

2009

## Knowledge libraries and information space

Eric Rayner  
*University of Wollongong*

Follow this and additional works at: <https://ro.uow.edu.au/theses>

### University of Wollongong

#### Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material.

Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

---

### Recommended Citation

Rayner, Eric, Knowledge libraries and information space, Doctor of Philosophy thesis, School of Computer Science and Software Engineering - Faculty of Informatics, University of Wollongong, 2009.  
<https://ro.uow.edu.au/theses/3027>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

## **NOTE**

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

## **UNIVERSITY OF WOLLONGONG**

### **COPYRIGHT WARNING**

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

# **KNOWLEDGE LIBRARIES AND INFORMATION SPACE**

A thesis submitted in partial fulfilment of the requirements for the award of  
the degree

## **DOCTOR OF PHILOSOPHY**

from the

## **UNIVERSITY OF WOLLONGONG**

by

**ERIC RAYNER, BCompSc(Hons)**

**SCHOOL OF COMPUTER SCIENCE AND  
SOFTWARE ENGINEERING**

**2009**



## Thesis Certification

I, Eric P. Rayner, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Information and Computer Science, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Eric Rayner  
July 27, 2009

# Contents

List of Figures	ix
List of Tables	xiii
Abbreviation, Notation and Typographical Conventions	xvii
Abstract	xix
Acknowledgements	xxi
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Overview . . . . .	1
1.2 The Contribution of this Thesis . . . . .	6
1.3 The Organisation of Information . . . . .	7
1.4 Thesis Methodology . . . . .	9
1.4.1 Gap in the literature . . . . .	10
1.4.2 Research hypothesis . . . . .	11
1.4.3 Experiment . . . . .	11
1.5 How to Read this Thesis . . . . .	12
1.6 Thesis Outline . . . . .	13
1.7 The Nature of Information . . . . .	14
1.8 Limitations of this Thesis . . . . .	17
1.9 Knowledge Libraries . . . . .	18
1.10 Information Space . . . . .	19
1.11 Information Organisation Terms . . . . .	23
<b>2 The Organisation of Information</b>	<b>25</b>
2.1 Overview . . . . .	25
2.2 Introduction . . . . .	26
2.3 Traditional Classification . . . . .	27
2.3.1 Faceted classification . . . . .	29
2.3.2 ISBN, ISSN, MARC and CIP . . . . .	30

2.3.3	A critique of traditional classification . . . . .	32
2.4	Computer File Systems . . . . .	34
2.4.1	A critique of computer file systems . . . . .	36
2.5	The Database . . . . .	36
2.5.1	Critique of the database . . . . .	38
2.6	Information Retrieval . . . . .	38
2.6.1	Measuring the effectiveness of IR . . . . .	40
2.6.2	A critique of IR . . . . .	41
2.7	Data Warehousing . . . . .	44
2.7.1	Critique . . . . .	48
2.8	The Internet, World Wide Web and Semantic Web . . . . .	48
2.8.1	The internet . . . . .	48
2.8.2	The world wide web . . . . .	49
2.8.3	The Semantic Web . . . . .	50
2.9	Discussion and Summary . . . . .	52
2.10	What this chapter achieved . . . . .	52
<b>3</b>	<b>Knowledge Libraries</b>	<b>53</b>
3.1	Overview . . . . .	53
3.2	Introduction . . . . .	54
3.3	Knowledge Library User Scenarios . . . . .	55
3.4	The Uses of Knowledge Libraries . . . . .	69
3.4.1	Knowledge Libraries for Research . . . . .	69
3.4.2	Knowledge Libraries for Education . . . . .	71
3.4.3	Knowledge Libraries for Business . . . . .	72
3.5	Core Knowledge Library Functionality . . . . .	72
3.5.1	Core knowledge library administration functionality . . . . .	73
3.5.2	Core knowledge library end use functionality . . . . .	73
3.6	Knowledge Library Security . . . . .	74
3.7	Extended Knowledge Library Functionality . . . . .	74
3.7.1	Automatic dimensions . . . . .	75
3.7.2	Dynamic dimensions . . . . .	75
3.7.3	Automatic report generation . . . . .	76
3.7.4	Automatic notification . . . . .	77
3.7.5	Knowledge library graphical user interface . . . . .	78
3.8	What this chapter achieved . . . . .	78
<b>4</b>	<b>Spaces for Information Organisation</b>	<b>79</b>
4.1	Overview . . . . .	79
4.2	Potential Mathematical Bases . . . . .	80

