing Probability and statis- tics Probability and statis- tics 2.1. I.1.1 2.1.1.1 I.1.2.10. 2.1.1.1 Probability and statis- tics Probability and statis- tics 2.1.1.1 Probability and statis- tics Probability and statis- tics 2.1.1.3. Probability and statis- tics Probability and statis- tics 2.1.1.4. Probability and statis- tics Probability and statis- pactor graphs 2.1.1.5. Probability and statis- tics Probability and statis- stations 2.1.1.8.1 Probability and statis- stations Kernel density esti- mators 2.1.2.1 Probability and statis- stations Kernel density esti- mators 2.1.2.1. Probability and statis- stations Kernel density esti- mators 2.1.2.1. Probability and statis- station Kernel density esti- mators 2.1.2.1. Probability and statis- statis Kernel density esti- mators 2.1.2.1. Probability and statis Kernel density esti- mators 2.1.2.2. Probability and statis						
1.1.2.10. L.2.10. 1.1.2.11. L.2.10. 1.1.2.11. Enclashing accuring incomplete, incomale tent, and uncertain databases privacy and security incomplete, incomale tent, and uncertain databases Enclashing accuring incomplete, incomale tent, and uncertain databases 2.1. Enclashing accuring incomplete, incomale tent, and uncertain databases Enclashing accuring incomplete, incomale tent, and uncertain databases 2.1.1. Enclashing accuring incomplete, incomale tent, and uncertain databases Enclashing accuring incomplete, incomale tent, and uncertain databases 2.1.1.1. Enclashing accuring incomplete, incomale tent, and uncertain databases Enclashing accuring incomele tent, and uncertain databases 2.1.1.1. Enclashing accuring incomele tent, and uncertain databases Enclashing accuring accuring incomele tent, and uncertain databases 2.1.1.1. Enclashing accuring accuring incomele tent, and uncertain databases Enclashing accuring	1 1 2 0					
1.1.2.11. Image: Second se						
1.12.12. 1.12.12.Mathema results ingProbability and statis- tice representationsprivacy and security ingprivacy and security ing2.1.Probability and statis- tice representationsProbability- tice representationsprobability- sentationsprobability- sentations2.1.1.Probability- tice representationsBayesian networks Factor graphs Decision diagrams Equational modelskernel density estimators spline models2.1.1.8.Probability- tice representationsBayesian networks Pactor graphs Decision diagrams Equational models2.1.1.8.Probability- tice representationsKernel density estimators spline models2.1.1.8.Probability- tice representationsKernel density estimators spline models2.1.2.1.Probability- tice representationsMaximum likelihood configureanterior Density estimation Density estimation Bayesian computation Density estimation Bayesian computation Density estimation adory belief problems2.1.2.1.Probability- tice representation algorithmsMaximum likelihood configureanterion Density estimation Density estimat						
1.1.2.12. Mathematics of compute compute incomplete, inconsistent of the complete, inconsistent of the compute incomplete, inconsistent of the compute incomplete, inconsistent of the complete, inconsi	1.1.2.11.					
2.Mathem tics of comput- ingProbability and statis- ticsProbability and statis- ticsLent, 'and uncertain databases2.1.Probability ticsProbability sentationsBayesian networks Pactor graphsHermiter in the sentations Bayesian networks Pactor graphs2.1.1.1. 2.1.1.2. 2.1.1.3.Probability ticsBayesian networks Pactor graphsKernel density estimation Stochastic differential equations2.1.1.8.Probability sentationsProbability sentationsKernel density estimation patient in models2.1.1.8.Probability stimuter in the sentationsMaximum likelihood sentationsKernel density estimators spline medis patient in compara- metric models2.1.2.1.Probability sentation 2.1.2.2.Probability ference problemsMaximum likelihood computation Datainon Datainon tationQuantile regression2.1.2.1.Probability sentation Pactors 2.1.2.4.Probability ference soning algorithmMaximum likelihood computation Datainon Variable elimination Max marginal compu- tationQuantile regression2.1.3.6.Probability senting algorithm sinuit computation Carlo methodsGibbs sampling Matro-chain Motte Carlo methods2.1.3.6.Probability senting algorithm sinuitation 2.1.3.5.1Fibe senting algorithm senting algorithm Sinuitated annealing Matro-chain Motte Carlo methodsGibbs sampling Matro-chain Motte measures2.1.3.6.Probability senting algorithm Sinuitated annealing Matro-chain Motte<					privacy and security	
2.Mathem tics of comput- ingProbability and statis- ticsProbability and statis- ticsLent, 'and uncertain databases2.1.Probability ticsProbability sentationsBayesian networks Pactor graphsHermiter in the sentations Bayesian networks Pactor graphs2.1.1.1. 2.1.1.2. 2.1.1.3.Probability ticsBayesian networks Pactor graphsKernel density estimation Stochastic differential equations2.1.1.8.Probability sentationsProbability sentationsKernel density estimation patient in models2.1.1.8.Probability stimuter in the sentationsMaximum likelihood sentationsKernel density estimators spline medis patient in compara- metric models2.1.2.1.Probability sentation 2.1.2.2.Probability ference problemsMaximum likelihood computation Datainon Datainon tationQuantile regression2.1.2.1.Probability sentation Pactors 2.1.2.4.Probability ference soning algorithmMaximum likelihood computation Datainon Variable elimination Max marginal compu- tationQuantile regression2.1.3.6.Probability senting algorithm sinuit computation Carlo methodsGibbs sampling Matro-chain Motte Carlo methods2.1.3.6.Probability senting algorithm sinuitation 2.1.3.5.1Fibe senting algorithm senting algorithm Sinuitated annealing Matro-chain Motte Carlo methodsGibbs sampling Matro-chain Motte measures2.1.3.6.Probability senting algorithm Sinuitated annealing Matro-chain Motte<	1.1.2.12.				Incomplete, inconsis-	
2.Mathematics of computing ingRobalilie and statisticsIdatabasesIdatabases2.1.1Indication itesProbabilie tic representationBayesian networks Markov networks Markov networks Stochastic differential equational models Causal networks Stochastic differential equationsKernel density esti- mators Spline models and statistics2.1.1.8.1Interpret itesProbabilis tic representationsKernel density esti- mators Spline models2.1.1.8.1Interpret itesProbabilis tic representationsKernel density esti- mators2.1.2.1Interpret itesProbabilis tic representationsKernel density esti- mators2.1.2.1Interpret itesProbabilis tic representationsKernel density esti- mators2.1.2.1Interpret itesProbabilis tic representation Computing met probabilis tic representation Bayesian computation Computing met probabilie tic representation Bayesian computation Computing met probabilie tic representation Bayesian computation Computing met probabilie tic representation Markov-chain Motte Kernet density estimation Markov-chain Motte (and confidence tion density estimation <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
2. Mathematics of comput- ing Probability and statis- tics Probability and statis- tics Probability and statis- tics Probability and statis- tics Probability and statis- tics Probability and statis- tics Probability and statis- sentations Bayesian networks Markov networks Decision diagrams Probability Bayesian networks 2.1.1.1 Probability and statis- sentations Probability Bayesian networks Probability Bayesian networks 2.1.1.8 Probability and statis- tics Probability Bayesian networks Probability Bayesian computations 2.1.1.8 Probability and statis- tics Probability Bayesian computations Probability Bayesian computations 2.1.2.1 Probability and statis- tics Probability bits Probability Bayesian computations 2.1.2.1 Probability and statis- tics Probability bits Probability bits Probability bits 2.1.2.1 Probability and statis- tics Probability bits Probability bits Probability bits 2.1.2.1 Probability and statis- tics Probability bits Probability bits Probability bits 2.1.2.5 Probability and probability Probability bits Probability bits Probability bits 2.1.3.5 Probability bits Probability bit						
tics of comput- ingProbability and statis- ticsProbabilis- sentationsBayesian networks pactor graphs Calculational models Calculational models CalculationsKernel density esti- mators Spline models Bayesian networks stational models Calculations2.1.1.2.1.1.3.1.1.4.1.1.5.2.1.1.4.1.1.4.1.1.5.2.1.1.5.1.1.4.1.1.6.2.1.1.6.1.1.6.1.1.6.2.1.1.8.2.1.1.6.1.1.6.2.1.1.8.3.1.1.6.1.1.6.2.1.2.4.1.1.6.1.1.6.2.1.2.5.1.1.6.1.1.6.2.1.2.4.1.1.6.1.1.6.2.1.2.5.1.1.6.1.1.6.2.1.2.6.1.1.6.1.1.6.2.1.2.7.1.1.6.1.1.6.2.1.2.8.1.1.6.1.1.6.2.1.2.1.1.1.6.1.1.6.2.1.2.3.1.1.6.1.1.6.2.1.2.4.1.1.6.1.1.6.2.1.2.5.1.1.6.1.1.6.2.1.2.6.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.1.1.1.6.1.1.6.2.1.3.2.1.1.6.1.1.6.2.1.3.3.1.1.6.1.1.6.2.1.3.4.1.1.6.1.1.6.2.1.3.5.1.1.1.6.1.1.6.2.1.3.6.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6. <td< td=""><td></td><td></td><td></td><td></td><td>databases</td><td></td></td<>					databases	
2.1.computing ingProbability and statis- ticsProbabilisity itic representationsProbabilisity itic representationsProbabilisity Bayesian networks Bayesian networks Bayesian networks Bayesian networks Causain networks Causain networks Causain networks Causain networks Causain networks Splinke models Causain networks Splinke models Causain networks Splinke models Splinke models Spl	2.					
ingingProbability and statis- ticsProbabilis sentationsBayesian networks Bayesian networks Factor graphs Equational models Calual networks Stochastic differential equationsKernel density esti- mators Spline models Bayesian networks adata and the sentations2.1.1.1. 2.1.1.3.1.1.6.1.1.7.2.1.1.8.1.1.8.1.1.8.2.1.1.8.1. 2.1.1.8.3.1.1.8.1.1.6.2.1.2.1.2.1.2.1.1.8.1.1.6.2.1.2.2. 2.1.2.3.1.1.6.1.1.6.2.1.2.4.1.1.6.1.1.6.2.1.2.5. 2.1.2.6.1.1.6.1.1.6.2.1.3.1. 2.1.3.1.1.1.6.1.1.6.2.1.3.2.1.1.6.1.1.6.2.1.3.3.1.1.6.1.1.6.2.1.3.4.1.1.6.1.1.6.2.1.3.5.1. 2.1.3.5.1.1.1.6.1.1.6.2.1.3.6.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.2.1.1.1.6.1.1.6.2.1.2.2.1.1.6.1.1.6.2.1.3.1.1.1.6.1.1.6.2.1.3.2.1.1.6.1.1.6.2.1.3.3.1.1.6.1.1.6.2.1.3.4.1.1.6.1.1.6.2.1.3.5.1.1.1.6.1.1.6.2.1.3.6.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.2.1.3.7. <td></td> <td>tics of</td> <td></td> <td></td> <td></td> <td></td>		tics of				
ingingProbability and statis- ticsProbabilis sentationsBayesian networks Bayesian networks Factor graphs Equational models Calual networks Stochastic differential equationsKernel density esti- mators Spline models Bayesian networks adata and the sentations2.1.1.1. 2.1.1.3.1.1.6.1.1.7.2.1.1.8.1.1.8.1.1.8.2.1.1.8.1. 2.1.1.8.3.1.1.8.1.1.6.2.1.2.1.2.1.2.1.1.8.1.1.6.2.1.2.2. 2.1.2.3.1.1.6.1.1.6.2.1.2.4.1.1.6.1.1.6.2.1.2.5. 2.1.2.6.1.1.6.1.1.6.2.1.3.1. 2.1.3.1.1.1.6.1.1.6.2.1.3.2.1.1.6.1.1.6.2.1.3.3.1.1.6.1.1.6.2.1.3.4.1.1.6.1.1.6.2.1.3.5.1. 2.1.3.5.1.1.1.6.1.1.6.2.1.3.6.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.2.1.1.1.6.1.1.6.2.1.2.2.1.1.6.1.1.6.2.1.3.1.1.1.6.1.1.6.2.1.3.2.1.1.6.1.1.6.2.1.3.3.1.1.6.1.1.6.2.1.3.4.1.1.6.1.1.6.2.1.3.5.1.1.1.6.1.1.6.2.1.3.6.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.1.1.6.2.1.3.7.1.1.6.2.1.3.7. <td></td> <td>comput-</td> <td></td> <td></td> <td></td> <td></td>		comput-				
2.1. Probability and statistics 2.1.1. Probabilistic representations 2.1.1.1 Probabilistic representations 2.1.1.1 Probabilistic representations 2.1.1.3 Probabilistic representations 2.1.1.4 Probabilistic representations 2.1.1.5 Probabilistic representations 2.1.1.6 Probabilistic representations 2.1.1.8.1 Probabilistic representations 2.1.1.8.2 Probabilistic representations 2.1.2.1 Probabilistic representations 2.1.2.1 Probabilistic representations 2.1.2.2 Probabilistic representation 2.1.2.3 Probabilistic representation 2.1.2.4 Probabilistic representation 2.1.2.5 Probabilistic representation 2.1.2.6 Probabilistic representation 2.1.3.6 Probabilistic representation 2.1.3.5 Probabilistic representation 2.1.3.6 Probabilistic representation 2.1.3.6 Probabilistic representation 2.1.3.7 Probabilistic representation 2.1.3.6 Probabilistic representation 2.1.3.6 Proba						
2.1.1.and statistic ticsProbabilis- tic representationsBayesian networks Markov networks Pactor graphs Decision diagrams Decision diagrams Decision Decis		ing				
2.1.1.1ticsProbabilis- tic representationsBayesian networks Bayesian networks Factor graphs Causal networks Stochastic differential equations Nonparametric representationsKernel density esti- mators Spline models Difference Spline models Difference Spline models Bayesian computation Computing models Difference Spline models Bayesian computation Computing models Difference Probabilis- tic in- hference 2.1.2.1.2.1.2.2.Kernel density esti- mators Spline models Bayesian nonpara- metric models2.1.2.1 2.1.2.2.Probabilis- tic in- inference problemsComputing models estimation Bayesian computation Density estimation Density estimation Density estimation Variable elimination Loopy belief propaga- tion Variable elimination Loopy belief propaga- tion Markov-chain Monte Carlo methodsGibbs sampling Metropolie-Matings Bayerian Bayesian Bayesian difference tion Markov-chain Monte Carlo methods21.3.6.Li.3.6.Sequential Monte Carlo methodsGibbs sampling Metropolie-Matings Bayerian Monte Carlo methods	2.1.					
2.1.1. Image: Section of Section Probabilistic representations Bayesian networks 2.1.1.1 Section diagrams Bayesian networks 2.1.1.2 Factor graphs Decision diagrams 2.1.1.3. Stochastic differential equations Spline models 2.1.1.8. Nonparametric representations Spline models 2.1.1.8.1. Probabilistic inference Spline models 2.1.1.8.2. Probabilistic inference Spline models 2.1.2.1 Probabilistic inference Spline models 2.1.2.1 Probabilistic inference Spline models 2.1.2.1 Decision scienterval Spline models 2.1.2.1 Density estimation Spline models 2.1.2.2 Density estimation Spline models 2.1.2.3. Density estimation Spline models 2.1.2.4. Density estimation Spline models 2.1.2.5. Spline models Spline models 2.1.2.6. Spline models Spline models 2.1.3.1. Spline models Spline models 2.1.3.1. Spline models Spline models 2.1.3.5. Spline models						
2.1.1.1. 1.1.2. 2.1.1.1. Bayesian networks 2.1.1.3. Markov networks 2.1.1.3. Causal networks 2.1.1.4. Causal networks 2.1.1.5. Causal networks 2.1.1.6. Stochastic differential equations 2.1.1.8. Stochastic differential equations 2.1.1.8.1 Probabilistic 2.1.1.8.2 Frobabilistic 2.1.2.1 Computing models 2.1.2.1. Splite models 2.1.2.1. Splite models 2.1.2.1. Computing most probabilistic 2.1.2.2 Computation 2.1.2.3. Probabilistic 2.1.2.4. Probabilistic 2.1.2.5 Computation 2.1.2.6. Probabilistic 2.1.3.1 Probabilistic 2.1.3.2. Probabilistic 1.3.3. Probabilistic 1.3.4. Probabilistic 1.3.5. Gibbs sampling 2.1.3.1. Markov-chain Monte 2.1.3.4. Probabilistic 2.1.3.5. Gibbs sampling 2.1.3.6. Sequential Monte Carlo methods Gibbs sampling Markov-chain Monte Carlo methods Carlo methods Gibbs			tics			
2.1.1.1. 1.1.2. 2.1.1.1. Bayesian networks 2.1.1.3. Markov networks 2.1.1.3. Causal networks 2.1.1.4. Causal networks 2.1.1.5. Causal networks 2.1.1.6. Stochastic differential equations 2.1.1.8. Stochastic differential equations 2.1.1.8.1 Probabilistic 2.1.1.8.2 Frobabilistic 2.1.2.1 Computing models 2.1.2.1. Splite models 2.1.2.1. Splite models 2.1.2.1. Computing most probabilistic 2.1.2.2 Computation 2.1.2.3. Probabilistic 2.1.2.4. Probabilistic 2.1.2.5 Computation 2.1.2.6. Probabilistic 2.1.3.1 Probabilistic 2.1.3.2. Probabilistic 1.3.3. Probabilistic 1.3.4. Probabilistic 1.3.5. Gibbs sampling 2.1.3.1. Markov-chain Monte 2.1.3.4. Probabilistic 2.1.3.5. Gibbs sampling 2.1.3.6. Sequential Monte Carlo methods Gibbs sampling Markov-chain Monte Carlo methods Carlo methods Gibbs	2.1.1.			Probabilis-		
2.1.1.1sentationsBayesian networks Markov networks Factor graphs Decision diagrams Equational models Causal networks Stochastic differential equationsKernel density esti- mators Spline models Bayesian nonpara- metric models2.1.1.6. 2.1.1.8.1Probabilis- tic ferners problemsMarkov networks Stochastic differential equationsKernel density esti- mators Spline models Bayesian nonpara- metric models2.1.2.1Probabilis- tic ferners problemsMaximum likelihood estimation Computation Density estimation Loopy belief propaga- tion2.1.2.4.Probabilis- tic reasoning algorithmsMaximum likelihood estimation Loopy belief propaga- tion2.1.2.5. 2.1.2.5.1Probabilis- tic reasoning algorithmsMaximum likelihood estimation Loopy belief propaga- tion2.1.3.5.Probabilis- tic tic reasoning algorithmsGibbs sampling Markov-chain Monte Carlo methods Expectation maximiza- ton Carlo methods2.1.3.5.1 2.1.3.5.1Sequential Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo methods2.1.3.6.Sequential Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo methods						
2.1.1.1. Bayesian networks 2.1.1.2. Markor networks 2.1.1.3. Probabilistic 2.1.1.6. Causal networks 2.1.1.7. Stochastic differential equations 2.1.1.8.1 Nonparametric representations 2.1.1.8.1 Nonparametric representations 2.1.1.8.3. Maximum likelihood estimation 2.1.2.1. Maximum likelihood estimation 2.1.2.1. Maximum likelihood estimation 2.1.2.1. Maximum likelihood estimation 2.1.2.2. Maximum likelihood estimation 2.1.2.3. Maximum likelihood estimation 2.1.2.4. Maximum likelihood estimation 2.1.2.5. Maximum likelihood estimation 2.1.3.6. Probabilistic reassoning algorithms 2.1.3.1. Maximum likelihood estimation 2.1.3.2. Probabilistic reassoning algorithms 2.1.3.3.1. Yariable elimination 2.1.3.5. Markor-chain Monte Carlo methods Simulated annealing Markor-chain Monte Carlo methods Simulate annealing Markor-chain Monte Carlo methods Simulate annealing Markor-chain Monte 2.1.3						
2.1.1.2, Markov networks 2.1.1.3, Pactor graphs 2.1.1.4, Decision diagrams 2.1.1.5, Equational models 2.1.1.6, Stochastic differential 2.1.1.8.1 Nonparametric representations 2.1.1.8.1 Nonparametric representations 2.1.1.8.1 Nonparametric representations 2.1.2.1 Probabilistic inference 2.1.2.1 Markov networks 2.1.2.1 Probabilistic inference 2.1.2.1 Computing most probable sexplanation 2.1.2.2. Probabilistic representation 2.1.2.3. Probabilistic representation 2.1.2.4. Probabilistic representation 2.1.2.5. Naximum likelihood 2.1.2.6. Probabilistic representation 2.1.3.1 Probabilistic representation 2.1.3.2. Variable elimination 2.1.3.4. Probabilistic representation 2.1.3.5. Gibbs sampling Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo methods 2.1.3.5. Simulated annealing Markov-chain Monte Carlo methods 2.1.3.5.4. Sequential Monte Carlo methods 2.1.				sentations		
2.1.1.3, 2.1.1.4, 2.1.1.4, Decision diagrams 2.1.1.6, Causal networks 2.1.1.7, Equational models 2.1.1.8, Stochastic differential equations 2.1.1.8.1. Nonparametric representations 2.1.1.8.3. Probabilistic inference 2.1.2.1. Maximum likelihood estimation 2.1.2.2. Maximum likelihood estimation 2.1.2.3. Density estimation 2.1.2.4. Maximum likelihood estimation 2.1.2.5. Density estimation 2.1.2.6. Max marginal computation 2.1.3.6. Probabilistic reasoning algorithms variable elimination Max marginal compation Carlo methods Expectation maximization 2.1.3.5. Max marginal compation 2.1.3.5. Markov-chain Monte Carlo methods Expectation maximization Markov-chain Monte Gibbs sampling Markov-chain Monte Carlo methods Simulated annealing Markov-chain Monte Carlo methods Simulated annealing Markov-chain Monte Carlo methods Simulates and hid						
2.1.1.3, 2.1.1.4, 2.1.1.4, Decision diagrams 2.1.1.6, Causal networks 2.1.1.7, Equational models 2.1.1.8, Stochastic differential equations 2.1.1.8.1. Nonparametric representations 2.1.1.8.3. Probabilistic inference 2.1.2.1. Maximum likelihood estimation 2.1.2.2. Maximum likelihood estimation 2.1.2.3. Density estimation 2.1.2.4. Maximum likelihood estimation 2.1.2.5. Density estimation 2.1.2.6. Max marginal computation 2.1.3.6. Probabilistic reasoning algorithms variable elimination Max marginal compation Carlo methods Expectation maximization 2.1.3.5. Max marginal compation 2.1.3.5. Markov-chain Monte Carlo methods Expectation maximization Markov-chain Monte Gibbs sampling Markov-chain Monte Carlo methods Simulated annealing Markov-chain Monte Carlo methods Simulated annealing Markov-chain Monte Carlo methods Simulates and hid	2.1.1.2.				Markov networks	
2.1.1.4. 2.1.1.5. 2.1.1.5. Equational models 2.1.1.6. Stochastic differential 2.1.1.8. Nonparametric representations Stochastic differential Stochastic differential equations Nonparametric representations 2.1.1.8.1. Probabilis- 2.1.1.8.3. Probabilis- 2.1.2.1 Probabilis- 2.1.2.1 Maximum likelihood 2.1.2.1 Probabilis- 2.1.2.1 Probabilis- 2.1.2.2 Probabilis- 2.1.2.3. Probabilis- 2.1.2.4. Probabilis- 2.1.2.5. Probabilis- 2.1.2.6. Probabilis- 2.1.2.6. Probabilis- 2.1.3.1 Probabilis- 2.1.3.2. Probabilis- 2.1.3.4. Probabilis- 2.1.3.5. Probabilis- 2.1.3.6. Probabilis- 2.1.3.7. Probabilis- 2.1.3.6. Probabilis- 2.1.3.7. Probabilis- 2.1.3.6. Probabilis- 2.1.3.7. Probabilis-			1			
2.1.1.5. 2.1.1.6. 2.1.1.6. 2.1.1.6. 2.1.1.7. Causal networks 2.1.1.8. Stochastic differential equations 2.1.1.8.1. Nonparametric representations 2.1.1.8.2. Probabilistic inference problems 1.1.8.3. Probabilistic inference problems 2.1.2.1 Maximum likelihood estimation 2.1.2.2. Probabilistic representations 2.1.2.3. Maximum likelihood estimation 2.1.2.4. Output in most probabilistic representation 2.1.2.5. Hypothesis testing and confidence interval computation Density estimation 2.1.2.6. Probabilistic reasoning algorithms 2.1.3.1. Probabilistic reasoning algorithms 2.1.3.1. Variable elimination Loopy belief propagation 2.1.3.4. Variable elimination Markov-chain Monte Carlo methods 2.1.3.5. Carlo methods 2.1.3.5.1. Simulated annealing Metropolis-Hastings algorithm 2.1.3.6. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo convergence measures						
