
INDEX

- Abnormalities, R-MAT generator detection, 65
- Application domains, entity resolution, 317–318
- Approximate substructure, variant substructure patterns, frequent substructures, 107–108
- Apriori-based graph mining (AGM) approach and algorithm, 184
 - chemical graphs, 364–365
 - frequent substructures discovery, 100, 101–103, 102
 - frequent topological patterns, 149
 - graph grammar learning, 184
 - rooted tree mining, 385
- Artificial data experiments, Subdue system/inductive logic programming (ILP) compared, 174–176
- Attribute similarity, graph-based entity resolution, 326–327
- Automorphism:
 - defined, 118
 - rooted tree mining, subtree generation, 385–386
- Autonomous systems, Internet analysis, 50–55
- Average cycle length (ACL), defined, 446
- Average path length (APL), defined, 446

- Bayesian kernel methods, vertex classification, 268–271. *See also* Kernel-based link analysis
- Beamwise graph-based induction (B-GBI), chunkingless graph-based induction decision tree (CL-GBI), 206–207
- Bias control, Laplacian kernels, regularized, 290–296
- Biochemistry, graph grammar learning, 194–195
- Bioinformatics applications (rooted tree mining), 405–408
 - phylogenetic trees, 407–408
 - RNA structure, 405–407
- Bipartite cores, community effects, R-MAT generator, 69, 70
- Bipartite graphs, R-MAT generator experiments, 86
- Blocking, clustering, graph-based entity resolution, 330
- Bootstrapping, clustering, graph-based entity resolution, 330–331
- Border routers, autonomous systems, Internet analysis, 53
- Brandes betweenness centrality (BR), defined, 445

- Candidate generation, FSG algorithm, frequent topological patterns, 128–129
- Canonical extension (rooted tree mining):
 - SLEUTH algorithm, 401
 - subtree generation, 388–389
- Canonical labeling:
 - defined, 118
 - frequent topological patterns, 119–121
- Cascade model, chemical graphs, 370–375
- CASE system, chemical graphs, 356–358
- Characteristic path length (CPL), defined, 447
- Chemical graphs, 347–379
 - CASE system, 356–358
 - combinations, 366–375
 - cascade model, 370–375
 - inductive logic programming (ILP), 369–370
 - MultiCASE, 367–369
 - commentary, 375–377
 - linear fragments, 362–366
 - mining issues, 355–356