4.2.1	Metric Space . . . . .	81
4.2.2	Vector Space . . . . .	82
4.2.3	The vector model for information retrieval . . . . .	83
4.2.4	Lattices and Topological Space . . . . .	89
4.2.5	Formal Concept Analysis . . . . .	90
4.2.6	The Relational Model . . . . .	93
4.2.7	Online Analytical Processing (OLAP) . . . . .	95
4.3	Space . . . . .	96
4.4	$n$ -Dimensional Spaces . . . . .	97
4.5	Nested Spaces . . . . .	100
4.6	Span . . . . .	101
4.7	Spans of Points in $n$ -Dimensional Spaces . . . . .	104
4.8	The Generalised Triangle Inequality and Set Space . . . . .	105
4.9	Other Properties of Set Distance Functions . . . . .	106
4.9.1	$\subseteq$ -Reflexivity . . . . .	106
4.9.2	$\not\subseteq$ -Strict Positiveness . . . . .	107
4.9.3	$\not\subseteq^d$ -strict positiveness . . . . .	108
4.10	Signed Distances . . . . .	109
4.11	What this Chapter Achieved . . . . .	109
<b>5</b>	<b>Set Spaces</b>	<b>111</b>
5.1	Overview . . . . .	111
5.2	Manipulating Set Space . . . . .	112
5.2.1	$n$ -Dimensional Set Spaces . . . . .	112
5.2.2	Other Properties of $n$ -Dimensional Spaces . . . . .	113
5.2.3	Dimension Nesting . . . . .	114
5.2.4	Dilated and Translated Spaces . . . . .	115
5.3	Set Distance Functions Based on Set Operations . . . . .	116
5.4	The $d_{ij}^M$ Set Distance function . . . . .	118
5.4.1	The Triangle Inequality ( $\triangle I$ ) . . . . .	122
5.4.2	Span . . . . .	123
5.4.3	The Generalised Triangle Inequality ( $G\triangle I$ ) . . . . .	125
5.4.4	$\subseteq$ -reflexivity . . . . .	129
5.4.5	$\not\subseteq^d$ -strict positiveness . . . . .	129
5.5	What this Chapter Achieved . . . . .	130
<b>6</b>	<b><math>L</math>-Collections</b>	<b>133</b>
6.1	Overview . . . . .	133



6.2	Background . . . . .	134
6.2.1	Sets . . . . .	134
6.2.2	Multisets . . . . .	135
6.2.3	Merges and Joins . . . . .	136
6.2.4	Indexed Families . . . . .	137
6.2.5	Rough Sets . . . . .	138
6.2.6	Fuzzy sets . . . . .	139
6.3	$L$ -Collections . . . . .	142
6.3.1	$L$ -collection operators . . . . .	143
6.4	Sets, Multisets, Fuzzy Sets, Rough Sets and $L$ -Collections . . .	147
6.4.1	Sets and $L$ -Collections . . . . .	147
6.4.2	Multisets and $L$ -Collections . . . . .	148
6.4.3	Fuzzy Sets and $L$ -Collections . . . . .	148
6.4.4	Rough Sets and $L$ -Collections . . . . .	148
6.5	Extending Proofs Over Sets and Multisets to $\{1\}$ , $\mathbb{N}_1$ and $\mathbb{Q}^{>0}$ -Collections . . . . .	148
6.6	What this Chapter Achieved . . . . .	150
<b>7</b>	<b><math>L</math>-Collection Space</b>	<b>151</b>
7.1	Overview . . . . .	151
7.2	$L$ -Collection Distance Functions . . . . .	152
	An $L$ -Collection Distance Function . . . . .	152
7.2.1	$\Delta I$ for $ \mathcal{X}  -  \mathcal{X} \cap \mathcal{Y} $ . . . . .	153
7.2.2	$\mathcal{Q}^d$ -strict positiveness for $ \mathcal{X}  -  \mathcal{X} \cap \mathcal{Y} $ . . . . .	153
7.2.3	$\subseteq$ -reflexivity for $ \mathcal{X}  -  \mathcal{X} \cap \mathcal{Y} $ . . . . .	154
7.3	The $d_{ij}^{\mathcal{M}}$ $L$ -Collection Distance Function . . . . .	154
7.4	The ${}_k^{\mathcal{M}}d$ $L$ -Collection Distance Function . . . . .	156
7.4.1	Span and $G\Delta I$ for ${}_k^{\mathcal{M}}d$ . . . . .	159
7.4.2	$\subseteq$ -reflexivity for ${}_k^{\mathcal{M}}d$ distance functions . . . . .	162
7.4.3	$\mathcal{Q}^d$ -strict positiveness for ${}_k^{\mathcal{M}}d$ distance functions . . . . .	163
7.5	The ${}_{av}^{\mathcal{M}}d$ $L$ -Collection Distance Function . . . . .	164
7.5.1	$G\Delta I$ for ${}_{av}^{\mathcal{M}}d$ . . . . .	166
7.5.2	Other properties of ${}_{av}^{\mathcal{M}}d$ . . . . .	174
7.6	What this Chapter Achieved . . . . .	174
<b>8</b>	<b>Information Space</b>	<b>177</b>
8.1	Overview . . . . .	177
8.2	Introduction . . . . .	178
8.3	Networked Space . . . . .	179
8.4	Classification Space . . . . .	184