2.1.1.6. 2.1.1.7. 2.1.1.8. Stochastic differential equations 2.1.1.8.1. Nonparametric representations 2.1.1.8.1. Probabilistic inference 2.1.1.8.2. Probabilistic inference 2.1.2.1. Maximum likelihood estimation 2.1.2.1. Maximum likelihood estimation 2.1.2.2. Maximum likelihood estimation 2.1.2.3. Maximum likelihood estimation 2.1.2.4. Maximum likelihood estimation 2.1.2.5. Density estimation 2.1.2.6. Maximum likelihood estimation 2.1.2.6. Density estimation 2.1.3.1 Probabilistic reasion 2.1.3.2. Probabilistic reasion 2.1.3.3. Probabilistic reasion 2.1.3.4. Variable elimination 2.1.3.5. Variable elimination 2.1.3.5. Variable elimination 2.1.3.5. Maximum Maximum Maximization 2.1.3.5. Maximum Maximization 2.1.3.5. Spline addisention 2.1.3.5. Spline addisention 2.1.3.6. Sequential Monte 2.1.3.7. Sequential Monte						
2.1.1.7. Stochastic differential equations Frobabilistic constraints Kernel density estimators 2.1.1.8.1. Probabilistic inference Nonparametric representations Spline models 2.1.1.8.3. Probabilistic inference Maximum likelihood Bayesian computation 2.1.2.1. Maximum likelihood Bayesian computation Computing most probabilistics transmant 2.1.2.2. Maximum likelihood Maximum likelihood Guantile regression 2.1.2.3. Probabilistic reasoning algorithms Density estimation Quantile regression 2.1.2.4. Probabilistic reasoning algorithms Variable elimination Quantile regression 2.1.2.6. Probabilistic reasoning algorithms Variable elimination Quantile regression 2.1.3.1. Probabilistic reasoning algorithms Variable elimination Gibbs sampling 2.1.3.5. Probabilistic reasoning algorithms Simulated annealing Markov-chain Monte 2.1.3.5. Probabilistic Carlo methods Gibbs sampling 2.1.3.5. Probabilistic Carlo methods Carlo convergence 2.1.3.5. Probabilistic Carlo methods Carlo convergence 2.1.3						
2.1.1.7. Stochastic differential equations Stochastic differential equations 2.1.1.8.1 Nonparametric representations Nonparametric representations 2.1.1.8.1. Probabilistic inference Stochastic differential equations 2.1.1.8.1. Maximum likelihood Bayesian computation 2.1.2.1 Maximum likelihood Bayesian computation 2.1.2.2. Maximum likelihood Bayesian computation 2.1.2.3. Omputing most probable explanation Bayesian computation 2.1.2.4. Probabilistic reasoning algorithms Density estimation 2.1.2.5. Max marginal computation Density estimation 2.1.3.1. Probabilistic reasoning algorithms Variable elimination 2.1.3.2. Variable elimination Variable elimination 2.1.3.4. Variable elimination Variable elimination 2.1.3.5. Variable elimination Gibbs sampling 2.1.3.5.1. Simulated annealing Matrov-chain Monte 2.1.3.5.1. Simulated annealing Matrov-chain Monte 2.1.3.5.1. Sequential Monte Carlo methods Simulated annealing 2.1.3.5.4. Sequential Monte <td< td=""><td>2.1.1.6.</td><td></td><td>1</td><td></td><td>Causal networks</td><td></td></td<>	2.1.1.6.		1		Causal networks	
2.1.1.8. 2.1.1.8.1. 2.1.1.8.1. 2.1.1.8.2. 2.1.1.8.3. 2.1.1.8.3. 2.1.2. Probabilistic inference problems 1.1.2.1. Maximum likelihood estimation 2.1.2.1. Maximum likelihood estimation 2.1.2.3. Computing most probable science problems 2.1.2.4. Maximum likelihood estimation 2.1.2.5. Computing most probable science problems 2.1.2.6. Max marginal computation Computation Computation Density estimation 2.1.2.6. Max marginal computation Computation Computation Density estimation 2.1.3.1. Variable elimination Loopy belief propagation 2.1.3.2. Variational methods Expectation maximization 2.1.3.5. Gibbs sampling Metropolis-Hastings algorithms 2.1.3.5. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Sequential Monte Carlo methods	2.1.1.7.		1			
2.1.1.8. Probabilis- tic in- ference problems Nonparametric repre- sentations Kernel density esti- mators 2.1.1.8.2. Probabilis- tic in- ference problems Maximum likelihood estimation Kernel density esti- mators 2.1.2.1. Probabilis- tic in- ference problems Maximum likelihood estimation Representation 2.1.2.2. Probabilis- tic rea- soning algorithms Maximum likelihood estimation Quantile regression 2.1.2.4. Probabilis- tic rea- soning algorithms Nax marginal compu- tation Quantile regression 2.1.3.1. Probabilis- tic rea- soning algorithms Variable elimination Loopy belief propaga- tion Quantile regression 2.1.3.4. Probabilis- tic rea- soning algorithms Variable elimination Loopy belief propaga- tion Gibbs sampling Metropolis-Hastings algorithm 2.1.3.5. Sequential Monte Carlo methods Gibbs sampling Metropolis-Hastings algorithm 2.1.3.6. Sequential Monte Carlo methods Carlo methods						
2.1.1.8.1.sentationsKernel density estimators mators Spline models Bayesian compara- metric models2.1.2.1.Probabilis- tic in- ference problemsMaximum likelihood estimation Computing most prob- able explanation Computing most prob- able explanation Density estimationQuantile regression Max marginal compu- tation2.1.2.1.Probabilis- tic rea- soning algorithmsMaximum likelihood estimation Density estimation Density estimation Variable elimination Loopy belief propaga- tion Variable elimination Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo convergence measures2.1.3.6.Sequential Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo methods2.1.3.6.Sequential Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo methods	9110					
2.1.1.8.1. Probabilis- tic in- ference Kernel density esti- mators 2.1.2.1 Probabilis- tic in- ference Maximum likelihood 2.1.2.1 Maximum likelihood 2.1.2.1 Maximum likelihood 2.1.2.1 Maximum likelihood 2.1.2.1 Maximum likelihood 2.1.2.2 Maximum likelihood 2.1.2.3. Maximum likelihood 2.1.2.4 Maximum likelihood 2.1.2.5. Maximum likelihood 2.1.2.6. Max marginal computation Density estimation 2.1.2.6. Max marginal computation Density estimation 2.1.3.1 Probabilis- tic reasoning algorithms Variable elimination Loopy belief propaga- tion Markov-chain Monte Carlo methods 2.1.3.5. Markov-chain Monte Carlo methods Gibbs sampling Markov-chain Monte Carlo convergence meaures 2.1.3.6. Sequential Monte Carlo methods Gibbs convergence meaures	4.1.1.8.					
2.1.1.8.2. 2.1.1.8.3.Probabilis- tic in- ference problemsmators Spline models2.1.2.1.Probabilis- tic in- ference problemsMaximum likelihood estimation Computing most prob- able explanation Computing most prob- able explanation Density estimationQuantile regression quantile regression to rea- soning algorithms2.1.2.5.1.1.2.6.2.1.2.6.Probabilis- tic rea- soning algorithmsQuantile regression to rea- soning algorithms2.1.3.1.Probabilis- tic rea- soning algorithmsQuantile regression to rea- soning algorithms2.1.3.1.Probabilis- tic rea- soning algorithmsVariable elimination Loopy belief propaga- tion Markov-chain Monte Carlo methods2.1.3.5.1Sequential Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo convergence measures2.1.3.6.Sequential Monte Carlo methodsGibbs mapping Markov-chain Monte Carlo convergence measures					sentations	
2.1.1.8.2. 2.1.1.8.3.Probabilis- tic in- ference problemsmators Spline models2.1.2.1.Probabilis- tic in- ference problemsMaximum likelihood estimation Computing most prob- able explanation Computing most prob- able explanation Density estimationQuantile regression quantile regression2.1.2.4.Probabilis- tic rea- soning algorithmsMaximum likelihood estimation Density estimationQuantile regression tic rea- soning algorithms2.1.3.1. 2.1.3.2.Probabilis- tic rea- soning algorithmsVariable elimination Loopy belief propaga- tion Markov-chain Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo methods2.1.3.5.1 2.1.3.5.1Sequential Monte Carlo methodsGibbs sampling Markov-chain Monte Carlo methods2.1.3.6. 2.1.3.7.Sequential Monte Carlo methodsGibbs sampling metric covergence measures	2.1.1.8.1.					Kernel density esti-
2.1.1.8.2. 2.1.1.8.3. 2.1.1.8.3. Probabilistic inference problems Spline models 2.1.2.1. Maximum likelihood estimation Bayesian nonparametric models 2.1.2.2. Maximum likelihood estimation Guantile regression 2.1.2.3. Probabilistic computation Guantile regression 2.1.2.4. Maximum likelihood estimation Quantile regression 2.1.2.5.1 Density estimation Quantile regression 2.1.2.6. Max marginal computation Quantile regression 2.1.3.1. Probabilistic reassoning algorithms Variable elimination Quantile regression 2.1.3.2. Maximum methods Expectation maximization Gibbs sampling Metropolis-Hastings algorithm 2.1.3.5.1. Markov-chain Monte Gibbs sampling Markov-chain Monte Galo convergence measures 2.1.3.6. Sequential Monte Galo convergence measures Markov-chain Mide						
2.1.1.8.3. Bayesian nonparametric models 2.1.2. Probabiliatic inference problems Bayesian comparametric models 2.1.2.1. Maximum likelihood estimation Bayesian comparametric models 2.1.2.2. Despectation Maximum likelihood estimation Bayesian comparametric models 2.1.2.3. Despectation Bayesian computation Bayesian computation 2.1.2.4. Despectation Hypothesis testing and confidence interval computation Density estimation 2.1.2.5. Density estimation Density estimation Density estimation Density estimation 2.1.2.6. Max marginal computation Density estimation Density estimation Density estimation 2.1.3.1. Density estimation Variable elimination Density estimation Density estimation 2.1.3.1. Density estimation Variational methods Expectation maximization Markov-chain Monte 2.1.3.5. Density estimation Gibbs sampling Matropolis-Hastings algorithm 2.1.3.5. Density estimation Sequential Monte Carlo convergence 2.1.3.6. Sequential Monte Carlo convergence Measures 2.1.3.6. <t< td=""><td>21182</td><td></td><td></td><td></td><td></td><td></td></t<>	21182					
2.1.2. Probabilistic inference problems Maximum likelihood estimation Bayesian computation Computing most prob- able explanation Hypothesis testing and computention 2.1.2.3. Maximum likelihood estimation Bayesian computation Density estimation Density estimatio						
2.1.2. Probabilistic inference problems Maximum likelihood estimation 2.1.2.1. Maximum likelihood estimation 2.1.2.2. Maximum likelihood estimation 2.1.2.3. Computing most probable explanation 2.1.2.4. Optimize testing and confidence interval computation 2.1.2.5. Computing most probable explanation 2.1.2.5. Max marginal computation 2.1.2.6. Max marginal computation 2.1.3.6. Probabilistic reassoning algorithms 2.1.3.1. Variable elimination 2.1.3.2. Variable elimination 2.1.3.4. Maximum likelihood 2.1.3.5. Max marginal computation 2.1.3.5.1. Maximum likelihood 2.1.3.5.1. Maximum likelihood 2.1.3.5.1. Maximum likelihood 2.1.3.5.1. Markov-chain Monte 2.1.3.5.1. Markov-chain Monte 2.1.3.5.1. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-	2.1.1.8.3.		1			
2.1.2.1.2.1.2.1.2.1.2.2.2.1.2.3.2.1.2.3.2.1.2.4.2.1.2.5.2.1.2.5.2.1.2.6.2.1.3.1.2.1.3.1.2.1.3.2.2.1.3.3.2.1.3.4.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.3.2.1.3.6.2.1.3.6.2.1.3.7.	1		1			metric models
2.1.2.1.2.1.2.1.2.1.2.2.2.1.2.3.2.1.2.3.2.1.2.4.2.1.2.5.2.1.2.5.2.1.2.6.2.1.3.1.2.1.3.1.2.1.3.2.2.1.3.3.2.1.3.4.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.1.2.1.3.5.3.2.1.3.6.2.1.3.6.2.1.3.7.	2.1.2.		1	Probabilis-		
2.1.2.1.Maximum likelihood estimation2.1.2.2.Maximum likelihood estimation2.1.2.3.Maximum likelihood estimation2.1.2.4.Maximum likelihood estimation2.1.2.5.Computation Computing most prob- able explanation2.1.2.5.Density estimation2.1.2.6.Max marginal compu- tation2.1.3.1.Max marginal compu- tation2.1.3.2.Variable elimination Loopy belief propaga- tion2.1.3.3.Variational methods Expectation maximiza- tion2.1.3.5.Markov-chain Monte Carlo methods2.1.3.5.1.Markov-chain Monte Carlo methods2.1.3.5.1.Sequential Monte Carlo methods2.1.3.6.Sequential Monte Carlo methods2.1.3.7.Sequential Monte Carlo methods	1					
2.1.2.1.problemsMaximum likelihood estimation Bayesian computation Computing most prob- able explanation Hypothesis testing and confidence interval computation Density estimation L.1.2.6.Quantile regression2.1.2.5.Probabilis- tic rea- soning algorithmsMax marginal compu- tationQuantile regression2.1.3.6.Probabilis- tic rea- soning algorithmsVariable elimination Loopy belief propaga- tion Markov-chain Monte Carlo methodsGibbs sampling Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence2.1.3.6.Sequential Monte Carlo methodsGibbs sampling Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence measures						
2.1.2.1. . Maximum likelihood estimation 2.1.2.2. Bayesian computation 2.1.2.3. Computing most probable explanation 2.1.2.4. Computation 2.1.2.5. Computation 2.1.2.6. Density estimation 2.1.2.6. Maximum likelihood 2.1.2.6. Maximum likelihood 2.1.3.1. Probabilistic rearsoning algorithms 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.5. Maximum likelihood 2.1.3.5.1. Maximum likelihood 2.1.3.5.1. Maximum likelihood 2.1.3.5.3. Sequential Monte Carlo methods Simulated annealing Markov-chain Monte Carlo convergence Carlo methods Simulated annealing	1		1			
2.1.2.2. 2.1.2.3.estimation Bayesian computation Computing most prob- able explanation Density estimation Density estimation Density estimation Density estimation 2.1.2.6.Quantile regression Quantile regression2.1.2.6. 2.1.3.6.Probabilis- tic reas- soning algorithmsMax marginal compu- tationQuantile regression2.1.3.1. 2.1.3.2.Variable elimination Loopy belief propaga- tion Variational methods Expectation maximiza- tionGibbs sampling Metropolis-Hastings algorithm2.1.3.5.1. 2.1.3.5.1. 2.1.3.5.4.Sequential Monte Carlo methodsGibbs sampling Metropolis-Hastings algorithm2.1.3.6. 2.1.3.7.Sequential Monte Carlo methodsGibbs sampling Metropolis-Hastings algorithm	1			problems		
2.1.2.2. 2.1.2.3.estimation Bayesian computation Computing most prob- able explanation Density estimation Density estimation Density estimation Density estimation 2.1.2.6.Quantile regression Quantile regression2.1.2.6. 2.1.3.6.Probabilis- tic reas- soning algorithmsMax marginal compu- tationQuantile regression2.1.3.1. 2.1.3.2.Variable elimination Loopy belief propaga- tion Variational methods Expectation maximiza- tionGibbs sampling Metropolis-Hastings algorithm2.1.3.5.1. 2.1.3.5.1. 2.1.3.5.4.Sequential Monte Carlo methodsGibbs sampling Metropolis-Hastings algorithm2.1.3.6. 2.1.3.7.Sequential Monte Carlo methodsGibbs sampling Metropolis-Hastings algorithm	2.1.2.1.				Maximum likelihood	
2.1.2.2. Bayesian computation 2.1.2.3. Bayesian computation 2.1.2.4. Computing most probable explanation 2.1.2.4. Hypothesis testing and comfidence interval computation 2.1.2.5. Density estimation 2.1.2.6. Max marginal computation 2.1.3.1. Probabilistic reasoning algorithms 2.1.3.2. Variable elimination 2.1.3.3. Variable elimination 2.1.3.4. Variational methods 2.1.3.5. Markov-chain 2.1.3.5.1. Markov-chain 2.1.3.5.1. Markov-chain 2.1.3.5.1. Sequential 2.1.3.6. Sequential 2.1.3.7. Sequential	1		1			
2.1.2.3. Computing nost probable explanation 2.1.2.4. Hypothesis testing and confidence interval computation 2.1.2.5. Computing nost probable explanation 2.1.2.5.1 Density estimation 2.1.2.6. Max marginal computation 2.1.3.1. Probabilistic reasoning algorithms 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.4. Variational methods 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Gibbs sampling 2.1.3.5.1. Gibbs sampling 2.1.3.5.1. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-	2122					
2.1.2.4. able explanation - Hypothesis testing and confidence interval computation Density estimation Quantile regression 2.1.2.5.						
2.1.2.4. Hypothesis testing and confidence interval computation 2.1.2.5.1 Density estimation 2.1.2.6. Max marginal computation 2.1.3.6. Probabilistic reassoning algorithms 2.1.3.1. Variable elimination 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.5. Variational methods 2.1.3.5.1 Max Narpinal methods 2.1.3.5.1 Max Nov-chain Monte 2.1.3.5.1 Max Nov-chain Monte 2.1.3.5.3 Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-	2.1.2.3.		1			
2.1.2.4. Hypothesis testing and confidence interval computation 2.1.2.5.1 Density estimation 2.1.2.6. Max marginal computation 2.1.3.6. Probabilistic reassoning algorithms 2.1.3.1. Variable elimination 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.5. Variational methods 2.1.3.5.1 Max Narpinal methods 2.1.3.5.1 Max Nov-chain Monte 2.1.3.5.1 Max Nov-chain Monte 2.1.3.5.3 Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-	1				able explanation	
2.1.2.5. 2.1.2.5.1. Confidence interval computation Density estimation Quantile regression 2.1.2.6. Max marginal computation Quantile regression 2.1.3.6. Probabilistic reasing algorithms Variable elimination Quantile regression 2.1.3.1. Variable elimination Variable elimination Gibbs sampling 2.1.3.2. Variable elimination Variational methods Gibbs sampling 2.1.3.3. Variational methods Expectation maximization Gibbs sampling 2.1.3.5.1 Markov-chain Monte Carlo methods Gibbs sampling 2.1.3.5.1 Sequential Monte Carlo convergence Markov-chain Monte 2.1.3.6. Sequential Monte Carlo methods Markov-chain Monte 2.1.3.7. Kalman filters and hid- Monte Carlo convergence	2.1.2.4.		1			
2.1.2.5. 2.1.2.5.1. Computation Quantile regression 2.1.2.6. Max marginal computation Max marginal computation Quantile regression 2.1.3.6. Probabilistic reasoning algorithms Variable elimination Quantile regression 2.1.3.1. Variable elimination Loopy belief propagation Variational methods Probabilistion 2.1.3.3. Variational methods Expectation maximization Variational methods Variational methods 2.1.3.5. Variational methods Expectation maximization Markov-chain Monte Gibbs sampling 2.1.3.5.1. Variational methods Simulated annealing Markov-chain Monte Galorothm 2.1.3.5.3. Variational Monte Carlo methods Gibbs sampling Markov-chain Monte 2.1.3.6. Sequential Monte Carlo methods Markov-chain Monte Carlo convergence 2.1.3.7. Kalman filters and hid- Monte Variational Monte Variational Monte Carlo Kalman filters and hid- Markov-chain Monte Variational Monte Variational Monte Variational Monte Variational Monte Variational Monte Variational Monte Variational	1					
2.1.2.5. 2.1.2.5. 2.1.2.5.1. 2.1.2.6. 2.1.2.6. Probabilistic reasoning algorithms 2.1.3.1. Variable elimination Loopy belief propagation 2.1.3.3. Variational methods 2.1.3.4. Variational methods 2.1.3.5. Gibbs sampling Metropolis-Hastings algorithms 2.1.3.5.1. Markov-chain Monte Carlo methods 2.1.3.5.2. Sequential Monte Carlo methods 2.1.3.5.4. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-						
2.1.2.5.1. Quantile regression 2.1.2.6. Max marginal computation 2.1.3.1. Probabilistic reasoning algorithms 2.1.3.2. Variable elimination 2.1.3.3. Variable elimination 2.1.3.4. Variational methods 2.1.3.5. Max marginal computation 2.1.3.5.1. Max marginal computation 2.1.3.5.3. Max marginal computation 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	1					
2.1.2.6. Max marginal computation 2.1.3.1. Probabilistic reasoning algorithms 2.1.3.1. Variable elimination 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.4. Variational methods 2.1.3.5. Max marginal computation 2.1.3.5.1. Max marginal computation 2.1.3.5.2. Max marginal computation 2.1.3.5.3. Max marginal computation 2.1.3.5.4. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-					Density estimation	
2.1.2.6. Max marginal computation 2.1.3.1. Probabilistic reasoning algorithms 2.1.3.1. Variable elimination 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.4. Variational methods 2.1.3.5. Max marginal computation 2.1.3.5.1 Max marginal computation 2.1.3.5.2 Max marginal computation 2.1.3.5.3. Max marginal computation 2.1.3.5.3. Sequential Monte Carlo methods Carlo methods 2.1.3.6. Sequential Monte 2.1.3.7. Kauman filters and hid-	2.1.2.5.1.		1			Quantile regression
2.1.3. Probabilis- tic rea- soning algorithms tation 2.1.3.1. tation 2.1.3.2. Variable elimination Loopy belief propaga- tion 2.1.3.3. Variable elimination Loopy belief propaga- tion 2.1.3.4. Expectation maximiza- tion Markov-chain Monte Carlo methods 2.1.3.5.1. Markov-chain Monte Carlo methods 2.1.3.5.3. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte 		1	1	1	Max marginal compu-	
2.1.3. Probabilistic reasoning algorithms 2.1.3.1. Variable elimination 2.1.3.2. Variable elimination 2.1.3.3. Variational methods 2.1.3.4. Variational methods 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Markov-chain Monte 2.1.3.5.2. Markov-chain Monte 2.1.3.5.3. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-						
2.1.3.1. tic reasoning algorithms 2.1.3.2. Variable elimination Loopy belief propaga- tion 2.1.3.3. Variable elimination Loopy belief propaga- tion 2.1.3.4. Expectation maximiza- tion 2.1.3.5. Markov-chain Monte Carlo methods 2.1.3.5.2. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Sequential Monte Carlo methods	0.1.0			D 1 1	Lation	
2.1.3.1. soning algorithms Variable elimination Loopy belief propaga- tion 2.1.3.3. Variable elimination Loopy belief propaga- tion 2.1.3.3. Variational methods Expectation maximiza- tion 2.1.3.5. Markov-chain Monte Carlo methods 2.1.3.5.1. Gibbs sampling Metropolis-Hastings algorithm 2.1.3.5.2. Sequential Monte Carlo methods 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.					