- Chemical graphs, (*continued*)
 molecular representation, 347–355
 connection table, 350–351
 SMILES and Markush structures, 354–355
 stereochemistry, 352–353
 structure and structure diagrams, 347–350
 structure and substructure search, 353
 quantitative estimation, 358–362
- Child test, rooted tree mining, frequency
 computation, 395–396
- Chomsky hierarchy, graph grammar learning, 192
- Chunkingless graph-based induction (CL-GBI).
See also Decision tree chunkingless
 graph-based induction (DT-CLGBI)
 algorithm, 209–210
 approach, 208–209
 experimental evaluation, 211–214
 feature construction by, 215–216
 implementation issues, 210–211
 unsolved problem, 211
- Chunking problem, decision tree construction
 (CL-GBI), 207–208
- Circumference, defined, 451
- Class of grammars, graph grammar learning,
 192–193
- Clique, defined, 411
- Cliqueishness, defined, 448–449
- Closed frequent substructure, variant substructure
 patterns, frequent substructures, 107
- Closeness centrality, defined, 446
- CLUSMOL, chemical graphs, 362–363
- Clustering:
 defined, 3
 graph-based entity resolution, 330–332
- Clustering coefficient, community effects,
 R-MAT generator, 69, 70
- Coherent substructure, variant substructure
 patterns, frequent substructures, 108
- Collective entity resolution, graph-based, 320
- Color code, rooted tree mining, subtree
 generation, 386
- Community effects:
 dense subgraph extraction, 416
 R-MAT generator, 69–70
- Compaction, topology-shape-metrics graph
 visualization, 46
- Complex domains, graph grammar learning,
 194–197
- Complexity, Galois lattice, graph description,
 237, 239–240
- Compression, R-MAT generator detection, 66
- Computational biology, graph visualization
 applications, 38
- Concept lattice, formal concept analysis (FCA),
 231–235
- Connected graph, defined, 118
- Connection strength measurement, dense
 subgraph extraction, connection subgraphs,
 429–434
- Connection subgraphs:
 commentary, 438
 dense subgraph extraction, 429–438
 algorithms for, 434–435
 connection strength measurement, 429–434
 experiments, 437–438
 optimization heuristics, 435–437
- Connection table, molecular representation,
 350–351
- Constraints, graph matching, Terrorist Modus
 Operandi Detection System (TMODS), 458
- Contrast substructure, variant substructure
 patterns, frequent substructures, 108
- Core, defined, 236
- Counterterrorism, graph grammar learning,
 195–196. *See also* Terrorist Modus
 Operandi Detection System (TMODS)
- Cousin test (rooted tree mining):
 distinct occurrences counting, 398–399
 frequency computation, 394–395, 396–397
- CProgol, Subdue system/inductive logic
 programming (ILP) compared, 172
- Crossing reduction, hierarchical graph
 visualization methods, 43
- CSLOGS dataset experiments, rooted tree
 mining, 402
- Cycle-based graph kernels:
 empirical evaluation, 263–265
 kernel methods, 257–262
- Cycles removal, hierarchical graph visualization
 methods, 42–43
- Data mining:
 defined, 2
 field of, xiii, 2
 graph representation, 1–2
 graph visualization applications, 36–38
 terminology in, 2–3
- Decision tree chunkingless graph-based induction
 (DT-CLGBI), 203–226. *See also*
 Chunkingless graph-based induction
 (CL-GBI)
 chunkingless graph-based induction (CL-GBI),
 208–214
 algorithm, 209–210
 approach, 208–209
 experimental evaluation, 211–214

- implementation issues, 210–211
 - unsolved problem, 211
 - chunking problem, 207–208
 - classification, 217–218
 - experimental evaluation, 218–224
 - feature construction by CI-GBI, 215–216
 - graph-based induction (GBI), 205–207
 - beamwise graph-based induction (B-GBI), 206–207
 - principles, 205–206
 - graph-structured data, 214–215
 - overview, 203–205
 - working example, 216–217
- Degree, defined, 450
- Degree distribution:
- defined, 68
 - R-MAT generator, 90–91
- Dense subgraph:
- defined, 411
 - occurrence of, 411–412
- Dense subgraph extraction, 411–441
- connection subgraphs, 429–438
 - algorithms for, 434–435
 - connection strength measurement, 429–434
 - experiments, 437–438
 - optimization heuristics, 435–437
 - densest subgraph location, 416–418
 - graph shingling, 421–428
 - algorithm details, 425
 - example, 425–427
 - high-level description, 424–425
 - implementation, 427
 - shingling, 422–424
 - Web host graph, 427–428
 - massive graph algorithms, 412
 - overview, 411–414
 - preliminaries, 412–413
 - related work, 414–416
 - communities in social networks, 416
 - graph mining, 414
 - web-based community finding, 414–415
 - web graph searching, 415–416
 - trawling, 418–421
- Dense substructure, variant substructure patterns, frequent substructures, 109
- Density, defined, 450
- Descendant test (rooted tree mining):
- distinct occurrences counting, 398
 - frequency computation, 393–394
- Description lattice, formal concept analysis (FCA), 231–235
- DFS code, frequent substructures discovery, pattern growth approach, 106
- Diameter, defined, 450
- Diffusion kernels, relatedness measure, kernel-based link analysis, 296–297
- Directed graphs, R-MAT generator experiments, 85
- Discovery of frequent substructures. *See* Frequent substructures
- Discriminative frequent substructure, variant substructure patterns, 108–109
- Distinct occurrences counting, rooted tree mining, 397–399
- Distributed genetic search algorithm, Terrorist Modus Operandi Detection System (TMODS), 462
- DNA:
- graph grammar learning, 188
 - Mutagenesis dataset, 6–8
- Duplicate graph, frequent substructures discovery, pattern growth approach, 104
- Edge betweenness (stress), R-MAT generator, 71
- Edit path, defined, 22–23
- Effective diameter, defined, 69
- Efficiency, entity resolution, 315
- Efficient subgraph extension, vertical SiGraM algorithm, frequent topological patterns, 138–139
- Electrical networks, connection strength measurement, dense subgraph extraction, connection subgraphs, 431–432
- Embedded subtrees, rooted tree mining, frequency computation, 392–395
- Embedding identification, horizontal SiGraM algorithm, frequent topological patterns, 133–134
- Entity resolution, 311–344
- commentary, 341–342
 - graph-based, 318–341
 - author resolution, 321–322
 - clustering, 330–332
 - experimental evaluation, 333–341
 - generally, 318–319
 - issues in, 319–321
 - problem formulation, 322–325
 - similarity measures, 325–329
 - overview, 311–314
 - related work, 314–318
 - application domains, 317–318
 - efficiency, 315
 - evaluation matrices, 318
 - graph-based approaches, 315–316
 - probabilistic inference, 316–317
 - probabilistic modeling, 315
 - string similarity, 314