8.4.1	Classification spaces for uncertain and partial classifications . . . . .	185
8.4.2	Many levelled classification spaces . . . . .	187
8.4.3	Projected classification spaces . . . . .	188
8.5	Working with Classification Space . . . . .	190
8.5.1	Creation . . . . .	190
8.5.2	Addition and subtraction . . . . .	190
8.5.3	Point selection . . . . .	191
8.5.4	Ordering . . . . .	194
8.6	Attaching Information Units to Points in Classification Spaces . . . . .	194
8.6.1	Distance . . . . .	195
8.6.2	Indexing classification spaces . . . . .	196
8.7	Information Space . . . . .	198
8.8	Working with Information Space . . . . .	199
8.8.1	Creation . . . . .	199
8.8.2	Index Manipulation . . . . .	199
8.8.3	Information unit selection . . . . .	202
8.8.4	Selecting points in information space . . . . .	204
8.9	What this Chapter Achieved . . . . .	205
<b>9</b>	<b>Basing Knowledge Libraries on Information Space</b>	<b>207</b>
9.1	Overview . . . . .	207
9.2	Information Space for Questionnaire Knowledge Libraries . . . . .	208
9.2.1	Example Questionnaire Information Space . . . . .	210
9.3	Information Space for Research Paper Knowledge Libraries . . . . .	213
9.3.1	Selecting and Comparing Research Papers . . . . .	214
9.3.2	Extended Dimensions . . . . .	215
9.3.3	Further Dimensions . . . . .	217
9.4	What this Chapter Achieved . . . . .	218
<b>10</b>	<b>The Efficient Implementation of Knowledge Libraries</b>	<b>219</b>
10.1	Overview . . . . .	219
10.2	Introduction . . . . .	220
10.2.1	Distance query . . . . .	220
10.2.2	Range query . . . . .	221
10.2.3	$k$ Nearest neighbour query . . . . .	221
10.2.4	Ranked query . . . . .	221
10.2.5	Sequential search range query algorithm . . . . .	222