2.1.3.1. algorithms 2.1.3.1. Variable elimination 2.1.3.2. Loopy belief propaga- tion 2.1.3.3. Variational methods 2.1.3.4. Expectation maximiza- tion 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Simulated annealing 2.1.3.5.4. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-	1			tic rea-		
2.1.3.1. algorithms 2.1.3.1. Variable elimination 2.1.3.2. Loopy belief propaga- tion 2.1.3.3. Variational methods 2.1.3.4. Expectation maximiza- tion 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Simulated annealing 2.1.3.5.4. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-						
2.1.3.1. Variable elimination 2.1.3.2. Loopy belief propagation 2.1.3.3. Variational methods 2.1.3.3. Variational methods 2.1.3.4. Expectation maximization 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Sequential Monte 2.1.3.5.4. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-						
2.1.3.2. Loopy belief propagation 2.1.3.3. Variational methods 2.1.3.4. Expectation maximization 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Sequential Monte 2.1.3.5.3. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-				soning		
2.1.3.3. tion 1.0 2.1.3.4. Variational methods 2.1.3.5. Expectation maximization 2.1.3.5.1. Markov-chain Monte 2.1.3.5.2. Carlo methods 2.1.3.5.3. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-	0121			soning	Variable elimination	
2.1.3.3. Variational methods 2.1.3.4. Expectation maximization 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Simulated annealing 2.1.3.5.4. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-				soning		
2.1.3.4. Expectation maximization 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Metropolis-Hastings algorithm 2.1.3.5.3. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-				soning	Loopy belief propaga-	
2.1.3.4. Expectation maximization 2.1.3.5. Markov-chain Monte 2.1.3.5.1. Carlo methods 2.1.3.5.2. Metropolis-Hastings algorithm 2.1.3.5.3. Sequential Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid-				soning	Loopy belief propaga-	
2.1.3.5. tion Markov-chain Monte Carlo methods 2.1.3.5.1. Gibbs sampling Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence measures 2.1.3.6. Sequential Markov-chain Monte Carlo convergence measures 2.1.3.7. Sequential Kalman filters and hid-	2.1.3.2.			soning	Loopy belief propaga- tion	
2.1.3.5. Markov-chain Monte Carlo methods Gibbs sampling Metropolis-Hastings algorithm 2.1.3.5.2. Simulated annealing Markov-chain Monte Carlo convergence measures 2.1.3.6. Sequential Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3.			soning	Loopy belief propaga- tion Variational methods	
2.1.3.5.1. Carlo methods Gibbs sampling Metropolis-Hastings algorithm 2.1.3.5.3. Simulated annealing Markov-chain Monte Carlo convergence measures 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza-	
2.1.3.5.1. Gibbs sampling 2.1.3.5.2. Metropolis-Hastings 2.1.3.5.3. Jagorithm 2.1.3.5.4. Markov-chain Monte Carlo convergence 2.1.3.6. Sequential 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion	
2.1.3.5.1. Gibbs sampling 2.1.3.5.2. Metropolis-Hastings 2.1.3.5.3. Jagorithm 2.1.3.5.4. Markov-chain Monte Carlo convergence 2.1.3.6. Sequential 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	
2.1.3.5.2. Metropolis-Hastings algorithm 2.1.3.5.3. Simulated annealing Markov-chain Monte Carlo convergence measures 2.1.3.6. Sequential Carlo methods Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	
2.1.3.5.3. 2.1.3.5.4. 2.1.3.6. 2.1.3.7.	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Gibbs sampling
2.1.3.5.3. Simulated annealing 2.1.3.5.4. Markov-chain Monte 2.1.3.6. Sequential Monte 2.1.3.7. Kalman filters and hid	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	
2.1.3.5.4. Markov-chain Monte Carlo convergence measures 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Metropolis-Hastings
2.1.3.5.4. Markov-chain Monte Carlo convergence measures 2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Metropolis-Hastings algorithm
2.1.3.6. Sequential Monte Carlo methods Xiaman filters and hid	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Metropolis-Hastings algorithm
2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Metropolis-Hastings algorithm Simulated annealing
2.1.3.6. Sequential Monte Carlo methods 2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte
2.1.3.7. Carlo methods Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence
2.1.3.7. Carlo methods Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3. 2.1.3.5.4.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte Carlo methods	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence
2.1.3.7. Kalman filters and hid-	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3. 2.1.3.5.4.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte Carlo methods Sequential Monte	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence
	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3. 2.1.3.5.4.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte Carlo methods Sequential Monte	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence
LUCH MARKOV MODELS	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3. 2.1.3.5.4. 2.1.3.6.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte Carlo methods Sequential Monte Carlo methods	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence
	2.1.3.2. 2.1.3.3. 2.1.3.4. 2.1.3.5. 2.1.3.5.1. 2.1.3.5.2. 2.1.3.5.3. 2.1.3.5.4. 2.1.3.6.			soning	Loopy belief propaga- tion Variational methods Expectation maximiza- tion Markov-chain Monte Carlo methods Sequential Monte Carlo methods	Metropolis-Hastings algorithm Simulated annealing Markov-chain Monte Carlo convergence

2.1.3.7.1.* Probabilis- Resampling methods Factorial HMM 2.1.3.8.1 Resampling methods Bootstrapping 2.1.3.9. Random number generation Jackknifing	
2.1.3.8.1. Bootstrapping 2.1.3.8.2. Jackknifing 2.1.3.9. Random number generation	
2.1.3.8.2. 2.1.3.9. Random number gener- ation Jackknifing	
2.1.3.9. Random number gener- ation	
2.1.3.9. Random number gener- ation	
ation	
2.1.4. Frobabilis-	
tic algo-	
rithms	
2.1.5. Statistical	
paradigms	
2.1.5.1. Queueing theory	
2.1.5.2. Contingency table	
analysis	
2.1.5.3. Regression analysis	
2.1.5.3.1. Robust regression	
2.1.5.4. Time series analysis	
2.1.5.5. Survival analysis	
2.1.5.6. Renewal theory	
tion	
2.1.5.8. Cluster analysis	
2.1.5.9. Statistical graphics	
2.1.5.10. Exploratory data anal-	
ysis	
2.1.6. Stochastic	
processes	
2.1.6.1. Markov processes	
2.1.7. Nonpara-	
metric	
statistics	
2.1.8. Distribu-	
tion func-	
tions	
2.1.9. Multiva-	
riate	
statistics	
3. Informa-	
tion	
systems	
3.1. Data man-	
agement	
systems	
3.1.1. Database	
design and	
models	
3.1.1.1. Relational database	
model	
3.1.1.2. Entity relationship	
models	
3.1.1.3. Graph-based database	
models	
3.1.1.3.1. Hierarchical data	
models	
3.1.1.3.2. Network data models	
3.1.1.4. Physical data models	
3.1.1.5. Data model extensions	
3.1.1.5.1. Semi-structured	
data	
3.1.1.5.2. Data streams	
3.1.1.5.3. Data provenance	
3.1.1.5.4. Incomplete data	
3.1.1.5.5. Temporal data	
3.1.1.5.6. Uncertainty	
3.1.1.5.7. Inconsistent data	
3.1.2. Data	
structures	
3.1.2.1. Data access methods	
3.1.2.1.1. Data access methods Multidimensional	
range search	
3.1.2.1.2. Data scans	
3.1.2.1.3. Point lookups	
3.1.2.1.3. Point lookups 3.1.2.1.4. Unidimensional	
3.1.2.1.3. Point lookups	

3.1.2.2. 3.1.2.2.3. Data layout Data compression Data compression Record and block layout 3.1.2.2.3. Database agement orgines DBMS engine architec- tures Database query pro- cessing Query optimization Query languages query languages qu		 				
3.1.2.2.2. 3.1.2.2.3. Data bace many processing Data bace many processing Record and block layout 3.1.3.1. DBMS engine architector bases query processing Query optimization Query planning Join algorithms 3.1.3.2. Database transaction logging Database covery and distributed database transaction logging Database recovery and distributed DBMSs Database transaction logging Database covery and distributed database transaction logging Database recovery and distributed database transaction logging Database recovery and distributed database transaction and distributed database transaction and transaction logging Database recovery and distributed database transaction and transaction logging Database recovery and distributed database transaction and transaction and transaction logging Database recovery and distributed database transaction and transa	3.1.2.2.			Data layout		
3.1.2.2.3. Image: I					Data compression	
3.1.2.2.3. Image: I	3.1.2.2.2.				Data encryption	
3.1.3. Database man- system engines Database man- system engines DBMS engine archites- tures Query optimization Query optimization Natal obtime Parallel and ditributed DBMSs 31.3.5.1 31.3.5.1 31.3.5.1 Triggers and rules Database views Integrity theoking 31.3.5.2 31.3.5.1 Triggers and rules Database views Integrity theoking Database views Integrity theoking Database Views Integrity theoking Database Views Integrity theoking Database Views Integrity theoking Database Views Nate Nate Nate Nate Nate Nate Nate Nate	3.1.2.2.3.					
3.1.3. Database agement system engines DBMS engine architec- tures Database query pro- cessing Guery optimization (Guery optimization (Guery optimizet in logging Join algorithms 3.1.3.1. Database query pro- cessing Database transaction processing Guery optimization (Guery optimizet in logging Join algorithms 3.1.3.2. J.3.3.3. Database transaction processing Framework processing 3.1.3.3. J.3.3.3. Triggers and rules tributed DBMSs Framework processing 3.1.3.5. Triggers and rules processing engines Triggers and rules obtainabase views Integrity checking Distributed database processing engines 3.1.3.6. Triggers and rules and distributed Distributed database processing engines 3.1.3.9. Triggers and rules processing engines Distributed database processing engines 3.1.3.9. Main memory engines optimise Distributed database processing engines 3.1.3.0. Triggers and rules processing engines Structured Query Language 3.1.3.10. Triggers and rules processing engines Structured Query Language 3.1.4.1. Distributed database processing engines Distributed database processing engines 3.1.3.1.4.1. Deduplication Extraction transforma- tion integration all data in- tegration all data in- tegration Main database <td>0.1.2.2.0.</td> <td></td> <td></td> <td></td> <td></td> <td></td>	0.1.2.2.0.					
3.1.3.1. anan- system ongines DBMS engine architec- tures Query optimization Query optimization Query optimization Query optimization Query paining Join algorithms 3.1.3.2. 3.1.3.2. Database query pro- cessing Query optimization Query optimization Database recovery 3.1.3.5. Triggers and rules Database views Integrity checking Distributed database transactions Distributed database transactions 3.1.3.0. Triggers and rules Database views Integrity checking Distributed database transactions Distributed database transactions 3.1.3.0. Main memory engines Stream management Strictured Query Imaguages 3.1.3.1. Main memory engines Stream management Yeath XQuery 3.1.4.1.1 Main memory engines Stream management Main memory engines Pata checking Deduptication Extraction transforma- tion integrity resolution 3.1.4.1.1. Call level interfaces XPath XQuery 3.1.4.1.2. Database Paduptication Extraction transforma- tion integrity resolution 3.1.5.6. Deduptication Extraction transforma- tion integrity resolution	0.1.0		D (1		layout	
3.1.3.1. agement engines DBMS engine architec- tures Query optimization Query optimization Patalocking Transaction logging Database recovery 3.1.3.5. Triggers and rules Distributed database transactions Distributed database precovery 3.1.3.6. Triggers and rules Distributed database query languages Distributed database precovery 3.1.3.0. Main memory engines Online analytical pro- tegine engines Stream management Distributed database precovery 3.1.3.1. Main memory engines Online analytical pro- tegine engines Stream management Structured Query Languages 3.1.4.1.1. Main interpreter patholicking Database recovery Main Memory engines Online analytical pro- tegine engines Stream management 3.1.4.1.1. Main memory engines Online analytical pro- tegine engines Stream management Main Memory engines Online analytical pro- tegine engines Stream management 3.1.4.1.1. Deduplication Extraction transforma-	3.1.3.					
3.1.3.1. Signame engines DBMS engine architectures 3.1.3.2. Database query pro- cessing Query optimization Query optimization Patalocking 3.1.3.5.1 Tisgers and rules Database views Integrity checking Distributed database virus and distributed Database views Integrity checking Distributed database recovery 3.1.3.5.1 Main memory engines Online anglytical pro- cessing engines Stream management and distributed database recovery 3.1.3.1.1 Main memory engines Online anglytical pro- cessing engines Stream management and distributed database recovery 3.1.3.1.1 Main memory engines Online anglytical pro- cessing engines Stream management and distributed database recovery 3.1.4.1 Main memory engines Online anglytical pro- cessing engines Streat management and distributed database recovery 3.1.4.1 Informa- tigration Call level interfaces 3.1.4.3 Deduplication Extraction transforma- tion and loading Data exchange Data exchange Data cenaning 3.1.5.6 Deduplication Extraction transforma- tion and loading Wappers (data min- tigration						
3.1.3.1enginesDBMS engine archite-tures Database query pro- cessingDBMS engine archite- tures Database query pro- cessingQuery optimization Query and distributed DBMSs3.1.3.6. 3.1.3.7. 3.1.3.9.Triggers and rules Distributed database tranaactionsDistributed database toking Distributed database tercovery3.1.3.0. 3.1.3.1.Query languages nonine gaugesStructured Query Language And distributed Distributed database tercovery3.1.4.1.Patalos Distributed interfacesStructured Query Language Maine many tengines Structured application Extraction transforma- ton and loading Database3.1.4.1.Informa- grationCall level interfacesStructured Query Language3.1.5.1. 3.1.5.1.<			agement			
3.1.3.1enginesDBMS engine archite-tures Database query pro- cessingDBMS engine archite- tures Database query pro- cessingQuery optimization Query and distributed DBMSs3.1.3.6. 3.1.3.7. 3.1.3.9.Triggers and rules Distributed database tranaactionsDistributed database toking Distributed database tercovery3.1.3.0. 3.1.3.1.Query languages nonine gaugesStructured Query Language And distributed Distributed database tercovery3.1.4.1.Patalos Distributed interfacesStructured Query Language Maine many tengines Structured application Extraction transforma- ton and loading Database3.1.4.1.Informa- grationCall level interfacesStructured Query Language3.1.5.1. 3.1.5.1.<			system			
3.1.3.1. DBMS engine architectures 3.1.3.2. Database query processing 3.1.3.2.1. Query optimization Query framangement 3.1.3.3.1. Triggers and rules Database views Integrity checking Distributed database views Integrity checking Distributed database recovery 3.1.3.5.1. Main memory engines Distributed database views Integrity nangueges 3.1.3.5.1. Main memory engines Distributed database views Integrity nangueges 3.1.3.5.1. Main memory engines Distributed database views Integrity nangueges 3.1.3.1. Main memory engines Datalocks 3.1.3.1. Query languages for none latonal engines Datalocks 3.1.4.1. Information the gration Call level interfaces Structured Query Languages for none latonal engines 3.1.4.1. Deduplication Extraction transformation and loading Data exchange Data ecchange Data ecchange Data ecchange Data ecchange Data ecch						
1.1.3.2 tures ures	0.1.0.1		engines	DDMG · 1.4		
3.1.3.2.1. 3.1.3.2.1. 3.1.3.2.1. Query optimization Query planning January (Parallel and distributed DBMSs in the tributed database recovery and distributed database recovery in the tributed databa	3.1.3.1.					
31.3.2.1. acressing Query optimisation Query optimisation Processing 1.1.3.1. Patabase transaction Processing Data locking Database recovery 3.1.3.5.1. Triggers and rules Database views Integrity checking Distributed database transactions 3.1.3.6. Triggers and rules Database views Integrity checking Distributed database transactions 3.1.3.9.1. Distributed database online analytical pro- covery 3.1.3.1. Query lan- guages 3.1.3.1. Query languages Online analytical pro- covery 3.1.3.1. Query languages Online analytical pro- covery 3.1.4.1. Main memory engines Online analytical pro- covery 3.1.4.1. Proma- guages 3.1.5.1. Proma- guages						
3.1.3.2.1 Query operators Query operators 3.1.3.2.3 Query operators Query poperators 3.1.3.2.3 Query poperators Query poperators 3.1.3.3.1 Database transaction Database recovery 3.1.3.3.1 Database recovery Transaction logging 3.1.3.4 Parallel and distributed DBMSs Key-value stores 3.1.3.5.1 Triggers and rules MapReduce-base 3.1.3.5.3 Triggers and rules Database 3.1.3.5.3 Triggers and rules Database 3.1.3.6. Triggers and rules Database 3.1.3.6. Triggers and rules Distributed database 3.1.3.9.1 Triggers and rules Distributed database 3.1.3.9.3 Record and stabase Distributed database 3.1.3.9.1 Record and parallel and Main memory engines 3.1.3.9.2 Structured Query Parallel and 3.1.3.1.3 Query lan- Relational database 3.1.3.1.3 Relational database Structured Query 3.1.4.1.1 Query languages MapReduce lan- 3.1.4.1.1 Call level interfaces	3.1.3.2.			Database query pro-		
3.1.3.2.2. 3.1.3.2.3. Query operators Query planning 3.1.3.2.3. Join algorithms Join algorithms 3.1.3.3.1. Join algorithms Join algorithms 3.1.3.3.1. Database transaction Data locking 3.1.3.4. Transaction legging Data locking 3.1.3.5. Record and buffer Mainsequent 3.1.3.5.1. Triggers and rules MapReduce-based 3.1.3.5.3. Triggers and rules DBMSs 3.1.3.6. Triggers and rules DBMSs 3.1.3.6. Triggers and rules DBMSs 3.1.3.7. Distributed database Integrity checking 3.1.3.8. Integrity checking Distributed database 3.1.3.9.1 Distributed database Integrity checking 3.1.3.9.1 Main memory engines Structured Query 3.1.3.10. Structured Query XPath 3.1.4.1. Query languages Structured Query 3.1.4.1. Anapreduce MapReduce languages 3.1.4.2. Anaptage XPath 3.1.4.2. Call level interfaces MapReduce languages				cessing		
3.1.3.2.2. 3.1.3.2.3. Query operators Query planning 3.1.3.2.3. Join algorithms Join algorithms 3.1.3.3.1. Join algorithms Join algorithms 3.1.3.3.1. Database transaction Data locking 3.1.3.4. Transaction legging Data locking 3.1.3.5. Record and buffer Mainsequent 3.1.3.5.1. Triggers and rules MapReduce-based 3.1.3.5.3. Triggers and rules DBMSs 3.1.3.6. Triggers and rules DBMSs 3.1.3.6. Triggers and rules DBMSs 3.1.3.7. Distributed database Integrity checking 3.1.3.8. Integrity checking Distributed database 3.1.3.9.1 Distributed database Integrity checking 3.1.3.9.1 Main memory engines Structured Query 3.1.3.10. Structured Query XPath 3.1.4.1. Query languages Structured Query 3.1.4.1. Anapreduce MapReduce languages 3.1.4.2. Anaptage XPath 3.1.4.2. Call level interfaces MapReduce languages	3.1.3.2.1.			-	Query optimization	
3.1.3.2.3. 3.1.3.2.3. Query landing Join algorithms 3.1.3.3.1. Join algorithms Join algorithms 3.1.3.3.1. Parabase transaction Parasaction logging 3.1.3.3.3. Parallel mand dis Parallel mand dis 3.1.3.5.1 Parallel mand dis Parallel mand dis 3.1.3.6 Parallel mand dis Parallel mand dis 3.1.3.6 Parallel mand dis Parallel mand dis 3.1.3.7 Parallel mand dis Parallel mand dis 3.1.3.6 Parallel mand dis Parallel mand dis 3.1.3.7 Parallel mand dis Parallel mand dis 3.1.3.8 Parallel mand dis Parallel mand dis 3.1.3.9 Parallel mand dis Parallel mand dis 3.1.3.9 Parallel mand dis Parallel mand dis 3.1.3.1 Parallel mand dis Parallel mand dis 3.1.3.1 Parallel mand dis Parallel mand dis 3.1.3.1 Paralle						
3.1.3.2.3. Join algorithms 3.1.3.3. Join algorithms 3.1.3.3. Database transaction processing Database recovery 3.1.3.4. Record and buffer Transaction logging 3.1.3.5. Tributed DBMSs Key-value stores 3.1.3.6. Triggers and rules MarReduce-based 3.1.3.6. Triggers and rules Database recovery 3.1.3.7. Triggers and rules Database 3.1.3.8. Triggers and rules Database 3.1.3.8. Triggers and rules Database 3.1.3.8. Triggers and rules Database 3.1.3.9. Triggers and rules Database 3.1.3.9. Triggers and rules Database 3.1.3.9. Trinsactions DBMSs 3.1.3.9. Main memory engines Distributed database 3.1.3.9. Relational database Truerry languages 3.1.3.10. Main memory engines Stream management 3.1.4.1. Main interprotecking Stream management 3.1.4.1.1. Main interprotecking Main memory engines 3.1.4.2. Mapreduce language <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
3.1.3.3. 3.1.3.3.1 3.1.3.3.1. 3.1.3.3.2 3.1.3.3.2. 3.1.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3						
3.1.3.3.1. and buffer Data locking 3.1.3.3.2. and buffer Transaction logging 3.1.3.3.3. anagement Batabase recovery 3.1.3.5.1. and distributed 3.1.3.5.2. and management 3.1.3.5.3. and distributed 3.1.3.5.3. Triggers and rules 3.1.3.9.1 Triggers and rules 3.1.3.9.1 Triggers and rules 3.1.3.9.1 Database views 3.1.3.9.2 Trinsactions 3.1.3.9.3 Trinsactions 3.1.3.9.1 Main memory engines 3.1.3.9.2 Trinsactions 3.1.3.10. Stream management 3.1.3.10. Relational database 3.1.4.1.1 Relational database guarges Tructured Query 3.1.4.1.1 Main memory engines 3.1.4.1.1 Triggers for 3.1.4.1.1 Call level interfaces 3.1.4.1.1 Deduplication 3.1.4.2.1 Triggers (data min- 3.1.4.3.1 Data echaning 3.1.5.1. Data echaning <td></td> <td></td> <td></td> <td></td> <td>Join algorithms</td> <td></td>					Join algorithms	
3.1.3.3.1. 3.1.3.3.2, 3.1.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3	3.1.3.3.					