- Entity resolution, (*continued*)
 - theoretical bounds, 314
 - tools, 317
- Equivalence class-based extension (rooted tree mining):
 - SLEUTH algorithm, 401
 - subtree generation, 389–391
- Error-tolerant graph matching, graph matching methods, 22–24
- Evaluation matrices, entity resolution, 318
- Event detection, Terrorist Modus Operandi Detection System (TMODS), 463–464
- Exact graph matching, graph matching methods, 19–21
- Exponential family distributions, Bayesian kernel methods, vertex classification, 268–269
- Extension lattice, formal concept analysis (FCA), 231–235
- External-to-internal (E-to-I) ratio, defined, 449–450
- Extrapolation, R-MAT generator detection, 66

- FFSM, frequent topological patterns, 150
- First-order logic (FOL), inductive logic programming (ILP) systems, 184
- Flat domains, graph grammar learning, 193–194
- Forced-directed methods, graph visualization techniques, 39–41
- Formal concept analysis (FCA), 227–251
 - commentary, 249–250
 - concepts and notation, 228
 - defined, 229
 - extension lattice and description lattice give concept lattice, 231–235
 - Galois lattice, 230–231
 - graph description and Galois lattice, 235–240
 - complexity, 237, 239–240
 - homomorphism and product operator, 237
 - labeled graphs, 235–236
 - locally injective digraph, 238
 - graph mining, 240–249
 - experimentation, 247–249
 - generally, 240–241
 - irreducible elements, 243–245
 - irreducible elements and pattern mining, 246–247
 - language equivalence, 242–243
 - lattice properties, 241–242
 - overview, 227
- Frequency computation:
 - horizontal SiGraM algorithm, frequent topological patterns, 135
 - rooted tree mining, 392–397
 - embedded subtrees, 392–395
 - induced subtrees, 395–397
- Frequency counting:
 - FSG algorithm, frequent topological patterns, 129–131
 - horizontal SiGraM algorithm, frequent topological patterns, 133
 - vertical SiGraM algorithm, frequent topological patterns, 139
- Frequent graph, defined, 100–101
- Frequent subgraph (FSG), graph grammar learning, 184. *See also* FSG algorithm
- Frequent substructures, 99–115
 - apriori-based approach, 101–103
 - commentary, 112–113
 - concepts, 100–101
 - experiments, 109–112
 - overview, 99–100
 - pattern growth approach, 103–107
 - Subdue system and, 165–170
 - real dataset experiments, 168–170
 - synthetic dataset experiments, 167–168
 - variant substructure patterns, 107–109
 - approximate substructure, 107–108
 - closed frequent substructure, 107
 - coherent substructure, 108
 - contrast substructure, 108
 - dense substructure, 109
 - discriminative frequent substructure, 108–109
- Frequent topological patterns, 117–158
 - canonical labeling, 119–121
 - commentary, 151–152
 - datasets, 152–154
 - definitions, 118
 - FSG algorithm, 127–131
 - candidate generation, 128–129
 - frequency counting, 129–131
 - generally, 127–128
 - Grew algorithm, 141–148
 - graph representation, 143–144
 - meta-strategies, 146–148
 - multiedge collapsing, 146
 - properties of, 141–142
 - single-edge collapsing, 144–145
 - maximum independent set, 121
 - notations, 119
 - overview, 117–118
 - problem definitions, 122–127
 - algorithms for, 126–127
 - possibilities in, 122–126
 - related work, 149–151
 - SiGraM algorithms, 131–141
 - generally, 131–132
 - horizontal, 132–135