10.3	Metric Space Algorithms . . . . .	222
10.3.1	Relative ordering . . . . .	223
10.3.2	Radius partitioning . . . . .	224
10.3.3	Hyperplane partitioning . . . . .	226
10.3.4	Ranked query and $k$ -NN query algorithms . . . . .	227
10.3.5	A critique of the literature . . . . .	228
10.4	Adapting Metric Space Algorithms for Set Space . . . . .	229
10.4.1	Relative ordering for set spaces . . . . .	230
10.4.2	Radius partitioning for set space . . . . .	231
10.4.3	Hyperplane partitioning for set spaces . . . . .	233
10.5	Searching hard spaces . . . . .	234
10.5.1	Specialised algorithms for searching hard spaces . . . . .	235
10.5.2	Specialised algorithms for searching $n$ -dimensional spaces . . . . .	236
10.6	What this Chapter Achieved . . . . .	238
<b>11</b>	<b>Experimental Results and Discussion</b>	<b>241</b>
11.1	Overview . . . . .	241
11.2	Introduction . . . . .	242
11.3	Set Space Radius Partitioning Implementation: Non symmetric Experiments . . . . .	243
11.3.1	Non Symmetric Experiment setup . . . . .	244
11.3.2	Non Symmetric Experimental results . . . . .	247
11.3.3	Discussion of non symmetric results . . . . .	248
11.4	Variance Experiments . . . . .	251
11.4.1	Variance experimental results . . . . .	253
11.4.2	Random edge weight experimental results . . . . .	253
11.4.3	Euclidean space experimental results . . . . .	256
11.5	Center Selection and Multiple Tree Experiments . . . . .	257
11.5.1	Greatest minimum center selection experiment . . . . .	259
11.5.2	Standard deviation center selection experiment . . . . .	259
11.5.3	Discussion of center selection experiments . . . . .	261
11.5.4	Experiment with multiple search trees . . . . .	261
11.6	Experiments with Multi-Dimensional Spaces . . . . .	264
11.6.1	Multi-Dimensional experiment setup . . . . .	264
11.6.2	Multi-Dimensional experimental results and discussion . . . . .	265
11.6.3	Multiple tree experiments . . . . .	265
11.7	Set Space Experiments . . . . .	269
11.7.1	Discussion of set space results . . . . .	270
11.8	Sequential search algorithms . . . . .	270

11.8.1	Sequential search for $n$ -dimensional spaces . . . . .	272
11.8.2	Sequential search over set spaces with set distance func- tions . . . . .	272
11.8.3	Discussion of sequential search results . . . . .	273
11.9	Introducing the Sequential-Hybrid Algorithm . . . . .	274
11.9.1	Discussion of sequential-hybrid results . . . . .	274
11.10	Summary, discussion and recommendations . . . . .	275
11.11	What this Chapter Achieved . . . . .	276
<b>12</b>	<b>Summary, Discussion and Future Work</b>	<b>279</b>
12.1	Chapter Overview . . . . .	279
12.2	Thesis Summary . . . . .	280
12.2.1	The importance of information systems . . . . .	280
12.2.2	The significance of Knowledge Libraries . . . . .	281
12.2.3	The mathematical basis for Knowledge Libraries . . . .	282
12.2.4	Implementing Knowledge Libraries . . . . .	284
12.3	Discussion . . . . .	284
12.3.1	The Contribution of this Thesis . . . . .	285
12.3.2	The more formal development of Knowledge Libraries .	286
12.3.3	The flexibility of information space . . . . .	286
12.4	Future Work . . . . .	287
12.4.1	The implementation of Knowledge Libraries . . . . .	287
12.4.2	Graphical Interface . . . . .	287
12.4.3	Improving existing systems . . . . .	288
12.4.4	The dissemination of knowledge . . . . .	289
<b>Appendix A: A Guide to the Accompanying CD</b>		<b>291</b>
<b>Appendix B: Publications Relating to this Thesis</b>		<b>299</b>
<b>Appendix C: Glossary of Information Organisation Terms</b>		<b>301</b>

# List of Figures

2.1	A 13-digit ISBN with EAN-13 bar code . . . . .	31
4.1	A Hasse diagram of a concept lattice of objects = $\{1, \dots, 10\}$ and attributes = {composite, even, odd, prime, square}. . . .	92
4.2	Three balls $X, Z, Y$ in $\mathbb{R}^2$ . . . . .	102
5.1	Subset element counts for $\subseteq$ -reflexive proofs. $a =  X $ , $b =$ $ Y - X $ . . . . .	116
5.2	Subset element counts for $\Delta I$ proofs. $a =  Z - X - Y $ , $b =  Z \cap X - Y $ , $c =  Z \cap X \cap Y $ ... . . . .	117
5.3	Subset element counts for $\mathcal{Z}^d$ -strict positive proofs. $a =  X -$ $Y $ , $b =  X \cap Y $ and $c =  Y - X $ . . . . .	117
7.1	Sub $L$ -collection element counts for $\Delta I$ proofs. $a =  \mathcal{Z} - \mathcal{X} -$ $\mathcal{Y} $ , $b =  \mathcal{Z} \cap \mathcal{X} - \mathcal{Y} $ , $c =  \mathcal{Z} \cap \mathcal{X} \cap \mathcal{Y} $ , etc. . . . .	153
7.2	Sub $L$ -collection element counts for $\mathcal{Z}$ -strict positive proofs. $a =  \mathcal{X} - \mathcal{Y} $ , $b =  \mathcal{X} \cap \mathcal{Y} $ and $c =  \mathcal{Y} - \mathcal{X} $ . . . . .	153
7.3	Sub $L$ -collection element counts for $\subseteq$ -reflexive proofs. $a =$ $ \mathcal{X} $ , $b =  \mathcal{Y} - \mathcal{X} $ . . . . .	154
11.1	Frequency of distances for typical 1000-point network spaces with $maxDist = 25$ and 1500, 2000, 2500 and 3000 directed, $weight = 1$ edges. . . . .	245
11.2	Frequency of distances for typical 1000-point network spaces with $maxDist = 25$ . <b>LHS:</b> 1100 undirected, $weight = 1$ edges. <b>RHS:</b> 2200 directed, $weight = 1$ edges. . . . .	247