3.1.3.3.2. 3.1.3.3.2. 3.1.3.3.4. Record and buffer management Parallel and distributed DBMSs Transaction Jogging Database recovery 3.1.3.5.1. Status of the second and buffer management Parallel and distributed DBMSs Key-value stores wasterns Relational parallel and distributed Database views 3.1.3.6. Status views Triggers and rules Database views Distributed database views 3.1.3.9.1 Distributed database views Distributed database views Distributed database views 3.1.3.9.1 Straum anagement stransactions Distributed database views Distributed database views 3.1.3.9.2 Straum anagement stransactions Distributed database views Distributed database views 3.1.3.10. Straum anagement stransactions Distributed database views Distributed database views 3.1.4.1. Query languages Straum anagement stransactions Distributed database views 3.1.4.1.1. Relational database views Call level interfaces KPath XQuery 3.1.4.2. Call level interfaces Structured Query MapReduce languages 3.1.4.3.1. Deduplication Extraction transformation and loading Data exchange				processing		
3.1.3.3.2. 3.1.3.3.2. 3.1.3.3.4. Record and buffer management Parallel and distributed DBMSs Transaction logging Database recovery 3.1.3.5.1. Record and buffer management Parallel and distributed DBMSs Key-value stores systems Relational parallel and distributed Database rises 3.1.3.6. S1.3.5.1. Triggers and rules Database views Integrity checking Distributed database rises Distributed database recovery 3.1.3.9.1. Distributed database views Integrity checking Distributed database rises and rules Conline analytical processing engines Distributed database recovery 3.1.3.9.1. Stream management Distributed database recovery 3.1.3.1. Relational database recovery Distributed database recovery 3.1.3.1. Relational database recovery Distributed database recovery 3.1.3.1. Record and public recovery Stream management 3.1.3.1. Recovery languages Structured Query Language 3.1.4.1. Relational database recovery Structured Query Language 3.1.4.1.1. Call level interfaces Structured Query Languages for non-relational engines Data exchange Data e	3.1.3.3.1.				Data locking	
3.1.3.3.3. 3.1.3.4. 3.1.3.4. Second and buffer management Parallel and disintributed DBMSs Key-value stores MapReduce-based systems Relational parallel and distributed DBMSs 3.1.3.5.1. Second and buffer Parallel and distributed DBMSs Triggers and rules Database views Integrity to characterize the systems Relational parallel and distributed database transactions Key-value stores MapReduce-based systems Relational parallel and distributed database transactions 3.1.3.6. Second and buffer Parallel and distributed database transactions Distributed database transactions 3.1.3.9.1 Integrity checking Distributed database transactions Distributed database transactions 3.1.3.9.3. Second and buffer Parallel and distributed database transactions Distributed database transactions 3.1.3.10. Main memory engines Stream management Structured Query Languages 3.1.4.1. Relational database for non-relational engines Structured Query Languages for non-relational engines 3.1.4.2. Structured Call level interfaces XPath XQuery MapReduce Languages 3.1.4.3.1. Deduplication Extraction transformation and loading Data exchange D						
3.1.3.4. Record and buffer management Parallel and distributed DBMSs Key-value stores systems Relational parallel and distributed DBMSs 3.1.3.5.1 S.1.3.5.2 Triggers and rules Database views Integrity checking Distributed database transactions Relational parallel and distributed DBMSs 3.1.3.6 S.1.3.7. Triggers and rules Database views Integrity checking Distributed database transactions Distributed database transactions 3.1.3.9.1 Main memory engines Distributed database transactions Distributed database transactions 3.1.3.10. Straut management Relational database transactional engines Structured Query Languages 3.1.4.1. Straut management Kul query languages for non-relational engines Muery Anguages for non-relational engines 3.1.4.2. Call level interfaces Deduplication Extractions MapReduce languages 3.1.4.3.1. Deduplication Extractional engines Triggers (data mining) 3.1.4.4. Distributed chatabase for non-relational engines MapReduce languages 3.1.5.1. Call level interfaces Muery hanguages for non-relational engines 3.1.5.1. Deduplication Extraction transformation and loading pata exchange Data exch						
3.1.3.5.1 3.1.3.5.2.Image ment Parallel and dis- tributed DBMSsKey-value stores mayReduce-based systems Relational parallel and distributed DBMSs3.1.3.5.3.Image ment Parallel and dis- tributed DBMSsKey-value stores mayReduce-based systems Relational parallel and distributed DBMSs3.1.3.6. 3.1.3.9.1Triggers and rules Database views Integrity thecking Distributed database transactionsKey-value stores mayReduce-based systems Relational parallel and distributed database transactions3.1.3.9.1. 3.1.3.9.2. 3.1.3.9.3.Main memory engines Online analytical pro- cessing engines Stream management stream management stream management3.1.4.1. 3.1.4.1.1Main memory engines Online analytical pro- cessing engines3.1.4.1.2. 3.1.4.2. 3.1.4.3.1.Imforma- tion integration gration integration tion and loading Data cenange Marpeduce languages3.1.4.3.1. 3.1.5.1. 3.1.5.4. 3.1.5.4. 3.1.5.4. 3.1.5.5.Imforma- tion and loading Data cenange Marpeduce languages3.1.5.6. a.1.5.7.Imforma- tion and loading Data cenange Wrappers (data min- tion and loading Data cenange Wrappers (data min- tegration Entity resolution3.1.5.6. 3.1.5.7.Import ment tegrationDefunction ment tegration Entity resolution3.1.5.7.Import ment tegration Entity resolution				Descel a 1 1 m	Database recovery	
3.1.3.5.1 3.1.3.5.1 3.1.3.5.2 and distributed DBMSs 3.1.3.5.3.2 and distributed DBMSs 3.1.3.5.3.2 and distributed DBMSs 3.1.3.5.3.2 and distributed DBMSs 3.1.3.6 and distributed Database views 3.1.3.6 and distributed Database views 3.1.3.7 Database views 3.1.3.8 Database views 3.1.3.9.1 Database views 3.1.3.9.1 Database views 3.1.3.9.1 Database views 3.1.3.9.1 Distributed database 3.1.3.9.3 Frequencies 3.1.3.9.3 Distributed database 3.1.3.10 Relational database 3.1.4.1 Relational database 3.1.4.1 Relational database 3.1.4.1.1 Relational database 3.1.4.1.2 XML query languages 3.1.4.2 XML query languages 3.1.4.2 Information integration 3.1.4.3.1 Deduplication 3.1.4.3.1 Deduplication 3.1.4.3.1 Deduplication 3.1.4.4.3 Deduplication 3.1	3.1.3.4.					
1.3.5.1tributed DBMSsKey-value stores MapReduce-based systems Relational parallel and distributed DBMSs3.1.3.5.3.Triggers and rules Database views 3.1.3.8.Triggers and rules Database views Integrity checking Distributed database transactionsDistributed database Detailouted database recovery3.1.3.0. 3.1.3.10. 3.1.3.10. 3.1.3.11. 3.1.3.12. 3.1.4.1.Query languagesDistributed database recovery3.1.4.1. 3.1.4.1.Query languagesStructured Query LanguagesStructured Query Languages3.1.4.1.1. 3.1.4.2. 3.1.4.2. 3.1.4.3.1.Informa- tion integrationCall level interfaces Data change Data cchange Data cchange Wrappers (data min- ing) Data cchange Wrappers (data min- ing) Main set of a data in- tegrationMarkeduce lan- guages3.1.5.6. 3.1.5.7.Distributed of a database transaction integrationDistributed database point and point and po						
3.1.3.5.1. 3.1.3.5.2. Key-value stores wares relational parallel and distributed database transactions 3.1.3.6. 1.3.8. Triggers and rules parallels parallels parallels parallels and distributed database transactions 3.1.3.9.1. Structured database transactions Distributed database recovery 3.1.3.9.3 Main memory engines or cossing engine	3.1.3.5.					
3.1.3.5.1. 3.1.3.5.2. Key-value stores wares relational parallel and distributed database transactions 3.1.3.6. 1.3.8. Triggers and rules parallels parallels parallels parallels and distributed database transactions 3.1.3.9.1. Structured database transactions Distributed database recovery 3.1.3.9.3 Main memory engines or cossing engine				tributed DBMSs		
3.1.3.5.2. 3.1.3.5.3. Image: Second sec	3.1.3.5.1				Key-value stores	
3.1.3.5.3. Image: Structure of Constraints of Constrating Constraints of Constraints of Constraints of Constrai						
3.1.3.5.3. Image: state st	0.1.0.0.2.					
A.1.3.6.Triggers and rules Database views Integrity checking Distributed database transactionsand distributed DBMSs3.1.3.7.Triggers and rules Database views Integrity checking Distributed database transactionsDistributed database locking Deadlocks Distributed database recovery3.1.3.9.1.Main memory engines Online analytical pro- cessing engines Stream managementDistributed database recovery3.1.3.10.Main memory engines Online analytical pro- cessing enginesStream management3.1.4.1.1Main memory engines Query languagesStream management3.1.4.1.1Main memory engines Query languagesStructured Query Language3.1.4.2.Main memory engines Query languagesStructured Query Language3.1.4.2.Informa- tion integrationCall level interfaces3.1.4.4.Informa- tion and loading Data cekning Data cekning Medjators and data in- tegrationMapReduce lan- guages3.1.5.4.Informa- tion and loading Data cekning Medjators and data in- tegrationMapReduce lan- guages3.1.5.6.Mediators and data in- tegrationMaprens (data min- ing) Mediators and data in- tegration3.1.5.7.Mediators and data in- tegrationMaprens (data min- ing) Mediators and data in- tegration3.1.5.7.Mediators and data in- tegrationMapReduce lang mappers (data min- ing)3.1.5.7.Mediators and data in- tegration3.1.5.7.Mediators and data in- tegration						
3.1.3.6. 3.1.3.7. 3.1.3.8. 3.1.3.9.1.Triggers and rules Database views Integrity checking Distributed database transactionsDBMSs3.1.3.9.1. 3.1.3.9.2. 3.1.3.9.3.Integrity checking Distributed database transactionsDistributed data Deadlocks Distributed database transactions3.1.3.9.2. 3.1.3.9.3.Main memory engines Online analytical pro- cessing engines Stream managementDistributed data Deadlocks Distributed database query languages3.1.4.1. 3.1.4.1.1. 3.1.4.2. 3.1.4.2. 3.1.4.2. 3.1.4.3.1.2.Main memory engines Stream managementStructured Query Languages3.1.4.1.1. 3.1.4.2. 3.1.4.3.1.Informa- tion integration grationMain memory engines Stream managementStructured Query Languages3.1.4.3.1. 3.1.4.3.1.5.Informa- tion integration grationCall level interfaces Deduplication Extraction transforma- tion and loading Data exchange Data cleaning Wrappers (data min- ing) Mediators and data in- tegration3.1.5.6. 3.1.5.7.Main forma- ting mathematical engines3.1.5.7.Distributed database provide database transaction provide database transport data in- tegration3.1.5.6. 3.1.5.7.Main memory engines structured provide database transport data in- tegration3.1.5.7.Main memory engines tion integration3.1.5.7.Main memory engines tion integration3.1.5.7.Main memory engines tion integration3.1.5.7.Main memory engines tion integration Bata cleaning Mediators and data in- tegration </td <td>3.1.3.5.3.</td> <td></td> <td></td> <td></td> <td></td> <td></td>	3.1.3.5.3.					
3.1.3.6. 3.1.3.7. 3.1.3.7. Triggers and rules 3.1.3.8. Distributed database views 3.1.3.9. Integrity checking 3.1.3.9. Distributed database 3.1.3.9. Distributed database 3.1.3.9.1. Distributed database 3.1.3.9.2. Main memory engines 3.1.3.10. Online analytical processing engines 3.1.3.11. Relational database 3.1.4.1. XML query languages 3.1.4.1. XML query languages 3.1.4.2. XML query languages 3.1.4.2. XML query languages 3.1.4.3.1. Query languages 3.1.4.3.1. Deduplication 3.1.4.3.1. Deduplication 3.1.4.4. Deduplication 3.1.5.1. Deduplication 3.1.5.1. Data exchange 3.1.5.3. Mediators and data integration 3.1.5.6. Mediators and data integration					and distributed	
3.1.3.6. 3.1.3.7. 3.1.3.7. Triggers and rules 3.1.3.8. Distributed database views 3.1.3.9. Integrity checking 3.1.3.9. Distributed database 3.1.3.9. Distributed database 3.1.3.9.1. Distributed database 3.1.3.9.2. Main memory engines 3.1.3.10. Online analytical processing engines 3.1.3.11. Relational database 3.1.4.1. XML query languages 3.1.4.1. XML query languages 3.1.4.2. XML query languages 3.1.4.2. XML query languages 3.1.4.3.1. Query languages 3.1.4.3.1. Deduplication 3.1.4.3.1. Deduplication 3.1.4.4. Deduplication 3.1.5.1. Deduplication 3.1.5.1. Data exchange 3.1.5.3. Mediators and data integration 3.1.5.6. Mediators and data integration					DBMSs	
3.1.3.7. 3.1.3.8. 3.1.3.9.1. Database views 3.1.3.9.1. Distributed database 3.1.3.9.1. Distributed database 3.1.3.9.2. Main memory engines 3.1.3.10. Stream management 3.1.4.1. Stream management 3.1.4.1. Stream management 3.1.4.1. Main memory engines 3.1.4.1. Stream management 3.1.4.2. XML query languages 3.1.4.2. XML query languages 3.1.4.2. Thormation integration 3.1.4.3.1. Description 3.1.5.1. Description 3.1.5.1. Description 3.1.5.1. Description 3.1.5.3. Description 3.1.5.6. Description 3.1.5.6. Description	3136			Triggers and rules		
3.1.3.8. 3.1.3.9. Integrity checking Distributed database 3.1.3.9.1. Distributed database Distributed database 3.1.3.9.2. Stributed database Distributed database 3.1.3.9.2. Main memory engines Distributed database 3.1.3.9.2. Main memory engines Distributed database 3.1.3.10. Stributed database Distributed database 3.1.3.12. Stributed database Distributed database 3.1.4.1. Relational database Distributed database 3.1.4.1. Relational database Structured Query 3.1.4.2. Main memory engines Structured Query 3.1.4.2.1 Query languages Structured Query 3.1.4.2.2 Main query languages for Nucrey anguages 3.1.4.3.1. Query languages for NapReduce languages 3.1.4.3.1. Deduplication Extraction transformation and loading 3.1.5.1. Deduplication Extraction transformation and loading 3.1.5.1. Deduplication Extraction transformation and loading 3.1.5.6. MapReduce mining Mapreduce mining Materion and loading				Database views		
3.1.3.9.1 Distributed database transactions Distributed database transactions 3.1.3.9.1. Distributed database transactions Distributed database transactions 3.1.3.9.2. Distributed database transactions Distributed database transactions 3.1.3.9.3. Main memory engines Distributed database recovery 3.1.3.10. Stream management Stream management 3.1.4.1. Relational database query languages Structured Query Languages 3.1.4.1. XML query languages XTath XQuery 3.1.4.2. Information integration Call level interfaces MapReduce languages 3.1.4.3.1. Deduplication transformation and loading Data exchange MapReduce languages 3.1.4.3.1. Deduplication transformation and loading Data exchange MapReduce languages 3.1.5.1. Deduplication transformation and loading Data exchange MapReduce languages 3.1.5.6. Mediators and data integration Mediators and data integration 3.1.5.6. Mediators and data integration MapReduce languages						
3.1.3.9.1. 3.1.3.9.2. 3.1.3.9.3.transactionsDistributed data locking Deadlocks Distributed database recovery3.1.3.9.2. 3.1.3.9.3.Main memory engines Online analytical pro- cessing engines Stream managementDistributed database recovery3.1.3.10. 3.1.3.12. 3.1.4.1.Query lan- guagesMain memory engines Stream managementDistributed database recovery3.1.4.1. 3.1.4.1.Query lan- guagesRelational database query languagesStructured Query Language3.1.4.1. 3.1.4.2. 3.1.4.2.1 3.1.4.3.1.Informa- tion inte- grationQuery languages for non-relational enginesMapReduce lan- guages3.1.4.4. 3.1.5.1. 3.1.5.1. 3.1.5.1. 3.1.5.3. 3.1.5.4. 3.1.5.5.Informa- tion inte- grationDeduplication Extraction transforma- tion and loading Data cleaning Wrappers (data min- ing) Mediators and data in- tegration3.1.5.6. 3.1.5.7.Entity resolutionDetution						
3.1.3.9.1. Imformation integration Main memory engines Distributed data 3.1.3.9.2. 3.1.3.9.3. Main memory engines Distributed database 3.1.3.10. Main memory engines Distributed database 3.1.3.10. Stream management Frequencies 3.1.4.1. Relational database Frequencies 3.1.4.1. Main memory engines Structured Query 3.1.4.1.1. Main memory engines Structured Query 3.1.4.2. Main memory engines Structured Query 3.1.4.2.1 Main memory engines Structured Query 3.1.4.3.1. Query languages Main memory engines 3.1.4.3.1. Main memory engines Structured Query 3.1.4.3.1. Main memory engines Structured Query 3.1.4.3.1. Main memory engines Structured Query 3.1.5.1. Information integration Call level interfaces Main memory engines 3.1.5.1. Deduplication Deduplication Main memory engines 3.1.5.3. Main memory engines Main memory engines Main memory engines 3.1.5.6. Main memory engines Main memory	3.1.3.9.					
3.1.3.9.3.Main memory engines Online analytical pro- cssing enginesIocking Deadlocks Distributed database recovery3.1.3.10. 3.1.3.12. 3.1.3.12. 3.1.4.1.Query lan- guagesMain memory engines Orline analytical pro- cssing enginesStructured Query Languages3.1.4.1. 3.1.4.1.Relational database query languagesStructured Query Languages3.1.4.1.1. 3.1.4.2. 3.1.4.2.1 3.1.4.2.1 3.1.4.2.1 3.1.4.2.1 3.1.4.3.1Information integration grationStructured Query Languages3.1.4.1.3. 3.1.4.2.1 3.1.4.3.1Information integration grationCall level interfacesMapReduce lan- guages3.1.4.4. 3.1.5.1. 3.1.5.1. 3.1.5.5.Information and loading Data cleaning Wrappers (data min- ing) Mediators and data in- tegrationDeduplication Extraction transformation at cleaning Wrappers (data min- ing) Mediators and data in- tegration				transactions		
3.1.3.9.2. 3.1.3.9.3. Deadlocks 3.1.3.9.3. Main memory engines Distributed database 3.1.3.10. Online analytical processing engines Distributed database 3.1.3.12. Stream management Structured Query 3.1.4.1. Relational database Structured Query 3.1.4.2. XML query languages Structured Query 3.1.4.2.1 Query languages XML query languages 3.1.4.2.1 Query languages Structured Query 3.1.4.3.1. Query languages Main memory engines 3.1.4.3.1. Deadlocks Structured Query 3.1.4.3.1. Query languages Structured Query 3.1.4.3.1. Deduplication Main memory engines 3.1.4.3.1. Deduplication Main memory engines 3.1.5.1. Deduplication Main memory engines 3.1.5.3. Deduplication Main exchange 3.1.5.4. Data cleaning Main exchange 3.1.5.6. Mediators and data in-tegration Mediators and data in-tegration 3.1.5.7. Mediators and data in-tegration Struty endition	3.1.3.9.1.				Distributed data	
3.1.3.9.2. 3.1.3.9.3. Deadlocks 3.1.3.9.3. Main memory engines Distributed database 3.1.3.10. Online analytical processing engines Distributed database 3.1.3.12. Stream management Structured Query 3.1.4.1. Relational database Structured Query 3.1.4.2. XML query languages Structured Query 3.1.4.2.1 Query languages XML query languages 3.1.4.2.1 Query languages Structured Query 3.1.4.3.1. Query languages Main memory engines 3.1.4.3.1. Deadlocks Structured Query 3.1.4.3.1. Query languages Structured Query 3.1.4.3.1. Deduplication Main memory engines 3.1.4.3.1. Deduplication Main memory engines 3.1.5.1. Deduplication Main memory engines 3.1.5.3. Deduplication Main exchange 3.1.5.4. Data cleaning Main exchange 3.1.5.6. Mediators and data in-tegration Mediators and data in-tegration 3.1.5.7. Mediators and data in-tegration Struty endition					locking	
3.1.3.9.3. Image: Stream management Distributed database recovery 3.1.3.10. Stream management Stream management 3.1.3.12. Stream management Stream management 3.1.4.1. Relational database query languages Structured Query Languages 3.1.4.1.1. Main memory engines Stream management 3.1.4.1.1. Main memory engines Structured Query Languages 3.1.4.2. Main memory engines Stream management 3.1.4.2.1 Main memory engines Stream management 3.1.4.2.1 Main memory engines Stream management 3.1.4.3.1. Main memory engines Stream management 3.1.4.3.1. Main memory engines Structured Query Languages 3.1.4.3.1. Main memory engines Structured Query Languages 3.1.4.3.1. Main memory engines Structured Query Languages 3.1.4.3.1. Deduplication Main memory engines 3.1.5.1. Deduplication Structured Query Languages 3.1.5.3. Main memory engines Main memory engines 3.1.5.6. Main memory engines Main memory engines Main memory engines	31392				Deadlocks	
3.1.3.10. 3.1.3.11.Main memory engines Online analytical processing enginesrecovery3.1.3.12. 3.1.4.Query lan- guagesRelational database query languagesStructured Query Languages3.1.4.1.Main memory enginesStructured Query LanguagesStructured Query Language3.1.4.1.1.Main memory enginesStructured Query Languages3.1.4.2.Main memory enginesStructured Query Language3.1.4.2.Main memory enginesStructured Query Language3.1.4.2.1Main memory enginesStructured Query Language3.1.4.2.2Main memory enginesStructured Query Language3.1.4.3.1.Main memory enginesMain memory engines3.1.4.3.1.Main memory enginesStructured Query Language3.1.4.3.1.Main memory enginesMain memory engines3.1.4.3.1.Main memory enginesMain memory engines3.1.4.3.1.Main memory enginesMain memory engines3.1.4.4.Main memory enginesMain memory engines3.1.5.1.Main memory enginesMain memory engines3.1.5.3.Main memory enginesMain memory engines3.1.5.4.Main memory enginesMain memory engines3.1.5.6.Main memory enginesMain memory engines3.1.5.7.Main memory enginesMain memory enginesMain memory enginesMain memory enginesMain memory engines3.1.5.6.Main memory enginesMain memory engines3.1.5.7.Main memory enginesMain memor						
3.1.3.10. Main memory engines Online analytical pro- cessing engines Stream management Vertical pro- cessing engines Stream management 3.1.4.1. Puery lan- guages Relational database query languages Structured Query Language 3.1.4.1.1. Puery languages Structured Query Language Puery Language 3.1.4.2.1 Structured Query Puery Language Puery Language 3.1.4.2.1 Puery languages Puery Language Puery Language 3.1.4.3.1. Puery languages Puery Language Puery Language 3.1.5.1. Puery languages Puery Language Puery Language 3.1.5.3. Puery languages Puery Language Puery Language 3.1.5.6. Puery languages Puery Language Puery Language 3.1.5.7. Puery languages Puery Language Puery Language	3.1.3.9.3.					