- interactions between, 139–141
- vertical, 135–139
- FSG algorithm. *See also* Frequent subgraph (FSG)
- frequent topological patterns, 127–131, 149, 150
 - candidate generation, 128–129
 - frequency counting, 129–131
 - generally, 127–128
 - rooted tree mining, 385
- Galois lattice:
 - extension lattice and description lattice give concept lattice, 231–235
 - formal concept analysis (FCA), 230–231
 - graph description, 235–240
 - complexity, 237, 239–240
 - homomorphism and product operator, 237
 - labeled graphs, 235–236
 - locally injective digraph, 238
 - properties of, graph mining, 241–242
- GASTON:
 - frequent topological patterns, 149
 - rooted tree mining, 385
- Gaussian processes, multiclass transduction, kernel methods, vertex classification, 271–272
- Generalized linear preference (PLP), R-MAT generator experiments, 84
- Genetic search algorithm, Terrorist Modus Operandi Detection System (TMODS), 461–462
- Geographical models, graph generators, 73
- Girth, defined, 450
- Global efficiency (GE), defined, 447
- Global resolution, local resolution and, graph-based entity resolution, 320
- Grammar learning. *See* Graph grammar learning
- Graph, defined, 2, 18–19
- Graph-based algorithms, Subdue system compared, 165
- Graph-based data mining. *See* Mining graph data
- Graph-based entity resolution, 318–341. *See also* entity resolution
- Graph-based induction (GBI):
 - chemical graphs, 363–364
 - decision tree construction (CL-GBI), 205–207
 - beamwise (B-GBI), 206–207
 - principles, 205–206
 - frequent topological patterns, 149
- Graph classification (kernel methods), 254–265
 - commentary, 279
 - cycle-based graph kernels, 257–262
 - empirical evaluation, 263–265
 - generally, 254–255
 - walk-based graph kernels, 255–257
- Graph databases, 3–10
 - Internet Movie Database (IMDb), 3–6
 - Mutagenesis dataset, 6–8
 - World Wide Web, 8–10
- Graph drawing. *See* Graph visualization
- Graph edit distance, defined, 23–24
- Graph generators:
 - generally, 71–72
 - geographical models, 73
 - optimization-based models, 78
 - preferential attachment models, 72–73
 - random graph models, 72
 - taxonomy of, summary table, 74–77
- Graph grammar learning, 183–201
 - class of grammars, 192–193
 - empirical evaluation, 193–198
 - commentary, 197–199
 - complex domains, 194–197
 - flat domains, 193–194
 - graph grammars, 185–186
 - heuristics, 186–192
 - generally, 186–187
 - recursion, 188–189
 - relationships, 191–192
 - rules, 187–188
 - variables, 190–191
 - overview, 183–184
 - research summary, 184–185
- Graph grammars, graph grammar learning, 185–186
- Graph isomorphism, defined, 19–20
- Graph matching, 17–34
 - commentary, 31–32
 - experimental evaluation, 28–31
 - learning edit costs, 24–28
 - learning probabilistic edit costs, 25–26
 - self-organizing edit costs, 27–28
 - methods, 18–24
 - error-tolerant graph matching, 22–24
 - exact graph matching, 19–21
 - overview, 17–18
 - Terrorist Modus Operandi Detection System (TMODS), 455–459
 - constraints, 458
 - hierarchical patterns, 457–458
 - multiple choices and abstractions, 456–457
 - ontologies, 458–459
 - Terrorist Modus Operandi Detection System (TMODS), inexact matching, 455–456
- Graph mining:
 - chemical graphs, 355–356
 - dense subgraph extraction, 414