- 11.3 Data from a range query ( $x = 999$ ,  $t = 2$ ) on radius partitioning search trees ( $branchFactor = 2$ ,  $leafCapacity = 10$ ) over 1000 different, 1000-point, metric (network) spaces generated from networks with 1100  $weight = 1$  undirected edges and  $maxDist = 25$ . **LHS**: distribution of candidate point set size as a (truncated) percentage of space size. **RHS**: distribution of retrieved to candidate point set sizes (by truncated percentage). . . . . 251
- 11.4 Frequency of distances. **LHS**: a typical 1000-point network space with  $maxDist = 255$  and  $1.1n$  undirected edges with uniform random weights (integers 1 to 19) . Mean distance: 120.913, standard deviation: 39.2604. **RHS**: a 1-dimensional Euclidean space over integers 0–999. Mean distance: 333.333, standard deviation: 235.702. . . . . 255
- 11.5 Data from a range query ( $x = 999$ ,  $t = 50$ ) on radius partitioning search trees ( $branchFactor = 2$ ,  $leafCapacity = 10$ ) over 1000 different, 1000-point, metric (network) spaces (with 1100 randomly weighted undirected edges). **LHS**: distribution of candidate point set size as a (truncated) percentage of space size. **RHS**: distribution of retrieved to candidate point set sizes (by truncated percentage). Compare with figure 11.3. . . . . 255
- 11.6 Data from the range query  $x = 999$ ,  $t = 50$  on 1000 radius partitioning search trees (with randomly selected centers,  $branchFactor = 2$  and  $leafCapacity = 10$ ) over a 1000-point uniform Euclidean space. **LHS**: distribution of candidate point set size as a (truncated) percentage of space size. **RHS**: distribution of retrieved to candidate point set sizes (by truncated percentage). Compare with figure 11.3. . . . . 256

11.7	Data from a range query ( $x = 999$ , $t = 2$ ) on radius partitioning search trees ( $branchFactor = 2$ , $leafCapacity = 10$ ) with specially selected centers over 1000 different, 1000-point, metric (network) spaces (with 1100 $weight = 1$ undirected edges). <b>LHS</b> : distribution of candidate point set size as a (truncated) percentage of space size. <b>RHS</b> : distribution of retrieved to candidate point set sizes (by truncated percentage). Compare with figure 11.3. . . . .	260
11.8	Data for the range query $x = 999$ , $t = 2$ over 3 radius partitioning search trees (with random centers, $branchFactor = 2$ and $leafCapacity = 10$ ) for each of 1000 different, 1000-point, metric (network) spaces (with 1100 $weight = 1$ undirected edges). The intersection of the 3 resulting candidate point sets was taken as the candidate point set for this search method. <b>LHS</b> : distribution of candidate point set size as a (truncated) percentage of space size. <b>RHS</b> : distribution of retrieved to candidate point set sizes (by truncated percentage). . . . .	263
11.9	Distance frequencies for the “first” 1000 points in 2,3,4 and 5–dimensional uniform Euclidean spaces. Distances are truncated in the plot, but not when computing the mean and standard deviation. The 2–dimensional space has 32 coordinates each dimension. The others have 10, 6 and 4 (respectively). . . . .	266
11.10	Distribution of retrieved to candidate point set sizes (by truncated percentage) for the range query $x = 0$ , $t = 2$ over 1000 radius partitioning search trees (with random centers, $branchFactor = 2$ and $leafCapacity = 10$ ) for uniform 1000-point, 2,3,4 and 5–dimensional Euclidean space. . . . .	267