3.1.3.11. Online analytical processing engines 3.1.3.12. Stream management 3.1.4.1. Relational database query languages 3.1.4.1.1. Relational database query languages 3.1.4.2.1 XML query languages 3.1.4.2.1 Query languages 3.1.4.2.1 Query languages 3.1.4.2.1 Mapped Level interfaces 3.1.4.2.1 Information integration 3.1.4.3.1. Call level interfaces 3.1.4.4. Deduplication Extraction transformation and loading 3.1.5.1. Data cleaning 3.1.5.4. Wrappers (data mining) 3.1.5.6. Entity resolution					recovery	
3.1.3.12. Query languages cessing engines 3.1.4. Query languages Stream management 3.1.4.1. Relational database query languages Structured Query Languages 3.1.4.1.1. XML query languages Structured Query Languages 3.1.4.2. XML query languages XPath 3.1.4.2.1 Query languages XQuery 3.1.4.2.1 Query languages XML query languages 3.1.4.2.3 Query languages MapReduce languages 3.1.4.3.1. Call level interfaces MapReduce languages 3.1.4.3.1. Deduplication Extraction transformation aloding Data cleaning 3.1.5.1. Data cleaning Wappers (data mining) 3.1.5.4. Mediators and data integration Entity resolution				Main memory engines		
3.1.3.12. Query languages Stream management 3.1.4.1. Relational database query languages Relational database query languages 3.1.4.1.1. XML query languages Structured Query Languages 3.1.4.2. XML query languages XML query languages 3.1.4.2.1 Query languages XML query languages 3.1.4.2.1 non-relational engines XPath XQuery 3.1.4.3.1. Query languages for non-relational engines MapReduce languages 3.1.4.3.1. Deduplication Extraction transformation and loading Data cleaning 3.1.5.1. Data cleaning Data cleaning 3.1.5.4. Mapreduce languages Structure of transformation and loading 3.1.5.6. Extraction transformating Struct cleaning 3.1.5.7. Entity resolution Mediators and data integration	3.1.3.11.			Online analytical pro-		
3.1.3.12. Query languages Stream management 3.1.4.1. Relational database query languages Relational database query languages 3.1.4.1.1. XML query languages Structured Query Languages 3.1.4.2. XML query languages XML query languages 3.1.4.2.1 Query languages XML query languages 3.1.4.2.1 non-relational engines XPath XQuery 3.1.4.3.1. Query languages for non-relational engines MapReduce languages 3.1.4.3.1. Deduplication Extraction transformation and loading Data cleaning 3.1.5.1. Data cleaning Data cleaning 3.1.5.4. Mapreduce languages Structure of transformation and loading 3.1.5.6. Extraction transformating Struct cleaning 3.1.5.7. Entity resolution Mediators and data integration						
3.1.4. Query languages 3.1.4.1. Relational database query languages 3.1.4.1.1 XML query languages 3.1.4.2. XML query languages 3.1.4.2.1 Query languages 3.1.4.2.1 Query languages 3.1.4.2.1 Query languages 3.1.4.2.2 Query languages 3.1.4.3.1 Query languages for non-relational engines 3.1.4.3.1 Call level interfaces 3.1.4.4. Call level interfaces 3.1.5.1. Deduplication Extraction transformation and loading Data celonaing 3.1.5.3. Data celonaing 3.1.5.4. Mediators and data integration 3.1.5.6. Entity resolution	3 1 3 1 2					
3.1.4.1. guages Relational database query languages Structured Query Language 3.1.4.1.1. XML query languages Structured Query Language 3.1.4.2. XML query languages XML query languages 3.1.4.2.1 Query languages for non-relational engines XQuery 3.1.4.3.1. Call level interfaces MapReduce lan- guages 3.1.4.4. Deduplication Extraction transforma- tion and loading Data cleaning Deduplication Extraction transforma- ting 3.1.5.3. Mapres (data min- ing) Data cleaning Wrappers (data min- tegration 3.1.5.6. Entity resolution			0 1	Stream management		
3.1.4.1. Relational database query languages 3.1.4.1.1. XML query languages 3.1.4.2.1 XML query languages 3.1.4.2.1 XML query languages 3.1.4.2.1 Query languages for non-relational engines 3.1.4.3.1 Call level interfaces 3.1.4.4. Deduplication Extraction transformation and loading Data exchange Data exchange 3.1.5.3. Data cleaning Wrappers (data mining) 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution	3.1.4.					
3.1.4.1.1. Image: second sec			guages			
3.1.4.1.1. Number of the second s	3.1.4.1.					
3.1.4.1.1. Number of the second s				query languages		
3.1.4.2. SML query languages Language 3.1.4.2.1 XML query languages XPath 3.1.4.2.2. Query languages for MapReduce languages 3.1.4.3.1. Call level interfaces MapReduce languages 3.1.4.4. Call level interfaces MapReduce languages 3.1.5.1. Deduplication Extraction transformation and loading 3.1.5.3. Data celaning Wrappers (data mining) 3.1.5.6. Mediators and data integration Entity resolution	3.1.4.1.1.				Structured Querv	
3.1.4.2. XML query languages XPath XQuery 3.1.4.2.1 Query languages for non-relational engines MapReduce languages 3.1.4.3.1 Call level interfaces MapReduce languages 3.1.4.4. Deduplication Extraction transformation and loading Data echaning 3.1.5.1 Data echaning Data echaning 3.1.5.4. Data echaning MapReduce languages 3.1.5.6. Extraction transformation and loading Data echaning 3.1.5.7. Entity resolution Entity resolution						
3.1.4.2.1 3.1.4.2.1 XPath 3.1.4.2.2. Query languages for non-relational engines MapReduce languages 3.1.4.3.1. Information integration Call level interfaces 3.1.4.4. State Deduplication Extraction transformation and loading 3.1.5.3. Data cchange Data cchange 3.1.5.4. Wrappers (data mining) Mediators and data integration 3.1.5.7. Entity resolution Entity resolution	3142			XML query languages	0.000	
3.1.4.2.2. 3.1.4.3.1. 3.1.4.3.1. Query languages for non-relational engines MapReduce languages 3.1.4.3.1. Call level interfaces MapReduce languages 3.1.4.4. Call level interfaces Second Seco				runn query languages	V Path	
3.1.4.3. Query languages for non-relational engines 3.1.4.3.1. Information integration 3.1.4.4. Call level interfaces 3.1.5.1. Deduplication Extraction transformation and loading 3.1.5.3. Data cclaning 3.1.5.4. Wrappers (data mining) 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution						
3.1.4.3.1. Information Call level interfaces MapReduce languages 3.1.4.4. Call level interfaces Gall level interfaces 3.1.5.1. Deduplication Extraction transformation and dating 3.1.5.3. Data echange Data echange 3.1.5.4. MapReduce languages 3.1.5.6. MapReduce languages 3.1.5.7. Extraction transformation					AQuery	
3.1.4.3.1. MapReduce languages 3.1.4.4. Information integration Call level interfaces 3.1.5.1. Deduplication Extraction transformation and loading Data echange 3.1.5.3. Data cleaning MapReduce languages 3.1.5.4. Data echange MapReduce languages 3.1.5.5. MapReduce languages MapReduce languages 3.1.5.6. MapReduce languages MapReduce languages 3.1.5.7. Detains Detains	3.1.4.3.					
3.1.4.4. Information Call level interfaces guages 3.1.5.1. Information Deduplication Strate 3.1.5.1. Deduplication Extraction transformation and loading Data celaning 3.1.5.3. Data celaning Wrappers (data mining) Mediators and data integration 3.1.5.6. Entity resolution Entity resolution Entity resolution				non-relational engines		
3.1.4.4. Information Call level interfaces guages 3.1.5.1. Information Deduplication Strate 3.1.5.1. Deduplication Extraction transformation and loading Data celaning 3.1.5.3. Data celaning Wrappers (data mining) Mediators and data integration 3.1.5.6. Entity resolution Entity resolution Entity resolution	3.1.4.3.1.				MapReduce lan-	
3.1.4.4. Call level interfaces 3.1.5. Information integration 3.1.5.1. Deduplication 3.1.5.2. Extraction transformation and loading 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution						
3.1.5. Information 3.1.5.1. Deduplication 3.1.5.2. Deduplication 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution	2144			Call lovel interface-	Pare Pos	
3.1.5.1. ition integration 3.1.5.2. Deduplication 3.1.5.3. Extraction transformation and loading 3.1.5.4. Data exchange 3.1.5.5. Wrappers (data mining) 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution			т.с.	Can level interfaces		
3.1.5.1. gration 3.1.5.2. Deduplication Extraction transforma- tion and loading 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.5. Wrappers (data min- ing) 3.1.5.6. Mediators and data in- tegration 3.1.5.7. Entity resolution	3.1.5.					
3.1.5.1. Deduplication 3.1.5.2. Extraction transformation and loading 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.6. Wrappers (data mining) 3.1.5.7. Entity resolution						
3.1.5.1. Deduplication 3.1.5.2. Extraction transformation and loading 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.6. Wrappers (data mining) 3.1.5.7. Entity resolution			gration			
3.1.5.2. Extraction transformation and loading 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.5. Wrappers (data mining) 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution	3.1.5.1.			Deduplication		
3.1.5.3. tion and loading 3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.5. Wrappers (data min- ing) 3.1.5.6. Mediators and data in- tegration 3.1.5.7. Entity resolution						
3.1.5.3. Data exchange 3.1.5.4. Data cleaning 3.1.5.5. Wrappers (data min- ing) 3.1.5.6. Mediators and data in- tegration 3.1.5.7. Entity resolution	0.1.0.4.					
3.1.5.4. Data cleaning 3.1.5.5. Wrappers (data mining) 3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution						
3.1.5.5. Wrappers (data min- ing) 3.1.5.6. Mediators and data in- tegration 3.1.5.7. Entity resolution						
3.1.5.6. 3.1.5.7. Bridiators and data in- tegration Entity resolution				Data cleaning		
3.1.5.6. 3.1.5.7. Bridiators and data in- tegration Entity resolution				Wrappers (data min-		
3.1.5.6. Mediators and data integration 3.1.5.7. Entity resolution	3.1.5.4.					
3.1.5.7. tegration Entity resolution	3.1.5.4.					
3.1.5.7. Entity resolution	3.1.5.4. 3.1.5.5.			ing)		
	3.1.5.4. 3.1.5.5.			ing) Mediators and data in-		
3.1.5.8. Data warehouses	3.1.5.4. 3.1.5.5. 3.1.5.6.			ing) Mediators and data in- tegration		
	3.1.5.4. 3.1.5.5. 3.1.5.6. 3.1.5.7.			ing) Mediators and data in- tegration Entity resolution		

3.1.5.9. Informa-tributers are strained and states and states are strained and s						
1.2.1. tion systems applications Data mining Data mining Item-based 3.2.1.1. 3.2.1.2. Science Type of association 3.2.1.2.1.* Science Type of association 3.2.1.2.2.* Clustering Item-based 3.2.1.3.* Science Type of association 3.2.1.4.* Science Process 3.2.1.4.* Process Process 3.2.1.7.* Process Process 3.2.1.7.* Process Process 3.2.1.1.* Process Process 3.2.1.9.* Process Process 3.1.1.* Process Process 3.1.1.* Process Process	3.1.5.9.	T C		Federated databases		
3.2.1.tems tionsData min- ingData cleaning Collaborative filtering Collaborative filtering Collaborative filtering S2.1.2.2.3.Her-based Scalable3.2.1.2.2.* 3.2.1.3.13.2.1.2.* 3.2.1.3.1ScalableTypes of association rules scalable3.2.1.3.1S.2.1.3.1ScalableTypes of association rules scalable3.2.1.4.1ScalableTypes of association rules scalable3.2.1.4.2.* 3.2.1.4.1ScalableTypes of association rules formal computation3.2.1.4.1ScalableTypes of association rules scalable3.2.1.4.2.* 3.2.1.4.3ScalableTypes of association rules formal computation3.2.1.4.1ScalableScalable3.2.1.4.2.* 3.2.1.4.5.*ScalableTypes of association rules formal computation3.2.1.4.7.* 3.2.1.7.*ScalableScalable3.2.1.7.* 3.2.1.7.*ScalableScalable3.2.1.7.* 3.2.1.7.*ScalableScalable3.2.1.7.* 3.2.1.7.*ScalableScalable3.2.1.7.* 3.2.1.7.*ScalableScalable3.2.1.7.* 3.2.1.7.*ScalableScalable3.2.1.7.* 3.2.1.7.*ScalableScalable3.1.1.2.* 3.2.1.9.*ScalableScalable3.1.3.1 3.3.1.4ScalableScalable3.1.4.3 3.3.1.5ScalableScalable3.1.5.*ScalableScalable3.1.6.*ScalableScalable3.1.1.1.*ScalableScalable3.	3.2.					
3.2.1. a.2.1.1 3.2.1.1. a.2.1.2. 3.2.1.2.1. a.3.1.3.* 3.2.1.2.1. a.3.1.3.* 3.2.1.2.1.* a.3.1.3.* 3.2.1.2.* a.3.1.* 3.2.1.4.* a.4.* 3.2.1.4.* a.4.* 3.2.1.4.* a.4.* 3.2.1.4.* a.4.* 3.2.1.5.* a.4.* 3.2.1.5.* a.4.* 3.2.1.5.* a.4.* 3.2.1.5.* a.4.* 3.1.1.* a.4.*						
3.2.1. 3.2.1.2. 3.2.1.2.* 3.2.1.3. 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.4.* 3.2.1.4.** 3.3.1.** 3.3.						
3.2.1. Image: Solution of the second secon						
3.2.1.1 3.2.1.2. 3.2.1.2.* 3.2.1.2.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.4.* 3.2.1.7.* 3.2.1.1.* 3.2.1.1.* 3.2.1.1.* 3.2.1.1.* 3.2.1.1.* 3.3.1.* 3.3. 3.3. 3.3. 3.3. 3.3. 3.3. 3.3. 3.3. 3.3. 3.3. 3.3.1.* <br< td=""><td>9.9.1</td><td>tions</td><td>Data :</td><td></td><td></td><td></td></br<>	9.9.1	tions	Data :			
3.2.1.1. 3.2.1.2.* 3.2.1.2.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.4.* 3.2.1.5.* 3.2.1.7.* 3.2.1.0.* 3.2.1.9.* 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.4.* 3.3.1.6.	3.2.1.					
3.2.1.2. S.2.1.2.* Scalable Item-based 3.2.1.2.* Scalable Types of association 3.2.1.3.* Types of association rules 3.2.1.3.* Types of association rules 3.2.1.3.* Types of association rules 3.2.1.4.* Types of association rules 3.2.1.6. Types of association rules 3.2.1.7.* Types of association Types of association 3.2.1.1.1.*	0.0.1.1		ing	D ()		
3.2.1.2.1.* 3.2.1.2.1.* 3.2.1.3.3 3.2.1.3.* 3.2.1.3.3 3.2.1.3.* 3.2.1.3.3 Types of association rules 3.2.1.3.3.* Types of association rules 3.2.1.3.3.* Types of association rules 3.2.1.4.1.* Babbe 3.2.1.4.1.* Massive data clustering 3.2.1.4.3.* Clustering 3.2.1.4.4.* Clustering 3.2.1.4.4.* Clustering 3.2.1.4.7.* Scalable 3.2.1.4.7.* Scalable 3.2.1.4.7.* Scalable 3.2.1.7.* Scalable 3.2.1.5.* <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
3.2.1.2.2.* S.2.1.3 Scalable 3.2.1.3.1.* Types of association 3.2.1.3.* Factorial 3.2.1.3.* Factorial 3.2.1.3.* Factorial 3.2.1.3.* Factorial 3.2.1.4.* Factorial 3.2.1.6.* Factorial 3.2.1.7.* Factorial 3.2.1.1.1.* Factorial 3.2.1.1.1.* Factorial				Collaborative filtering		
3.2.1.3. 3.2.1.3. 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.3.* 3.2.1.4.* 3.2.1.5.* 3.2.1.7.* 3.2.1.3.* 3.3.1.3.* 3.3.1.3.* 3.3.1.3.* 3.3.1.3.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.4.* 3.3.1.5.* 3.						
3.2.1.3.1.* Types of association 3.2.1.3.2.* 3.2.1.3.3.* 3.2.1.3.3.* 3.2.1.4.* 3.2.1.4.* Clustering 3.2.1.4.* Parallel computation 3.2.1.6. Parallel computation 3.2.1.7.* Parallel computation 3.2.1.7.* Parallel computation 3.2.1.7.* Parallel computation 3.2.1.7.* Parallel computation 3.2.1.1.* Parallel computation 3.2.1.0.* Parallel computation 3.1.1.2.* Parallel computation 3.1.2.* Parallel computat				Ai-+i1	Scalable	
3.2.1.3.2* 3.2.1.3.8* 3.2.1.4.3.8* Informa- tion Informa- ti				Association rules	Types of accoriation	
3.2.1.3.2.* 3.2.1.3.2.* Interestingness Parallel computation 3.2.1.4.* 3.2.1.4.* Clustering Parallel computation 3.2.1.4.* 3.2.1.4.* Parallel computation Massive data clustering 3.2.1.4.* S.2.1.4.* Parallel computation Massive data clustering 3.2.1.4.* S.2.1.4.* Parallel computation Massive data clustering 3.2.1.4.* S.2.1.4.* Parallel computation Parallel computation 3.2.1.4.* Parallel computation Parallel computation 3.2.1.4.* Parallel computation Parallel computation 3.2.1.4.* Parallel computation Parallel computation 3.2.1.6. S.2.1.6.* Parallel computation 3.2.1.7.3.* Parallel computation Parallel computation 3.2.1.7.4.* Parallel computation Parallel computation 3.2.1.7.4.* Parallel computation Parallel computation 3.2.1.1.* Parallel computation Parallel computation 3.2.1.0.* Parallel computation Parallel computation 3.1.1.* Parallel computation Parallel computation 3.1.1.*	3.2.1.3.1.					
3.21.3.3.* 3.2.1.4.1.* 3.2.1.4.1.* 3.2.1.4.3.* 3.2.1.4.3.* 3.2.1.4.3.* 3.2.1.4.4.* 3.2.1.4.4.* 3.2.1.4.5.* 3.2.1.4.5.* 3.2.1.4.5.* 3.2.1.4.7.* 3.2.1.5.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.4.* 3.2.1.1.1.* 3.2.1.7.4.* 3.3.1.3.* 3.3.1.4.* 3.3.1.4.* 3.3.1.5.* 3.3.1.5.* 3.3.1.6.* 3.3.1.4.* 3.3.1.5.* 3.3.1.5.* 3.3.1.6.* 3.3.1.4.* 3.3.1.5.* 3.3.1.5.* 3.3.1.5.* 3.3.1.5.* 3.3.1.6.* 3.3.1	3 2 1 3 2 *					
3.2.1.4. 3.2.1.4.2.* 3.2.1.4.3.* 3.2.1.4.4.* 3.2.1.4.4.* 3.2.1.4.6.* 3.2.1.4.6.* 3.2.1.6. 3.2.1.6. 3.2.1.6. 3.2.1.7.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.3.1.0.* 3.3.1.2.* 3.3.1.3.* 3.3.1.4.* 1.10 ^{cmmatrice and backee and backee}						
3.2.1.4.1.* 3.2.1.4.2.* 3.2.1.4.2.* 3.2.1.4.2.* 3.2.1.4.4.* 3.2.1.4.4.* 3.2.1.4.4.* 3.2.1.4.4.* 3.2.1.6.* 3.2.1.6. 3.2.1.7.* 3.3.1.* 1.1.* 3.3.1.* 3.3.1.* 3.3.1.* 3.3.1.* 1.1.* 3.1.* 3.3.1.* 1.1.* 3.1.* 3.3.1.* 1.1.* 3.3.1.* 1.1.* 3.1.* 3.3.1.* 1.* 3.3.1.* 1.* 3.1.* 3.1.* 3.3.1.* 1.* 3.* 3.* 3.				Clustering	i uruner computation	
3.2.1.4.2.* 3.2.1.4.3.* cring Consensus clustering 3.2.1.4.3.* 3.2.1.4.3.* Additive clustering Fourse (Consensus clustering 3.2.1.4.5.* S.2.1.4.5.* Nearest-neighbor Feature weight clustering 3.2.1.4.6.* S.2.1.5. Biclustering Conceptual clustering 3.2.1.6. S.2.1.7.* Data stream mining Graph partitioning 3.2.1.7.3.* Process mining Graph based conception 3.2.1.7.3.* Process mining Anomaly detection 3.2.1.7.3.* Process mining Text categorization 3.2.1.1.1.* Process mining Text categorization 3.2.1.9.* Process mining Market graph 3.2.1.9.* Process mining Market graph 3.2.1.9.* Process mining Market graph 3.3.1.0.* World Web mining Site wrapping 3.3.1.3 Market graph Surfacing 3.3.1.3 Informa-tion Procesen analysis Market graph 3.3.1.3 Informa-tion Procese mining Market graph 3.3.1.3 Informa-tion Pocement Pocement				orabioring	Massive data clus-	
3.21.4.2.* 3.2.1.4.4.* 3.2.1.4.5.*Nearest-neighbor search Data stream mining Graph partitioning Frequent graph min- ing 3.2.1.7.1.* 3.2.1.7.1.* 3.2.1.7.3.*Nearest-neighbor search Data stream mining Graph partitioning Frequent graph min- ing Graph based concep- tual clustering Additive dustering Additive dustering Additive dustering How and a stream mining Graph partitioning Frequent graph min- ing Graph partitioning Frequent graph min- ing 3.2.1.7.3.*3.2.1.7.4.* 3.2.1.7.5.*Process mining Text mining 3.2.1.7.5.*Frequent graph min- ing Graph based concep- tual clustering Additive dustering Additive dustering Frequent graph min- ing Back area dustering Additive dustering Frequent graph Back area dustering Sequence mining Data extraction and integrationConcept Concept Persentition Surface analysis Sault and with the dustering Secuence result dedu- plication3.1.4.						
3.2.1.4.3.* S.2.1.4.3.* Fuzzy clustering Additive clustering 3.2.1.4.5.* Additive clustering Concorput clustering 3.2.1.4.6.* Data stream mining Concorput clustering 3.2.1.6. Data stream mining Graph partitioning 3.2.1.7.* Data stream mining Graph partitioning 3.2.1.7.3.* Process mining Graph based conceptual clustering 3.2.1.7.4.* Process mining Text categorization 3.2.1.1.* Process mining Text categorization 3.2.1.1.* Process mining Text categorization 3.2.1.1.* Process mining Text categorization 3.2.1.0.* Process mining Traiter categorization	3.2.1.4.2.*					
3.2.1.4.4.* 3.2.1.4.5.* 3.2.1.4.6.* 3.2.1.4.6.* 3.2.1.6. 3.2.1.6. 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.1.2.* 3.2.1.1.2.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.3.1.0.* 3.3.1.2.* 3.3.1.3.* 3.3.1.3.* 3.3.1.4.* 3.3.1.5.* 3.3.1.6.* 3.4.1.* 1.1.2.* 3.4.1.* 1.1.2.* 3.3.1.2.* 3.3.1.4.* 3.3.1.2.* 3.3.1.4.* 3.3.1.5.* 3.3.1.5.* 3.3.1.6.* 3.4.1.* 1.1.2.* 3.4.1.* 1.1.2.* 3.3.1.2.* 3.3.1.4.* 3.3.1.5.* 3.3.1.6.* 3.4.1.* 1.1.2.* 3.3.1.6.* 3.4.1.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1.1.2.* 1						
3.2.1.4.5.*						
3.2.1.4.6.*Nearest-neighbor search Data stream mining Graph partitioning Frequent graph mining Graph partitioning Frequent graph mining Graph based concep- tan clustering Anomaly detection Critical nodes detec- tion 3.2.1.7.3.*3.2.1.7.3.*Process mining Text miningText categorization Key-phrase indexing Sequence mining Text categorization Key-phrase indexing 3.2.1.7.3.*3.2.1.7.4.* 3.2.1.7.5.*Process mining Text miningText categorization Key-phrase indexing Market graph3.2.1.7.4.* 3.2.1.11.2.*Process mining Text miningText categorization Key-phrase indexing Market graph3.2.1.9.* 3.2.1.9.*World Wide WebFormal concept analy- sisRule and pattern discovery Trajectory cluster- ing Market graph3.3.1.0. 3.3.1.3.World Wide WebWeb min- ing Site wrapping Data extraction and integrationDeep web Surfacing Search results dedu- plication3.1.1.1 3.3.1.3.Informa- tion re- trievalWeb log analysis Traffic analysis Knowledge discoveryDeep web Surfacing Search results dedu- plication3.4.1.Informa- tion re- trievalDocument represen- tationDocument represen- tation						
3.2.1.4.6.* 3.2.1.4.7.* 3.2.1.5. 3.2.1.4.7.* 3.2.1.5. 3.2.1.6. 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.7.* 3.2.1.1.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.10.* 3.2.1.9.* 3.2.1.9.* 3.3.1 3.3.1.3 3.3.1.3 3.3.1.3 3.3.1.3 3.3.1.3 3.3.1.3 3.3.1.3 3.3.1.3 3.3.1.4 3.3.1.5 3.3.1.3 3.3.1.4 3.3.1.5 3.3.1.3 3.3.1.4 3.3.1.6 3.3.1.6 3.3.1.4 3.3.1.6 3.3.1.6 3.3.1.4 3.3.1.6 3.3.1						
3.2.1.4.7.*ing Biclustering3.2.1.6.3.2.1.6.3.2.1.7.1.*Data stream mining Graph mining3.2.1.7.1.*3.2.1.7.2.*3.2.1.7.3.*3.2.1.7.3.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.7.4.*3.2.1.1.*3.2.1.1.*3.2.1.1.*3.2.1.1.*3.2.1.1.1.*3.2.1.1.*3.2.1.1.1.*3.2.1.1.1.*3.2.1.1.2.*3.1.1.2.3.3.1.3.3.3.1.3.3.3.1.3.3.3.1.3.3.3.1.3.3.3.1.3.3.3.1.4. <t< td=""><td>3.2.1.4.6.*</td><td></td><td></td><td></td><td></td><td></td></t<>	3.2.1.4.6.*					
3.2.1.4.7.* 3.2.1.5. 3.2.1.6. 3.2.1.7.* 3.2.1.7.* 3.2.1.7.2.* 3.2.1.7.4.* 3.2.1.7.5.* 3.2.1.7.5.* 3.2.1.7.5.* 3.2.1.7.* 3.2.1.11.4.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.3.1.4. 3.4.1.2.* 3.4						
3.2.1.5. Nearest-neighbor 3.2.1.6. 3.2.1.7.1* 3.2.1.7.1* Graph mining 3.2.1.7.1* Graph mining 3.2.1.7.1* Graph mining 3.2.1.7.3.* Graph mining 3.2.1.7.4.* Process mining 3.2.1.7.4.* Process mining 3.2.1.7.4.* Process mining 3.2.1.7.4.* Process mining 3.2.1.1.* Process mining 3.2.1.1.* Process mining 3.2.1.1.* Process mining 3.2.1.1.1.* Process mining 3.2.1.1.2.* Process mining 3.1.3.* Process mining 3.3.1.*	3.2.1.4.7.*					
3.2.1.6. 3.2.1.7.* 3.2.1.7.1.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.5.*Search Data stream mining Graph mining Graph partitioning Frequent graph min- ing Graph based concep- tuol clustering Anomaly detection Critical nodes detec- tion3.2.1.7.4.* 3.2.1.7.5.*Process mining Text mining Text mining Data mining tools Sequence mining 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.*Process mining Text mining Text attegorization Key-phrase indexing Data mining tools Sequence mining Market graph3.1.2 3.3.1.3.World Wide Web Mind 3.3.1.3.World Web min- ing3.3.1.2 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.4.*Morld Web min- ing3.4.1.Informa- trievalDocument represen- trievalSite wrapping Data extraction and integration3.4.1.Informa- trievalDocument represen- taionProcess mining Process mining Trafic analysis Trafic analysis	3.2.1.5.			Nearest-neighbor	-	
3.2.1.7.* 3.2.1.7.1.* 3.2.1.7.2.*Graph miningGraph partitioning Frequent graph mining Graph based concep- tual clustering Anomaly detection Critical nodes detec- tion3.2.1.7.4.* 3.2.1.7.5.*Process mining Text miningText categorization Key-phrase indexing3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.9.* 3.2.1.9.*Process mining Text miningRule and pattern discovery Trajectory cluster- ing Market graph3.1.0.* 3.3.1.3.World Wide WebWeb mining mingFormal concept analy- sisRule and pattern discovery Trajectory cluster- ing Market graph3.1.1.3. 3.3.1.3.World Wide WebWeb mining mingSite wrapping Data extraction and integrationDeep web Surfacing Search results dedu- plication3.1.4. 3.1.5. 3.3.1.6.* 3.4.1.Informa- triovalDocument representationWeb log analysis Trafic ana				search		
3.2.1.7.1.* 3.2.1.7.3.* 3.2.1.7.3.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.5.*Graph partitioning Frequent graph min- ing Graph based concep- tual clustering Anomaly detection Critical nodes detec- tion3.2.1.7.4.* 3.2.1.7.5.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.11.* 3.2.1.10.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.3.1.0.* 3.3.1.3.Process mining Text categorization Key-phrase indexing Data mining tools Sequence mining Text categorization Key-phrase indexing Market graph3.1.3. 3.1.3. 3.1.3.World Wide WebFormal concept analy- sisWeb min- ing 3.3.1.3. 3.1.3.1. 3.3.1.3.Site wrapping Data extraction and integration3.1.3.1 3.3.1.3. 3.3.1.6.* 3.4.1.Informa- trieval4.1.Informa- trievalDocument represen- tationJunctionDocument represen- trievalJunctionDocument represen- trievalJunctionDocument represen- trieval	3.2.1.6.			Data stream mining		
3.2.1.7.2.* Image: state of the state	3.2.1.7.*			Graph mining		
3.2.1.7.3.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.5.*ing3.2.1.7.4.* 3.2.1.7.5.*Frocess mining Text mining3.2.1.7.4.* 3.2.1.7.5.*Process mining Text mining3.2.1.11.* 3.2.1.11.** 3.2.1.11.** 3.2.1.9.1.**Process mining Text mining3.2.1.9.1.* 3.2.1.9.1.**Data mining tools Sequence mining3.2.1.9.2.* 3.2.1.12.**Formal concept analy- sis3.3. 3.3.1.3.World Wide WebFormal concept analy- sis3.3.1.3. 3.3.1.3.World Wide WebSite wrapping Data extraction and integration3.3.1.3. 3.3.1.3.Informa- tion re- trievalSite wrapping Data extraction and integration3.4.1.Informa- tion re- trievalDocument represen- trievalA.1.Document represen- trievalDocument represen- taion	3.2.1.7.1.*				Graph partitioning	
3.2.1.7.3.* Graph based conceptual clustering 3.2.1.7.4.* 3.2.1.7.5.* 3.2.1.7.5.* Process mining 3.2.1.11.* Process mining 3.2.1.10.* Process mining 3.2.1.10.* Process mining 3.2.1.9.* Process mining 3.2.1.12.* Process mining 3.3.1. Process mining 3.3.1.2 Process mining 3.3.1.2 Process mining 3.3.1.3.1 Process mining 3.3.1.3.1 Process mining 3.3.1.3.1 Process mining 3.3.1.3.1 Process mining 3.3.1.4. Process mining	3.2.1.7.2.*				Frequent graph min-	
3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.4.* 3.2.1.7.* 3.2.1.1.4.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* 3.3.1.2.* 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.4. 3.3.1.6.* 3.4.1. Information retrived The formation of the format						
3.2.1.7.4.* 3.2.1.7.5.*Anomaly detection Critical nodes detec- tion3.2.1.8.* 3.2.1.11.* 3.2.1.11.2.* 3.2.1.0.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.2.*Process mining Text miningAnomaly detection Critical nodes detec- tion3.2.1.9.2.* 3.2.1.9.2.* 3.3.1.3.World Wide WebProcess mining tools Sequence miningText categorization Key-phrase indexing3.3.1. 3.3.1.2. 3.3.1.3.World Wide WebFormal concept analy- sisRule and pattern discovery Trajectory cluster- ing Data extraction and integration3.3.1.3.1 3.3.1.3.1 3.3.1.3.2.Informa- trivealSite wrapping Data extraction and integrationDeep web Surfacing Search results dedu- plication3.4.1.Informa- trivealDocument represen- trivealWeb log analysis Traffic analysis Knowledge discovery	3.2.1.7.3.*					
3.2.1.7.5.* Image: constraint of the second sec						
3.2.1.8.* 3.2.1.11.* 3.2.1.11.2.* 3.2.1.11.2.* 3.2.1.0.* 3.2.1.9.* 3.2.1.9.1.*Hore and pattern discovery Trajectory cluster- ing Market graph3.2.1.9.2.* 3.2.1.9.3.* 3.2.1.9.3.* 3.3.1.3.World Wide WebFormal concept analy- sisRule and pattern discovery Trajectory cluster- ing Market graph3.3.1.2. 3.3.1.3.World Wide WebWeb min- ing Site wrapping Data extraction and integrationDeep web Surfacing Surfacing Surfacing Surfacing Surfacing Surfacing Sarth and the provent Surfacing3.1.4. 3.3.1.6.* 3.4.1.Informa- trievalWeb log analysis Trafic analysis3.4.1.Informa- trieval						
3.2.1.8.* Process mining Text categorization 3.2.1.11.* Text mining Text mining 3.2.1.10.* Text mining Text mining 3.2.1.0.* Text mining Text mining 3.2.1.0.* Text mining Rule and pattern discovery 3.2.1.0.* Trajectory cluster-ing Market graph 3.2.1.12.* Text mining Site wrapping Market graph 3.3.1.2. Text mining Site wrapping Text market graph 3.3.1.3.1 Text mining Market graph 3.3.1.3.1 Text mining Site wrapping Text market graph 3.3.1.3.1 Text market graph Text market graph Text market graph 3.3.1.3.1 Text market graph Text market graph Text market graph 3.3.1.3.1 Text market graph Text market graph Text market graph 3.3.1.3.1 Text market graph Text market graph Text market graph 3.3.1.4. T	3.2.1.7.5.*					
3.2.1.11.* 3.2.1.11.* 3.2.1.11.2.* Data mining tools 3.2.1.9.* Data mining tools 3.2.1.9.* Rule and pattern discovery 3.2.1.9.* Trajectory clustering 3.2.1.9.* World 3.3. World Wide Web 3.3.1.2. World Wide Web 3.3.1.2. Site wrapping 3.3.1.3. World Wide Web 3.3.1.3. Web mining 3.3.1.3. World Wide Web 3.3.1.4. Information 3.3.1.4. Information retriver 3.4.1. Information retriver A.4. Information retriver				_	tion	
3.2.1.11.1.* Text categorization 3.2.1.10.* Data mining tools 3.2.1.9.* Data mining tools 3.2.1.9.* Bata mining tools 3.2.1.12.* Bata mining tools 3.3.1 Bata mining tools 3.3.1.2 Bata mining tools 3.3.1.3.1 Bata mining tools 3.3.1.4 Information 3.3.1.4. Information 3.4.1. Document trepresention Bata mining tools Bata mining tools Bocument trepresention Bata mining tools Bocument tools Bata mining tools <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td></td<>						
3.2.1.11.2.* 3.2.1.0.* 3.2.1.9.* 3.2.1.9.* 3.2.1.9.* Data mining tools 3.2.1.9.* Sequence mining 3.2.1.9.1.* Data mining tools 3.2.1.9.1.* Formal concept analysis 3.3.1.2. World 3.3.1.2. World Wide Web 3.3.1.2. World Wide Web 3.3.1.3. Web mining 3.3.1.4. Web mining 3.3.1.4. Market graph 3.3.1.4. Market graph 3.3.1.6.* Information Market graph Market graph Document represention Web log analysis Traffic analysis Market graph Secure results deduplication Market graph Market graph Market graph Secure results deduplication				Text mining		
3.2.1.10.* 3.2.1.9.* 3.2.1.9.1.* 3.2.1.9.2.* 3.2.1.9.2.* 3.3. 3.3.1.2. 3.3.1.3.1 3.3.1.4. 3.3.1.6.* 3.4.1. Document represen- triveval Document represen- tation						
3.2.1.9.* Sequence mining Rule and pattern discovery 3.2.1.9.2.* Formal concept analysis Rule and pattern discovery 3.2.1.9.3.* World Formal concept analysis Market graph 3.3.1. Wide Web Site wrapping Data extraction and integration 3.3.1.3.1 Site wrapping Data extraction and integration Deep web 3.3.1.3.1 Information retrieval Web log analysis Surfacing 3.3.1.6.* Information retrieval Document representation Document representation					Key-phrase indexing	
3.2.1.9.1.* Rule and pattern discovery Trajectory clustering 3.2.1.9.2.* Formal concept analysis 3.3. World Wide Web 3.3.1.2. Site wrapping Data extraction and integration 3.3.1.3.1 Deep web Surfacing Search results deduplication 3.3.1.4. Information retrieval 3.3.1.6.* Information retrieval 3.4.1. Document representiation						
3.2.1.9.2.* 3.2.1.9.3.* 3.2.1.12.* 3.3. 3.3.1.2. 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.2. 3.3.1.3.1 3.3.1.3.2. 3.3.1.3.3. 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.4. 3.3.1.6.* 3.4.1. Information retrieval Document representation and integration Deep web Surfacing Search results dedupication Search results dedupicat				Sequence mining	D 1 44	
3.2.1.9.2.*	3.2.1.9.1.*					
3.2.1.9.3.* 3.2.1.9.3.* 3.2.1.12.* 3.3. 3.3. 3.3.1.2. 3.3.1.3. 3.3.1.3.1 3.3.1.3.1 3.3.1.3.1 3.3.1.3.2. 3.3.1.3.3. 3.3.1.3.3. 3.3.1.3.3. 3.3.1.4. 3.3.1.5. 3.3.1.6.* 3.4.1. Informa- tion re- trieval Document represen- tation and integration and integrat	22102*					
3.2.1.9.3.* 3.2.1.12.* 3.3. 3.3. 3.3. 3.3. 3.3.1.2. 3.3.1.3. 3.3.1.3. 3.3.1.3. 3.3.1.3.1 3.3.1.3.2. 3.3.1.3.2. 3.3.1.3.2. 3.3.1.4. 3.3.1.5. 3.3.1.6.* 3.4.1. Morket graph Web min- ing Site wrapping Data extraction and integration Web log analysis Traffic ana	3.2.1.9.2.					
3.2.1.12.* World Formal concept analysis 3.3. World Web min- ing 3.3.1.2. Site wrapping 3.3.1.3.1 Data extraction and integration 3.3.1.3.1 Web log analysis 3.3.1.4. Traffic analysis 3.3.1.6.* Informa- trieval 3.4.1. Document represen- trieval	3 2 1 9 3 *					
3.3. World Wide Web sis 3.3.1.3.1 Site wrapping Data extraction and integration Deep web 3.3.1.3.1 Web log analysis Surfacing Surfacing 3.3.1.3.1 Web log analysis Straffic analysis 3.3.1.6.* Informa- tion re- trieval Document represen- tation Web log analysis				Formal concept spale	market graph	
3.3. World Wide Web 3.3.1. Web min- ing Site wrapping Data extraction and integration Deep web Surfacing Surfacing 3.3.1.3.1 3.3.1.3.1 Web log analysis Traffic analysis 3.3.1.5. Deep web Surfacing Search results dedu- plication 3.3.1.4. Nowledge discovery Web log analysis Traffic analysis Traffic analysis 3.4.1. Informa- tion re- trieval Document represen- tation	0.2.1.12.					
3.3.1.Wide WebWeb min- ingSite wrapping Data extraction and integrationDeep web Surfacing Search results dedu- plication3.3.1.3.1Deep web Surfacing Search results dedu- plication3.3.1.3.3Deep web Surfacing Search results dedu- plication3.3.1.4Deep web Surfacing Search results dedu- plication3.3.1.6.*Document represen- tationDeep web Surfacing Search results dedu- plication	3.3.	World				
3.3.1. Web min- ing 3.3.1.2. Site wrapping Data extraction and integration 3.3.1.3.1 Deep web Surfacing Search results dedu- plication 3.3.1.3.2. Web log analysis 3.3.1.4. Web log analysis 3.3.1.5. Traffic analysis 3.3.1.6.* Document represen- trieval 3.4.1. Document represen- tation						
3.3.1.2. ing Site wrapping Data extraction and integration Deep web 3.3.1.3.1 Data extraction and integration Deep web 3.3.1.3.2. Web log analysis Straffic analysis 3.3.1.4. Sal.1.5. Traffic analysis 3.3.1.6.* Informa- tion re- trieval Document represen- tation	3.3.1.		Web min-			
3.3.1.2. Site wrapping Data extraction and integration Deep web Surfacing Sarfacing Search results dedu- plication 3.3.1.3.1 Web log analysis Traffic analysis A.1.5 Traffic analysis Knowledge discovery 3.4.1.6.* Document trieval						
3.3.1.3. 3.3.1.3.1 3.3.1.3.1 3.3.1.3.2. 3.3.1.3.2. 3.3.1.3.3. 3.3.1.3.3. Search results deduplication 3.3.1.3.3. Web log analysis 3.3.1.5. Traffic analysis 3.3.1.6.* Information retrieval 3.4.1. Document representation	3.3.1.2.		-	Site wrapping		
3.3.1.3.1 integration Deep web 3.3.1.3.2. Surfacing Surfacing 3.3.1.3.3. Web log analysis Surfacing 3.3.1.4. Traffic analysis Surfacing 3.3.1.5. Traffic analysis Knowledge discovery 3.4.1. Document representation Document representation						
3.3.1.3.1 Deep web 3.3.1.3.2 Surfacing 3.3.1.3.3 Search results deduplication 3.3.1.4. Web log analysis 3.3.1.5. Traffic analysis 3.3.1.6.* Knowledge discovery 3.4.1. Document representation						
3.3.1.3.2. Surfacing 3.3.1.3.3. Search results deduplication 3.3.1.3.3. Web log analysis 3.3.1.5. Traffic analysis 3.3.1.6.* Information retrieval 3.4.1. Document representation	3.3.1.3.1				Deep web	
3.3.1.3.3. Search results deduplication 3.3.1.4. Web log analysis 3.3.1.5. Traffic analysis 3.3.1.6.* Knowledge discovery 3.4.1. Document representation						
3.3.1.4. 3.3.1.5. 3.3.1.6.* 3.4. Informa- tion re- trieval Document a.4.1. Document represen- tation label{eq:construction} label{eq:construction} plication label{eq:construction} plication label{eq:construction} label{eq						
3.3.1.4. Web log analysis 3.3.1.5. Traffic analysis 3.3.1.6.* Traffic analysis 3.4. Information retrieval 3.4.1. Document representation						
3.3.1.5. Traffic analysis 3.3.1.6.* Information retrieval 3.4.1. Document representation	3.3.1.4.			Web log analysis		
3.3.1.6.* 3.4. Informa- tion re- trieval Document ation - tation - tati	3.3.1.5.					
3.4. Informa- tion re- trieval Document represen- tation						
3.4.1. tion re- trieval Document represen- tation tation		Informa-				
3.4.1. Document represen- tation		tion re-				
represen- tation		trieval				
tation	3.4.1.					