11.11	Distribution of retrieved to candidate set sizes (by truncated percentage) for the range query $x = 0$ , $t = 2$ over 1000 different groups of three radius partitioning search trees (with random centers, $branchFactor = 2$ and $leafCapacity = 10$ ) for a uniform 1000-point, 2,3,4 and 5-dimensional Euclidean space. The group candidate set is the intersection of the candidate sets corresponding to each of the three trees. . . .	268
11.12	Data from 1000 range queries ( $0 \leq x \leq 999$ , $t = 50$ ) on 1000 different radius partitioning search trees (with random centers, $branchFactor = 2$ , $leafCapacity = 10$ ) over the space $\langle \{0, \dots, 999\}, x - y \rangle$ . <b>LHS</b> : distribution of candidate point set size as a (truncated) percentage of space size. <b>RHS</b> : distribution of retrieved to candidate point set sizes (by truncated percentage). . . . .	269
11.13	Data from a range query ( $x = 999$ , $t = 2$ ) on radius partitioning search trees ( $branchFactor = 2$ , $leafCapacity = 10$ ) over 1000 different, 1000-point, (non symmetric) network spaces generated from networks with 2200 $weight = 1$ directed edges and $maxDist = 25$ . <b>LHS</b> : distribution of candidate point set size as a (truncated) percentage of space size. <b>RHS</b> : distribution of retrieved to candidate point set sizes (by truncated percentage). . . . .	270



# List of Tables

2.1	Simplified Star Schema for Nationwide Retail Chain . . . . .	46
3.1	Dimension types . . . . .	56
4.1	Eight relations illustrating operations defined in [24] . . . . .	94
5.1	Distances for three distance functions over $X = \{1, 3\}$ , $Y = \{5, 9\}$ and $Z = \{3, 5\}$ . . . . .	123
10.1	Ball to enclosing hypercube volume (4 s.f.) for different $n$ in Euclidian space . . . . .	237
11.1	Average, over 1000 distinct queries ( $t = 2$ , $0 \leq x < 1000$ ), radius partitioning tree search ( $branchFactor = 2$ , $leafCapacity = 10$ ) and sequential search range query times (milliseconds) for typical symmetric network spaces generated from (random) $n = 1000$ , $e = 1100$ ; $n = 10000$ , $e = 13000$ ; and $n = 100000$ , $e = 150000$ (undirected) networks and non symmetric network spaces generated from (random) $n = 1000$ , $e = 2200$ ; $n = 10000$ , $e = 26000$ ; and $n = 100000$ , $e = 300000$ (directed) networks. . . . .	248
11.2	Radius partitioning search tree ( $branchFactor = 2$ and $leafCapacity = 10$ ) for a typical, random, 1000-point network space (with 1100 undirected $weight = 1$ edges). . . . .	254

11.3	Candidate nodes, collisions, pruned nodes and points, considered and retrieved points by level from a range query ( $x = 999$ , $t = 2$ ) on the radius partitioning search tree in table 11.2. The retrieved point set contained 7 points, while the candidate point set contained 198 points, giving a retrieved to considered ratio of approximately 3%. . . . .	254
11.4	Typical radius partitioning search tree (with random centers, $branchFactor = 2$ and $leafCapacity = 1$ ) for a 1-dimensional Euclidean space over integers 0–999. Compare with table 11.2.	258
11.5	Collisions, pruned nodes and points, considered and retrieved points by level from the range query $x = 999$ , $t = 50$ on the radius partitioning search tree in table 11.4. Both retrieved and candidate point sets contained 51 points, giving a retrieved to considered ratio of 100%. Compare with table 11.3. . . . .	258
11.6	Candidate set sizes for the range query $x = 999$ , $t = 2$ over 3 radius partitioning search trees (with random centers, $branchFactor = 2$ and $leafCapacity = 10$ ) for each of 10 different, 1000-point, metric (network) spaces (with 1100 $weight = 1$ undirected edges). The size of the intersection of these 3 sets, and the size of the retrieved set is also displayed. . . . .	262
11.7	Space size and query time for the sequential search algorithm (for a C++ implementation, using the <code>cmath sqrt()</code> and <code>pow()</code> functions, on a 1.8GHz machine with 265k memory running Linux) for various $n$ -dimensional spaces with 1000 points in each dimension and $10^6$ information elements. “Dimensional distances” $d_1, \dots, d_n$ are combined using $\sqrt{(\sum_{i=1}^n d_i^2)}$ . . . . .	272

11.8	Space size and query time for the sequential search algorithm (for a C++ implementation, using the <code>cmath sqrt()</code> and <code>pow()</code> functions, on a 1.8GHz machine with 265k memory running Linux) for a 3-dimensional space. Each dimension is a set space, based on an underlying space with 1000 points. The algorithm determines distances for $10^6$ information units, attached to random points in the space. “Dimensional distances” $d_1, d_2, d_3$ are combined using $\sqrt{d_1^2 + d_2^2 + d_3^2}$ . . . . .	273
11.9	Retrieved to candidate set sizes (by truncated percentage) for a sequential search range queries over 3,5,7 and 9-dimensional spaces (with $10^6$ randomly attached information elements). Each dimension consists of 1000 points. Distances are uniform random integers between 1 and 1000. In each $n$ -dimensional space, the first $n - 1$ dimensions were used to determine the candidate set. . . . .	275

# Abbreviation, Notation and Typographical Conventions

The set of real numbers is denoted by  $\mathbb{R}$ , the set of rational numbers by  $\mathbb{Q}$  and the set of natural numbers (integers, strictly larger than 0) by  $\mathbb{N}_1$ . Note that  $\mathbb{R}^{\geq 0}$  is the set of real numbers greater than or equal to 0, while  $\mathbb{R}^{>0}$  is the set of real numbers strictly greater than 0. Interval notation is used to denote real intervals, so  $(0, 1]$  is the set of real numbers less than or equal to 1 and strictly larger than 0. More generally, capital letters, such as  $L, M, X, Y$ , are used to denote sets. The power set of any set  $M$  (the set of all subsets of  $M$ ) is denoted  $\mathcal{P}(M)$ .

Lowercase “math bold font” letters denote vectors, so  $\mathbf{x}$  and  $\mathbf{y}$  are vectors.

$L$ -collections (introduced in chapter 6) are distinguished from sets by using “math calligraphy font”, so  $\mathcal{M}, \mathcal{X}, \mathcal{Y}$  are  $L$ -collections.

Enclosing vertical bars are used to denote the cardinality of a set ( $|M|$ ), the cardinality of an  $L$ -collection ( $|\mathcal{M}|$ ), the absolute value of a real number ( $|d(x, y)|$ ) and the magnitude of a vector ( $|\mathbf{x}|$ ).

Bold text is used to denote key terms that are defined (or at least described), both within, and (optionally) prior to, the definition. “Double quotes” are used for short quotations (which are also referenced) and when introducing key terms that are not defined.

Finally, *iff* is used as shorthand for “if and only if”.

Abbreviations in this thesis are preceded and introduced by the corresponding, non abbreviated, full term.



# Abstract

This research describes and develops **Knowledge Libraries**, idealised systems for organising and presenting information. By providing a mathematical basis, the definition of **information space** establishes a formal foundation for Knowledge Libraries. The definition of information space builds on the new definitions of ***L*-collections**, which generalise sets by allowing a real valued grade to be associated with each element, and **set space**, which generalises metric space to better model the relationships between **information units**.

The **multiple search tree** method improves existing metric space range query algorithms. These algorithms are also generalised to work over set space. The **sequential-hybrid algorithm** enables efficient range queries over multi-dimensional spaces.



# Acknowledgements

First and foremost, I would like to thank Dr. Ian Piper, my principal supervisor, for his continuing support throughout my candidacy. Acknowledgement is also due Prof. Martin Bunder for his assistance with the mathematical work in this thesis. Without Prof. Bunder's input, this thesis would not have the emphasis on mathematical correctness that it does.

I would also like to thank my family for their support, particularly my father, Prof. John Rayner, for his willingness to proof read chapters and discuss thesis related matters (at some considerable length).