		1				
3.4.1.1. Document structure						
			tation			

3.4.1.2. Document topic models 3.4.1.3. Content analysis and feature selection 3.4.1.4. Data encoding and canonicalization 3.4.1.5. Document collection models 3.4.1.6. Oncologies 3.4.1.7. Dictionaries 3.4.1.8. Thesauri 3.4.2. Information retrieval 3.4.2.1. Users and interactive retrieval 3.4.2.4. Users and interactive retrieval 3.4.3.1. State constant c	
3.4.1.3. Content analysis and feature selection 3.4.1.4. Data encoding and canonicalization 3.4.1.5. Document collection models 3.4.1.6. Ontologies 3.4.1.8. Information retrieval query processing 3.4.2. Information retrieval state 3.4.2.1. Query intent Query log analysis Query suggestion Query suggestion Task models Search interfaces 3.4.3.1. Search interfaces 3.4.3.4. Retrieval models and rank- ing	
3.4.1.4.feature selection Data encoding and canonicalization Document collection models3.4.1.5.Informa- tion re- trieval query processing3.4.2.Informa- tion re- trieval query processing3.4.2.1. 3.4.2.2.Informa- tion re- trieval query processing3.4.2.3. 3.4.2.4.Informa- tion re- trieval query processing3.4.3.1. 3.4.3.1.Users and interactive retrieval3.4.3.1. 3.4.3.2.Users and interactive retrieval3.4.3.4.Retrieval models and rank- ing	
3.4.1.4.feature selection Data encoding and canonicalization Document collection models3.4.1.5.Informa- tion re- trieval query processing3.4.2.Informa- tion re- trieval query processing3.4.2.1. 3.4.2.2.Informa- tion re- trieval query processing3.4.2.3. 3.4.2.4.Informa- tion re- trieval query processing3.4.3.1. 3.4.3.1.Users and interactive retrieval3.4.3.1. 3.4.3.2.Users and interactive retrieval3.4.3.4.Retrieval models and rank- ing	
3.4.1.4. Data encoding and canonicalization 3.4.1.5. Document collection models 3.4.1.6. Onclogies 3.4.1.7. Dictionaries 3.4.1.8. Thesauri 3.4.2.1. Information retrieval query processing 3.4.2.1. Users and interactive retrieval 3.4.2.3. Users and interactive retrieval 3.4.3.1. Users and interactive retrieval 3.4.3.1. Retrieval models 3.4.3.4. Retrieval models 3.4.3.4. Retrieval models	
3.4.1.5.canonicalization Document collection models3.4.1.6.Dictionaries Thesauri3.4.1.7.Informa- tion re- trieval query processing3.4.2.Informa- tion re- trieval query suggestion Query suggestion Query reformulation3.4.2.1.Users and interactive retrieval3.4.2.5.Users and interactive retrieval3.4.3.1.Search interfaces Collaborative search3.4.3.4.3.Retrieval models and rank-	
3.4.1.5. Document collection models 3.4.1.6. Ontologies 3.4.1.7. Dictionaries 3.4.1.8. Information retrieval query processing Thesauri 3.4.2. Information retrieval query intent Query log analysis 3.4.2.4. Users and interactive retrieval query reformulation Personalization Task models 3.4.3.1. Retrieval models Personalization Task models 3.4.3.4. Retrieval models Collaborative search	
3.4.1.6. 3.4.1.7. 3.4.1.8. 3.4.2.models Ontologies Dictionaries Thesauri3.4.2.1. 3.4.2.1. 3.4.2.3. 3.4.2.4. 3.4.2.5. 3.4.3.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1. 3.4.4.1.<	
3.4.1.6. Ontologies 3.4.1.7. Information retrieval 3.4.2.1. Information retrieval 3.4.2.1. Query representation 3.4.2.2. Query representation 3.4.2.3. Query representation 3.4.2.4. Query log analysis 3.4.2.5. Query reformulation 3.4.3.1. Users and interactive retrieval 3.4.3.1. Retrieval 3.4.3.4. Retrieval models and ranking 3.4.3.4. Retrieval models and ranking	
3.4.1.7. J.4.1.8. 3.4.1.8. Information retrieval query processing 3.4.2.1. Query representation Query intent Query suggestion Query suggestion Query suggestion Query suggestion Query reformulation 3.4.2.4. Users and interactive retrieval retrieval models and rank-ing Mark ing Mark	
3.4.1.8. Information retrieval Thesauri 3.4.2. Information retrieval Query representation 3.4.2.1. Query representation Query intent 3.4.2.3. Query representation Query representation 3.4.2.4. Query suggestion Query reformulation 3.4.2.5. Users and interactive retrieval Personalization 3.4.3.1. Personalization Task models 3.4.3.4. Retrieval models Collaborative search	
3.4.2. Information retrival query processing 3.4.2.1. Query representation Query intent 3.4.2.2. Query intent 3.4.2.3. Query suggestion Query reformulation 3.4.2.4. Query suggestion Query reformulation 3.4.3.1. Users and interactive retrieval 3.4.3.2. Personalization Task models 3.4.3.4. Retrieval models and ranking	
3.4.2.1.tion re- trieval query processing3.4.2.1.Query representation Query intent Query log analysis Query suggestion Query reformulation3.4.2.3.Users and interactive retrieval3.4.3.1.Personalization Task models Sarch interfaces Collaborative search3.4.3.4.Retrieval models and rank- ing	
3.4.2.1. 3.4.2.2.trieval query processingQuery representation Query intent Query log analysis Query suggestion Query reformulation3.4.2.4. 3.4.2.5.Users and interactive retrievalPersonalization Task models Search interfaces Collaborative search3.4.3.1. 3.4.3.2.Retrieval models and rank- ingPersonalization Task models Collaborative search	
3.4.2.1. query processing Query representation 3.4.2.2. Query intent 3.4.2.3. Query guanalysis 3.4.2.4. Query suggestion 3.4.3.1. Users and interactive retrieval 3.4.3.1. Personalization 3.4.3.2. Search interfaces 3.4.3.4. Retrieval 3.4.3.4. Retrieval	
3.4.2.1. query processing Query representation 3.4.2.2. Query intent 3.4.2.3. Query guanalysis 3.4.2.4. Query suggestion 3.4.3.1. Users and interactive retrieval 3.4.3.1. Personalization 3.4.3.2. Search interfaces 3.4.3.4. Retrieval 3.4.3.4. Retrieval	
3.4.2.1. processing Query representation 3.4.2.2. Query intent Query log analysis 3.4.2.3. Query log analysis Query segestion 3.4.2.4. Query segestion Query reformulation 3.4.3.1. Users and interactive retrieval Personalization 3.4.3.1. Personalization Task models 3.4.3.4. Retrieval Collaborative search 3.4.3.4. Retrieval Collaborative search	
3.4.2.1. Query representation 3.4.2.2. Query intent 3.4.2.3. Query suggestion 3.4.2.4. Query suggestion 3.4.2.5. Query reformulation 3.4.3.1. Users and interactive retrieval 3.4.3.2. Personalization 3.4.3.3. Retrieval models 3.4.3.4. Retrieval models 3.4.3.4. Retrieval models	
3.4.2.2. Query intent 3.4.2.3. Query log analysis 3.4.2.4. Query suggestion 3.4.2.5. Query reformulation 3.4.3.1. Personalization 3.4.3.2. Sarch interfaces 3.4.3.4. Retrieval 3.4.3.4. Retrieval	
3.4.2.3. Query log analysis 3.4.2.4. Query suggestion 3.4.2.5. Users and interactive retrieval 3.4.3.1. Personalization 3.4.3.2. Task models 3.4.3.4. Retrieval 3.4.3.4. Retrieval A.4.4. Retrieval	
3.4.2.4. Query suggestion 3.4.2.5. Users and interactive retrieval 3.4.3.1. Personalization 3.4.3.2. Saradian and rank-ing 3.4.3.4. Retrieval 3.4.3.4. Retrieval	
3.4.2.5. Query reformulation 3.4.3. Users and interactive retrieval 3.4.3.1. Personalization 3.4.3.2. Task models 3.4.3.4. Retrieval 3.4.3.4. Retrieval	
3.4.3. Users and interactive retrieval 3.4.3.1. Personalization Task models 3.4.3.2. Search interfaces 3.4.3.4. Retrieval models and rank- ing	
3.4.3.1. interactive retrieval 3.4.3.2. Personalization 3.4.3.3. Search interfaces 3.4.3.4. Retrieval 3.4.3.4. Retrieval models and rank-ing	
3.4.3.1. Personalization 3.4.3.2. Task models 3.4.3.3. Search interfaces 3.4.3.4. Collaborative search 3.4.4. Retrieval models and rank- ing Instrumentary	
3.4.3.1. Personalization 3.4.3.2. Task models 3.4.3.3. Search interfaces 3.4.3.4. Retrieval 3.4.4. Retrieval models and rank- ing Image: Search interfaces	
3.4.3.2. Task models 3.4.3.3. Search interfaces 3.4.3.4. Collaborative search 3.4.4. Retrieval models and rank- ing Image: Search interfaces	
3.4.3.3. Search interfaces 3.4.3.4. Collaborative search 3.4.4. Retrieval models and rank- ing Interfaces	
3.4.3.3. Search interfaces 3.4.3.4. Collaborative search 3.4.4. Retrieval models and rank- ing Interfaces	
3.4.3.4. 3.4.4. Retrieval models and rank- ing Collaborative search	
3.4.4. Retrieval models and rank- ing	
models and rank- ing	
and rank- ing	
ing	
3.4.4.1. Rank aggregation	
3.4.4.2. Probabilistic retrieval	
models	
3.4.4.3. Language models	
3.4.4.4. Similarity measures	
3.4.4.5. Learning to rank	
3.4.4.6. Combination fusion	
and federated search	
3.4.4.7. Information retrieval	
diversity	
3.4.4.8. Top-k retrieval in	
databases	
3.4.4.9. Novelty in information	
retrieval	
3.4.5. Retrieval	
tasks and	
goals	
3.4.5.1. Question answering	
3.4.5.2. Document filtering	
3.4.5.3. Recommender systems	
3.4.5.4. Information extraction	
3.4.5.5. Sentiment analysis	
3.4.5.6. Sentiment analysis Expert search	
3.4.5.7. Near-duplicate and	
plagiarism detection	
3.4.5.8. Clustering and classifi-	
cation	
3.4.5.9. Summarization	
3.4.5.10. Business intelligence	
3.4.6. Evaluation	
of retrieval	
results	
3.4.6.1. Test collections	
3.4.6.2. Iest collections Relevance assessment	
3.4.6.3. Retrieval effectiveness	
3.4.6.4. Retrieval efficiency	
3.4.6.5. Presentation of re-	
trieval results	
3.4.7. Specialized	
infor-	
mation	
retrieval	

0.4.8.4	1	1				
3.4.7.1.				Structure and multilin- gual text search		
3.4.7.1.1.				gual text search	Structured text search	
3.4.7.1.2.					Mathematics re- trieval	
3.4.7.1.3.					Chemical and bio- chemical retrieval	
3.4.7.1.4.					Multilingual and cross-lingual re- trieval	
3.4.7.2.				Multimedia and multi- modal retrieval	trieval	
3.4.7.2.1. 3.4.7.2.2.					Image search Video search	
3.4.7.2.3. 3.4.7.2.4.					Speech / audio search Music retrieval	
3.4.7.3.				Environment-specific retrieval	Music Tetrieval	
3.4.7.3.1. 3.4.7.3.2. 3.4.7.3.3.					Enterprise search Desktop search Web and social me- dia search	
4.	Human- centered comput- ing					
4.1.	8	Visualiza- tion				
4.1.2.			Visualiza- tion tech- niques			
4.1.2.1.				Treemaps		
4.1.2.2. 4.1.2.3.				Hyperbolic trees Heat maps		
4.1.2.4.				Graph drawings		
4.1.2.5.				Dendrograms		
4.1.2.6. 4.1.2.7.*				Cladograms Elastic maps		
4.1.3.			Visualiza- tion ap- plication domains	-		
4.1.3.1.				Scientific visualization		
4.1.3.2. 4.1.3.3.				Visual analytics Geographic visualiza-		
4.1.3.4.				tion Information visualiza-		
4.1.4.			Visualiza-	tion		
			tion sys- tems and tools			
4.1.4.1.				Visualization toolkits		
4.1.5.			Visualiza-			
			tion the- ory con-			
			cepts and			
4.1.6.			paradigms Empirical			
			studies in			
			visualiza-			
4.1.7.			tion Visualiza-			
			tion de-			
			sign and evaluation			
			methods			
5.	Compu-					
	ting method-					
	ologies					
•						

5.1.	Artificial				
	intelli-				
	gence				
5.1.1.	gonee	Natural			
0.1.1.					
		language			
		processing			
5.1.1.2.			Information extraction		
5.1.1.3.			Machine translation		
5.1.1.4.			Discourse dialogue and		
0.1.1.1.			pragmatics		
5.1.1.5.			Natural language gen-		
			eration		
5.1.1.6.			Speech recognition		
5.1.1.7.			Lexical semantics		
5.1.1.7.1.*				Wikipedia based se-	
0.1.1.1.1.1				mantics	
5.1.1.8.				mantres	
5.1.1.8.			Phonology / morphol-		
			ogy		
5.1.1.9.			Language resources		
5.1.2.		Knowledge			
		represen-			
		tation and			
				1	
		reasoning			
5.1.2.1.			Description logics		
5.1.2.2.			Semantic networks	1	
5.1.2.3.			Nonmonotonic, default		
-			reasoning and belief re-		
			vision	1	
F 1 0 4					
5.1.2.4.			Probabilistic reasoning		
5.1.2.5.			Vagueness and fuzzy		
			logic		
5.1.2.6.			Causal reasoning and		
			diagnostics		
5.1.2.7.			Temporal reasoning		
5.1.2.8.			Cognitive robotics		
5.1.2.9.			Ontology engineering		
5.1.2.10.			Logic programming		
			and answer set pro-		
			gramming		
5.1.2.11.			Spatial and physical		
0.1.2.11.					
			reasoning		
5.1.2.12.			Reasoning about belief		
			and knowledge		
5.1.3.		Computer	-		
		vision			
5191		VISION	Computer vision and		
5.1.3.1.			Computer vision prob-		
			lems	-	
5.1.3.1.1.				Interest point and	
				salient region detec-	
	1			tions	
5.1.3.1.2.				Image segmentation	
5.1.3.1.3.				Video segmentation	
5.1.3.1.4.				Shape inference	
5.1.3.1.5.				Object detection	
5.1.3.1.6.				Object recognition	
5.1.3.1.7.				Object identification	
5.1.3.1.8.				Tracking	
5.1.3.1.9.				Reconstruction	
5.1.3.1.10.				Matching	
5.1.3.2.			Computer vision repre-	1	
			sentations		
5.1.3.2.1.	1			Image representa-	
				tions	
F 1 0 0 1 1 *				010115	an nga
5.1.3.2.1.1.*					2D PCA
5.1.3.2.2.				Shape representa-	
				tions	
5.1.3.2.3.				Appearance and tex-	
				ture representations	
F 4 9 9 1					
				Hierarchical repre-	
5.1.3.2.4.				sentations	
		1			
5.1.3.2.4. 5.2.	Machine				
5.2.	Machine learning	Learning			
		Learning			
5.2.		Learning paradigms	Supervised learning		

5.2.1.2. 5.2.1.1.4. 5.2.1.1.5. 5.2.1.1.5. 5.2.1.1.5. 5.2.1.2. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.4. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.1. 5.2.2.1.4. 5.2.2.1.4.3. 5.2.2.1.4.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.						
5.2.1.1.3. Supervised learning Supervised learning Supervised learning Supervised learning 5.2.1.1.6. Unsupervised learning Cluster analysis Anomaly detection 5.2.1.2. Supervised learning Cluster analysis Anomaly detection 5.2.1.2. Supervised learning Supervised learning Graph employee 5.2.1.2. Supervised learning Supervised learning Supervised learning 5.2.1.2. Supervised learning Supervised learning Supervised learning 5.2.1.2.7. Supervised learning Supervised learning Supervised learning 5.2.1.2.6. Supervised learning Supervised learning Supervised learning 5.2.1.3. Supervised learning Supervised learning Supervised learning 5.2.1.4.3. Supervised learning	5.2.1.1.1.				Ranking	
5.2.1.1.4 5.2.1.1.5 5.2.1.2.5 5.2.1.2.1 5.2.1.2.1 5.2.1.2.1 5.2.1.2.1 5.2.1.2.1 5.2.1.2.4 5.2.1.2.4 5.2.1.2.4 5.2.1.2.4 5.2.1.2.4 5.2.1.2.7 5.2.1.2.7 5.2.1.2.7 5.2.1.2.7 5.2.1.3.3 5.2.1.3.4 5.2.1.3.4 5.2.1.3.4 5.2.1.3.4 5.2.1.3.5 5.2.1.3.4 5.2.1.3.5 5.2.1.3.4 5.2.1.3.5 5.2.1.4.3 5.2.1.3.5 5.2.1.4.3 5.2.1.3.5 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.2.4 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.3.1 5.2.3.1 5.2.3.1 5.2.3.1 5.2.3.1 5.2.3.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.3.1 5.2.	5.2.1.1.2.				Learning to rank	
5.2.1.1.4 5.2.1.1.5 5.2.1.2.5 5.2.1.2.1 5.2.1.2.1 5.2.1.2.1 5.2.1.2.1 5.2.1.2.1 5.2.1.2.4 5.2.1.2.4 5.2.1.2.4 5.2.1.2.4 5.2.1.2.4 5.2.1.2.7 5.2.1.2.7 5.2.1.2.7 5.2.1.2.7 5.2.1.3.3 5.2.1.3.4 5.2.1.3.4 5.2.1.3.4 5.2.1.3.4 5.2.1.3.5 5.2.1.3.4 5.2.1.3.5 5.2.1.3.4 5.2.1.3.5 5.2.1.4.3 5.2.1.3.5 5.2.1.4.3 5.2.1.3.5 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.2.4 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.3.1 5.2.3.1 5.2.3.1 5.2.3.1 5.2.3.1 5.2.3.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.1.4.3 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.2.5 5.2.3.1 5.2.	5.2.1.1.3.				Supervised learning	
5.2.1.1.4. 5.2.1.1.6. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.4. 5.2.1.3. 5.2.1.4. 5.2.1.3. 5.2.1.4. 5.2.1.3. 5.2.1.4. 5.2.1.3. 5.2.1.4. 5.2.1.3. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.3.1. 5.						
5.2.1.1.6 5.2.1.1.6 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.3. 5.2.1.4. 5.2.1.4. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.3.1. 5.2.3.1.* 5.2.3.2.* 5.	59114					
5.2.1.1.5. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.5. 5.2.1.4. 5.2.1.4. 5.2.1.5. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.*	0.2.1.1.4.			1		
5.2.1.1.6. 5.2.1.2. 5.2.1.2. 5.2.1.2. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.3. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.6. 5.2.1.3.6. 5.2.1.3.7. 5.2.1.3.7. 5.2.1.3.8. 5.2.1.4.1.2. 5.2.1.4.1.2. 5.2.2.1.5. 5.2.2.8. 5.2.2.8. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.1.*						
5.2.1.2. 5.2.1.2.1. 5.2.1.2.3. 5.2.1.2.3. 5.2.1.2.4. 5.2.1.2.6.ing Cluster analysis Anomaly detection Mixture modeling Topic modeling Supervised learning Motif discovery Motif discovery <td>5.2.1.1.5.</td> <td></td> <td></td> <td></td> <td>Structured outputs</td> <td></td>	5.2.1.1.5.				Structured outputs	
5.2.1.2. 5.2.1.2. 5.2.1.2.3 5.2.1.2.4 5.2.1.2.5 5.2.1.2.5 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.3.5. 5.2.2.4. 5.2.2.4. 5.2.2.1.5. 5.2.2.3. 5.2.2.1.5. 5.2.2.3. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	5.2.1.1.6.				Cost-sensitive learn-	
5.2.1.2. 5.2.1.2. 5.2.1.2.3 5.2.1.2.4 5.2.1.2.5 5.2.1.2.5 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.3.5. 5.2.1.3.5. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.3.5. 5.2.2.4. 5.2.2.4. 5.2.2.1.5. 5.2.2.3. 5.2.2.1.5. 5.2.2.3. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*					ing	
5.2.1.2.1. 5.2.1.2.3. 5.2.1.2.4. 5.2.1.2.5. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.5. 5.2.1.4.1. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.1.5.5. 5.2.2.4.1.5.5.5.5.5.5.5.5.5.5.5.5.5.5.5.5.5.5	5919			Unaupowized learning		
5.2.1.2.2. 5.2.1.2.3. 5.2.1.2.4. 5.2.1.2.5. 5.2.1.2.6. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7.1.* 5.2.1.2.7.2.* 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.4.1. 5.2.1.4.2. 5.2.2.4. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.				Clisupervised learning		
5.2.1.2.3. 5.2.1.2.4. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.2. 5.2.1.3.2. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.2. 5.2.1.4. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.6. 5.2.2.7. 5.2.3.1.* 5.2.						
5.2.1.2.4. 5.2.1.2.6. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.2.7. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.1.4. 5.2.2.1.5. 5.2.2.1.5. 5.2.2.1.5. 5.2.2.1.5. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.3.* 5.2.2.1.* 5.2.3.						
5.2.1.2.5. Source separation Motif discovery Dimensionality 5.2.1.2.7. Source separation Motif discovery Dimensionality 5.2.1.2.7.1* Source separation Graph embedding 5.2.1.2.7.2* Reinforcement learning Sequential decision Inverse reinforcement learning 5.2.1.3.1 Nulti-task learning Inverse reinforcement learning Multi-agent reinforcement learning 5.2.1.3.3. S.2.1.4.5 S.2.1.4.5 Multi-task learning Multi-agent reinforcement learning 5.2.1.4.3 S.2.2.1.4 Source separation Multi-agent reinforcement learning 5.2.2.1.4.5 S.2.2.6 Learning Transfer learning 5.2.2.6 Learning from critiques Learning from critiques Learning from critiques 5.2.2.7. Source separation Semisupervised learning settings Semisupervised learning settings 5.2.2.7. Source separation Semisupervised learning settings Semisupervised learning settings 5.2.2.7. Source separation Kernel approach Splitting criteria 5.2.3.1.* Source separation Splitting criteria S.2.3.1.* Sourcement learning setting setings	5.2.1.2.3.				Mixture modeling	
5.2.1.2.6. 5.2.1.2.7.1.* 5.2.1.2.7.1.* 5.2.1.2.7.1.* 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.6. 5.2.2.7.5.* 5.2.3.1.* 5.2.3.2.*	5.2.1.2.4.				Topic modeling	
5.2.1.2.6. 5.2.1.2.7.1.* 5.2.1.2.7.1.* 5.2.1.2.7.1.* 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.4.5. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.6. 5.2.2.7.5.* 5.2.3.1.* 5.2.3.2.*	5.2.1.2.5.				Source separation	
5.2.1.2.7. 5.2.1.2.7.1.* 5.2.1.2.7.2.* 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.3. 5.2.1.4. 5.2.2.6. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*						
5.2.1.2.7.1.* 5.2.1.2.7.2.*Peduction and maniful fold learning fold learning fold learning fold learning fold learning fold learning making Inverse reinforce- making Inverse reinforce- making Inverse reinforce- making Inverse reinforce- making Inverse reinforce- making Apprenticeship learning Adversarial learning Adversarial learning Concement learning Adversarial learning Harting hulti-task learning Concement learning Adversarial learning Lifelong machine learning under co- variate shiftSequential decision making Inverse reinforce- making Apprenticeship learning Learning machine learning Learning from cri- tiques 5.2.2.3.Sequential decision making Inverse reinforce- making Apprenticeship learning Learning machine learning Learning from cri- tiques Sequential learning Sequential decision Multi-agent reinforce- maring Adversarial learning Learning from cri- tiques Sequential decision Multi-agent reinforce- maring adversarial learning Learning from cri- tiques Sequential decision Learning from cri- tiques Sequential decision Learning from cri- tiques Sequential decision strations Learning from cri- tiques Sequential decision settingsGraphemeta- tiques Sequential decision adversarial learning settings5.2.3.1. 5.2.3.1.* 5.2.3.1.*Machine learning ing apprentive settingsFree langementa- tion Splitting criteria Model treesPranele implementa- tion Splitting criteria Model trees5.2.3.1.* 5.2.3.2.* 5.2.3.2.*Sequential decision tiquesSequential decision tiquesSequential decision tiques5.2.3.1.* 5.2.3	5 2 1 2 7					
5.2.1.2.7.1.* Science Fold learning Graph embedding 5.2.1.2.7.2.* Fold learning Graph embedding 5.2.1.3. Sequential decision Sequential decision 5.2.1.3.1 Sequential decision Making 5.2.1.3.2 Sequential decision Making 5.2.1.3.3. Sequential decision Making 5.2.1.3.4. Sequential decision Making 5.2.1.3.5. Multi-task learning Transfer learning 5.2.1.4.1. Sci.1.4.1. Sci.1.4.1. 5.2.1.4.2. Sci.1.4.1. Sci.1.4.1. 5.2.2.2. Sci.1.4.1. Sci.1.4.1. 5.2.2.4. Sci.1.4.1. Sci.1.4.1. 5.2.2.5. Sci.1.4.1. Sci.1.4.1. 5.2.2.4. Sci.1.4.1. Sci.1.4.1. 5.2.2.5. Sci.1.4.1. Sci.1.4.1. 5.2.2.6. Sci.1.4.1. Sci.1.4.1. 5.2.2.7. Sci.1.4.1. Sci.1.4.1. 5.2.2.8. Sci.1.4.1. Sci.1.4.1. 5.2.2.7. Sci.1.4.1.	0.2.1.2.7.					
5.2.1.2.7.1.* 5.2.1.3. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.3. 5.2.2.4. 5.2.2.4. 5.2.2.4. 5.2.2.4. 5.2.2.5. 5.2.2.4. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*						
5.2.1.2.7.2.* bedding 5.2.1.3. Supervised 5.2.1.3.1. Sequential decision 5.2.1.3.1. Sequential decision 5.2.1.3.2. Supervised 5.2.1.3.3. Multi-task learning 5.2.1.4.3. Multi-task learning 5.2.2.4.1.3. Multi-task learning 5.2.2.4.1.4.2. Multi-task learning 5.2.2.3. Multi-task learning 5.2.2.4. Multi-task learning 5.2.2.4. Multi-task learning 5.2.2.4. Multi-task learning from critiques 5.2.2.5. Machine 5.2.2.6. Machine 5.2.3.1.* Machine 1.5.2.3.1.* Machine <td></td> <td></td> <td></td> <td></td> <td>fold learning</td> <td></td>					fold learning	
5.2.1.2.7.2.* bedding 5.2.1.3. Supervised 5.2.1.3.1. Sequential decision 5.2.1.3.1. Sequential decision 5.2.1.3.2. Supervised 5.2.1.3.3. Multi-task learning 5.2.1.4.3. Multi-task learning 5.2.2.4.1.3. Multi-task learning 5.2.2.4.1.4.2. Multi-task learning 5.2.2.3. Multi-task learning 5.2.2.4. Multi-task learning 5.2.2.4. Multi-task learning 5.2.2.4. Multi-task learning from critiques 5.2.2.5. Machine 5.2.2.6. Machine 5.2.3.1.* Machine 1.5.2.3.1.* Machine <td>5.2.1.2.7.1.*</td> <td></td> <td></td> <td></td> <td></td> <td>Graph em-</td>	5.2.1.2.7.1.*					Graph em-
5.2.1.2.7.2.* Supervised dimesional- ity reduction 5.2.1.3.1 Sequential decision making Inverse reinforce- ment learning 5.2.1.3.5. 5.2.1.3.1 Sequential decision making Inverse reinforce- ment learning 5.2.1.4. 5.2.1.3.4 Sequential decision making Inverse reinforce- ment learning Apprenticeship learning toreament learning Adversarial learning 5.2.1.4. 5.2.1.4.1 S.2.1.4.2 5.2.1.4.2 Batch learning Online learning set- tings 5.2.2.3 Learning from dring tiques 5.2.2.4. Learning from cri- tiques 5.2.2.5 Learning from implicit feedback Active learning set- tings 5.2.3.1.* Supervised S.2.3.1.* 5.2.3.1.* Classification and re- gression trees 5.2.3.1.* Splitting criteria Model trees 5.2.3.2.* Supervised S.2.3.2.* 5.2.3.1.* Supervised learning for dring approaches 5.2.3.1.* Kernel methods 5.2.3.2.* Supervised learning Supervised learning 5.2.3.2.* Supervised learning for dring approaches 5.2.3.2.* Supervised learning Supervised learning 5.2.3.1.* Supervised learning 5.2.3.2.* Supervised learning 5.2.3.2.* Supervise				1		
5.2.1.3. 5.2.1.3. 5.2.1.3.1. Sequential decision making 5.2.1.3.2. Sequential decision making 5.2.1.3.3. Multi-sasc 5.2.1.3.4. Multi-task learning 5.2.1.4. Transfer learning 5.2.1.4. Transfer learning 5.2.1.4. Contine learning 5.2.1.4. Multi-task learning 5.2.1.4. Contine learning 5.2.1.4. Multi-task learning 5.2.1.4. Machine 5.2.2. Multi-task learning from cri- 5.2.2.1. Machine 5.2.2.1. Machine 5.2.2.1. Machine 5.2.3.1. Machine 1.3. Machine 1.5. Machine	591979*			1	1	
5.2.1.3. 5.2.1.3.1. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.4. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.1.5. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.2.1.1.* 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.3.1.* <b< td=""><td>0.2.1.2.1.2.</td><td></td><td></td><td>1</td><td></td><td></td></b<>	0.2.1.2.1.2.			1		
5.2.1.3. 5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.2.1.5. 5.2.2.1.5. 5.2.2.3. 5.2.2.1.5. 5.2.2.3. 5.2.2.3. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2				1		
5.2.1.3.1. 5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.3. 5.2.1.4. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.5. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.4.3. 5.2.2.2. 5.2.2.2. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.				1	1	ity reduction
5.2.1.3.2, 5.2.1.3.3, 5.2.1.3.4, 5.2.1.3.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.2.1, 5.2.2.2, 5.2.2.2, 5.2.2.3, 5.2.2.3, 5.2.2.5, 5.2.2.6, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.3,1,.*, 5.2.3,1,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*,*, 5.2.3,2,*, 5.2.3,2,*,*, 5.2.3,2,*,*, 5.2.3,2,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*,*, 5.2.3,2,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,				Reinforcement learning	1	
5.2.1.3.2, 5.2.1.3.3, 5.2.1.3.4, 5.2.1.3.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.1.4, 5.2.2.1, 5.2.2.2, 5.2.2.2, 5.2.2.3, 5.2.2.3, 5.2.2.5, 5.2.2.6, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.2.7, 5.2.3,1,.*, 5.2.3,1,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*, 5.2.3,2,*,*, 5.2.3,2,*, 5.2.3,2,*,*, 5.2.3,2,*,*, 5.2.3,2,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*, 5.2.3,2,*,*,*,*, 5.2.3,2,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,	5.2.1.3.1.				Sequential decision	
5.2.1.3.2. 5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.4. 5.2.1.4. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.2. 5.2.2.3. 5.2.2.4. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.*				1		
5.2.1.3.3. 5.2.1.3.4. 5.2.1.3.5. 5.2.1.4. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.2.2. 5.2.2.1. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.4. 5.2.2.3. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*	52132			1		
5.2.1.3.3. 5.2.1.3.4. 5.2.1.4. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.3. 5.2.2.4.3. 5.2.2.2. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.4. 5.2.2.3. 5.2.2.3. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	0.2.1.0.2.			1		
5.2.1.3.4. 5.2.1.3.5. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.4. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	* * * * * *					
5.2.1.3.4. 5.2.1.3.5. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.2.4. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1.* 5.2.3.2.* 5.2.3.1.* 5	5.2.1.3.3.			1		
5.2.1.3.5. 5.2.1.4.1. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. Multi-task learning 5.2.1.4.2. Learning 5.2.1.4.2. Learning 5.2.1.4.2. Learning 5.2.1.4.3. Learning 5.2.1.4.2. Learning 5.2.1.4.3. Learning 5.2.1.4.2. Learning 5.2.2.1 Batch learning form critings 5.2.2.2. Learning from demonstrations 5.2.2.3. Learning from orriting settings 5.2.2.4. Settings 5.2.2.5. Learning from orriting settings 5.2.2.6. Settings 5.2.2.7.1.* Settings 5.2.2.7.1.* Settings 5.2.3.1. Machine learning from implicit feedback 6.2.3.1.* Classification and regression trees 5.2.3.1.* Splitting criteria Model trees 5.2.3.1.* Kernel methods 5.2.3.1.* Kernel methods 5.2.3.1.* Gaussian processes 5.2.3.1.* Kernel methods 5.2.3.2.* Gaussian processes 5.2.3.2.* <				1		
5.2.1.3.5. 5.2.1.4.1. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. Multi-task learning 5.2.1.4.2. Learning 5.2.1.4.2. Learning 5.2.1.4.2. Learning 5.2.1.4.3. Learning 5.2.1.4.2. Learning 5.2.1.4.3. Learning 5.2.1.4.2. Learning 5.2.2.1 Batch learning form critings 5.2.2.2. Learning from demonstrations 5.2.2.3. Learning from orriting settings 5.2.2.4. Settings 5.2.2.5. Learning from orriting settings 5.2.2.6. Settings 5.2.2.7.1.* Settings 5.2.2.7.1.* Settings 5.2.3.1. Machine learning from implicit feedback 6.2.3.1.* Classification and regression trees 5.2.3.1.* Splitting criteria Model trees 5.2.3.1.* Kernel methods 5.2.3.1.* Kernel methods 5.2.3.1.* Gaussian processes 5.2.3.1.* Kernel methods 5.2.3.2.* Gaussian processes 5.2.3.2.* <	5.2.1.3.4.	I		1	Multi-agent rein-	
5.21.3.5. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.4. 5.2.2.1. 5.2.2.3. 5.2.2.3. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.2.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*				1	forcement learning	
5.2.1.4. 5.2.1.4.1. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.2. 5.2.1.4.3. 5.2.2.4. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	5.2.1.3.5					
5.21.4.1. 5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.1. 5.2.2.2. 5.2.2.1. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.6. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.5.* 5.2.3.* 5.2.5.* 5.2.5.* 5.2.5.				Multi-task learning	rigital lear ming	
5.2.1.4.2. 5.2.1.4.3. 5.2.1.4.3. 5.2.2.2. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.6. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*				muni-task learning	Transford 1.	
5.2.1.4.3. 5.2.2. 5.2.2. 5.2.2. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.2.				1		
5.2.1.4.3. 5.2.2. 5.2.2. 5.2.2.1. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*	5.2.1.4.2.			1		
5.2.2. Learning settings Batch learning online learning settings Variate shift 5.2.2.1. S.2.2.3. Batch learning from demonstrations Variate shift 5.2.2.2. Settings Batch learning from demonstrations Variate shift 5.2.2.3. Settings Learning from demonstrations Variate shift 5.2.2.4. Learning from critiques Learning from implicit feedback Variate shift 5.2.2.5. Semi-supervised learning settings Semi-supervised learning settings Variate shift 5.2.2.7. Semi-supervised learning settings Semi-supervised learning settings Variate shift 5.2.3.1. Semi-supervised learning settings Variate shift Variate shift 5.2.3.1. Semi-supervised learning methestation Semi-supervised learning settings Variate shift 5.2.3.1. Semi-supervised learning methestation Variate shift Variate shift Variate shift 5.2.3.1. Semi-supervised learning methestation Variate shift Variate shift Variate shift 5.2.3.1.* Semi-supervised learning methestation Variate shift Variate shift Variate shift 5.2.3.2.1.* Seminet shift <				1		
5.2.2. Learning settings Batch learning online learning settings Variate shift 5.2.2.1. S.2.2.3. Batch learning from demonstrations Variate shift 5.2.2.2. Settings Batch learning from demonstrations Variate shift 5.2.2.3. Settings Learning from demonstrations Variate shift 5.2.2.4. Learning from critiques Learning from implicit feedback Variate shift 5.2.2.5. Semi-supervised learning settings Semi-supervised learning settings Variate shift 5.2.2.7. Semi-supervised learning settings Semi-supervised learning settings Variate shift 5.2.3.1. Semi-supervised learning settings Variate shift Variate shift 5.2.3.1. Semi-supervised learning methestation Semi-supervised learning settings Variate shift 5.2.3.1. Semi-supervised learning methestation Variate shift Variate shift Variate shift 5.2.3.1. Semi-supervised learning methestation Variate shift Variate shift Variate shift 5.2.3.1.* Semi-supervised learning methestation Variate shift Variate shift Variate shift 5.2.3.2.1.* Seminet shift <	5.2.1.4.3.			1	Learning under co-	
5.2.2. 5.2.2.1. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*				1		
5.2.2.1. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.*	522		Learning	1		
5.2.2.1. 5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.2. 5.2.3.	0.2.2.			1		
5.2.2.2. 5.2.2.3. 5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.6. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.2.*	5 0 0 1		settings			
5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1.2* 5.2.3.1.2* 5.2.3.1.1* 5.2.3.1.2* 5.2.3.2.5. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.2.3.* 5.						
5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	5.2.2.2.				1	
5.2.2.3. 5.2.2.4. 5.2.2.5. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*				tings		
5.2.2.4. strations Learning from critiques 5.2.2.5. Learning from implicit feedback Learning settings 5.2.2.6. Active learning settings Kernel approach 5.2.2.7. Semi-supervised learning settings Kernel approach 5.2.3.1. Classification and regression trees Parallel implementation 5.2.3.1.* Kernel methods Splitting criteria Model trees 5.2.3.1.* Kernel methods Splitting criteria Model trees 5.2.3.2. Gaussian processes Kernel Matrix Dynamic	5.2.2.3.				1	
5.2.2.4. 5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.1.* 5.2.3.2.*						
5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	5 2 2 4					
5.2.2.5. 5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.2. 5.2.3.2. 5.2.3.2.* 5.2.3.*	0.2.2.4.				1	
5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.3.1. 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*						
5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7. 5.2.2.7.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	5.2.2.5.				1	
5.2.2.6. 5.2.2.7. 5.2.2.7. 5.2.2.7.1.* 5.2.3.1. 5.2.3.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.2. 5.2.3.2. 5.2.3.2.* 5.2.3.2.1.* 5.2.3.2.* 5.2.3.*						
5.2.2.7. 5.2.2.7.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.* 5.2.3.* 5.2.3.* 5.2.3.2.* 5.2.3.2.* 5.2.3.2.* 5.2.3.* 5	5.2.2.6					
5.2.2.7. 5.2.2.7.1.* 5.2.3.1. 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*					1	
5.2.3.1. 5.2.3.1.* 5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.2. 5.2.3.2.1.* 5.2.3.2.* 5.2.3.2.1.* 5.2.3.2.* 5.2.3.2.1.* 5.2.3.2.* 5.2.3.2.1.*	5 2 2 7				1	
5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.1.* 5.2.3.2.*	0.4.4.1.				1	
5.2.3. Machine learn- ing ap- proaches Adachine learn- ing ap- proaches Parallel implementa- tion Splitting criteria Splitting c				ing settings		
5.2.3.1. ing approaches Classification and regression trees Parallel implementation 5.2.3.1.1* Classification and regression trees Parallel implementation 5.2.3.1.2* Splitting criteria Model trees 5.2.3.2.1* Kernel methods Kernel support vector machines 5.2.3.2.1.* Gaussian processes Kernel Matrix 5.2.3.2.3.* Kernel Independent				1	Kernel approach	
5.2.3.1. 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.2. 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.5.* 5.2.3.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.	5.2.3.		Machine	1		
5.2.3.1. 5.2.3.1.1.* 5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.2. 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.3.* 5.2.5.* 5.2.3.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.			learn-	1	1	
5.2.3.1. 5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.2.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.*				1		
5.2.3.1.* 5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.2.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.*				1		
5.2.3.1.1.* gression trees Parallel implementation 5.2.3.1.2.* Splitting criteria 5.2.3.2.3.* Kernel methods 5.2.3.2.1.* Kernel support vector machines 5.2.3.2.1.* Gaussian processes 5.2.3.2.3.* Kernel Matrix 5.2.3.2.4.* Kernel Matrix	5001		proaches			
5.2.3.1.1.* 5.2.3.1.2.* 5.2.3.1.2.* 5.2.3.2. 5.2.3.2. 5.2.3.2. 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.4.* Dynamic Dynamic Dynamic	5.2.3.1.				1	
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $				gression trees		
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	5.2.3.1.1.*			1	Parallel implementa-	
5.2.3.1.2.* 5.2.3.1.3.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.4.* 5.2.3.2.* 5.2.3.* 5.2.3.2.* 5.2.5.* 5.2.3.2.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.5				1		
5.2.3.1.3.* Model trees 5.2.3.2.1.* Kernel methods 5.2.3.2.1.* Dynamic 5.2.3.2.1.* Gaussian processes 5.2.3.2.3.* Kernel Matrix	52312*			1		
5.2.3.2. 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.2.* 5.2.3.2.3.* 5.2.3.2.3.* 5.2.3.2.4.* 5.2.3.4.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5.2.5.* 5				1		
5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.1.* 5.2.3.2.3.* 5.2.3.2.3.* 5.2.3.2.4.* Kernel Matrix Kernel Independent					model trees	
5.2.3.2.1.1.* 5.2.3.2.2. 5.2.3.2.2. 5.2.3.2.3.* Caussian processes Kernel Matrix S.2.3.2.4.*				Kernel methods		
5.2.3.2.1.1.* 5.2.3.2.2. 5.2.3.2.2. 5.2.3.2.3.* Caussian processes Kernel Matrix S.2.3.2.4.*				1	Kernel support vec-	
5.2.3.2.1.1.* 5.2.3.2.2. 5.2.3.2.3.* 5.2.3.2.3.* 5.2.3.2.3.* 5.2.3.2.4.* Dynamic Gaussian processes Kernel Matrix Kernel Independent	5.2.3.2.1.*			1		
5.2.3.2.2. 5.2.3.2.3.* 5.2.3.2.4.* Gaussian processes Kernel Independent	5.2.3.2.1.*			1		Dynamic
5.2.3.2.3.* Kernel Matrix 5.2.3.2.4.* Kernel Independent					Gaussian processos	
5.2.3.2.4.* Kernel Independent	5.2.3.2.1.1.*		1	1		
	5.2.3.2.1.1.* 5.2.3.2.2.					
components	5.2.3.2.1.1.* 5.2.3.2.2. 5.2.3.2.3.*					
	5.2.3.2.1.1.* 5.2.3.2.2. 5.2.3.2.3.*				Kernel Independent	

5.2.3.2.5.*	te	ernel-based clus- ering	
5.2.3.3.	Neural networks		
5.2.3.3.1.*		elf organized map	
5.2.3.3.2.*	Tr	raining approaches	Englishing
5.2.3.3.2.1.*			Evolutionary approach
5.2.3.3.3.*	Be	epresentation	approach
5.2.3.3.3.1.*	100	opresentation	Rule-based
			netwok
			$\operatorname{archirtecture}$
5.2.3.3.3.2.*			Fuzzy repre-
			sentation
5.2.3.3.4.*		volving NN	
5.2.3.3.5.* 5.2.3.4.	Logical and relational	nsembling	
0.2.0.1.	learning		
5.2.3.4.1.		ductive logic	
		arning	
5.2.3.4.2.		tatistical relational	
		arning	
5.2.3.5.	Learning in probabilis-		
5.2.3.5.1.	tic graphical models	aximum likelihood	
0.2.0.0.1.		odeling	
5.2.3.5.2.		aximum entropy	
	m	odeling	
5.2.3.5.3.		laximum a posteri-	
		i modeling	
5.2.3.5.4.		lixture models	
5.2.3.5.5.	La	atent variable mod-	
5.2.3.5.6.		s ayesian network	
0.2.0.0.0.		odels	
5.2.3.5.7.*		larkov network	
		odels	
5.2.3.6.	Learning linear models		
5.2.3.6.1.		erceptron algo-	
5.2.3.6.2.*		thm inear Discriminant	
0.2.0.0.2.		nalysis	
5.2.3.6.2.1.*			Tensor repre-
			sentation
5.2.3.7.	Factorization methods		
5.2.3.7.1.		on-negative matrix	
5.2.3.7.2.		ctorization actor analysis	
5.2.3.7.3.		rincipal component	
0.2.0.1.0.		nalysis	
5.2.3.7.3.1.*		-	2D PCA
5.2.3.7.3.2.*			Sparse PCA
5.2.3.7.4.		anonical correla-	
5.2.3.7.6.		on analysis	
0.2.0.1.0.		atent Dirichlet al- cation	
5.2.3.7.8.*		dependent Compo-	
		ent Analysis	
5.2.3.7.9.*	No	onlinear Principal	
		omponents	
5.2.3.7.10.*		ultidimentional	
5 9 9 7 10 1 *	sc	aling	Least moduli
5.2.3.7.10.1.* 5.2.3.8.	Rule learning		Least moduli
5.2.3.8.1.*		euro-fuzzy ap-	
		roach	
5.2.3.9.	Instance-based learn-		
	ing		
5.2.3.10.	Markov decision pro-		
5 9 9 11	cesses		
5.2.3.11.	Partially-observable Markov decision pro-		
	cesses		
5.2.3.12.	Stochastic games		
5.2.3.13.	Learning latent repre-		
	sentations		

5.2.3.13.1.			Deep belief networks
5.2.3.14.*		Multiresolution	*
5.2.3.15.*		Support vector ma-	
		chines	
5.2.4.	Machine		
	learning		
	algorithms		
5.2.4.1.		Dynamic programming	
		for Markov decision	
		processes	
5.2.4.1.1.		Î	Value iteration
5.2.4.1.2.			Q-learning
5.2.4.1.3.			Policy iteration
5.2.4.1.4.			Temporal difference
			learning
5.2.4.1.5.			Approximate dy-
			namic programming
			methods
5.2.4.2.		Ensemble methods	
5.2.4.2.1.			Boosting
5.2.4.2.2.			Bagging
5.2.4.2.3.*			Fusion of classifiers
5.2.4.3.		Spectral methods	
5.2.4.3.1.*			Spectral clustering
5.2.4.4.		Feature selection	
5.2.4.5.		Regularization	
5.2.4.5.1.*			Generalized eigen-
			value
5.2.5.	Cross-		
	validation		