- Graph mining: (*continued*)
 - formal concept analysis (FCA), 240–249
 - experimentation, 247–249
 - generally, 240–241
 - irreducible elements, 243–245
 - irreducible elements and pattern mining, 246–247
 - language equivalence, 242–243
 - lattice properties, 241–242
- Graph partitions, community effects, R-MAT generator, 69, 70
- Graph representation, data mining, 1–2
- Graph shingling (dense subgraph extraction), 421–428
 - algorithm details, 425
 - example, 425–427
 - high-level description, 424–425
 - implementation, 427
 - shingling, 422–424
 - Web host graph, 427–428
- Graph subisomorphism, defined, 20
- Graph transactions, graph grammar learning, 184
- Graph visualization, 35–63
 - commentary, 55–57
 - data mining applications, 36–38
 - defined, 3, 35
 - examples, 48–55
 - Internet analysis, 50–55
 - web searching, 48–50
 - overview, 35–36
 - techniques, 38–48
 - forced-directed methods, 39–41
 - generally, 38–39
 - hierarchical methods, 41–44
 - topology-shape-metrics method, 44–48
- Greedy algorithm (GMIS):
 - chemical graphs, 363–364
 - frequent topological patterns, maximum independent set, 121
- Grew algorithm, 141–148
 - graph representation, 143–144
 - meta-strategies, 146–148
 - multiedge collapsing, 146
 - properties of, 141–142
 - single-edge collapsing, 144–145
- Group detection, social network analysis (SNA), 452
- GSpan:
 - frequent substructures discovery, pattern growth approach, 104
 - frequent topological patterns, 149–150
- Hierarchical methods:
 - graph grammar learning, 192–193
 - graph matching, Terrorist Modus Operandi Detection System (TMODS), 457–458
 - graph visualization techniques, 41–44
- HITS:
 - evaluation, 304
 - kernel-based link analysis, 285–289
- Homeland security, graph visualization applications, 37. *See also* Terrorist Modus Operandi Detection System (TMODS)
- Homogeneity, defined, 449
- Homomorphism, product operator and, Galois lattice, graph description, 237
- Hop-plot, defined, 69
- Horizontal coordinate assignment, hierarchical graph visualization methods, 43–44
- Horizontal SiGraM algorithm:
 - frequent topological patterns, 132–135
 - problem formulations, 139–141
- Induced subgraph, defined, 118
- Induced subtrees, rooted tree mining, frequency computation, 395–397
- Inductive logic programming (ILP):
 - chemical graphs, 369–370
 - decision tree construction (CL-GBI), 204
 - graph grammar learning, 184, 194, 199
 - Subdue system compared, 170–179
 - artificial data experiments, 174–176
 - CProgol, 172
 - generally, 170–172
 - Mutagenesis dataset experiments, 172–174, 176–179
- Inexact matching, graph matching, Terrorist Modus Operandi Detection System (TMODS), 455–456
- Information systems, graph visualization applications, 37
- Integration, defined, 451
- Internet analysis, graph visualization example, 50–55. *See also* World Wide Web
- Internet computing, graph visualization applications, 36
- Internet Movie Database (IMDb), described, 3–6
- Irreducible elements:
 - graph mining, formal concept analysis (FCA), 243–245
 - pattern mining and, graph mining, formal concept analysis (FCA), 246–247
- Isomorphic graph, defined, 118
- Kandisky drawing convention,
 - topology-shape-metrics graph visualization, 48
- Katz Status Index, kernel-based link analysis, 300

- Kernel-based link analysis, 283–310
classification, 284–286
importance, 285–286
relatedness, 285
commentary, 308
evaluation, 299–308
approximation, 307–308
example, 300–304
Neumann kernels, 304
parameter sensitivity estimation, 307
regularized Laplacian and diffusion kernels, 305–306
overview, 283–284
practical issues, 297–299
computational issues, 298–299
parameter tuning, 297–298
relatedness measure, 290–297
diffusion kernels, 296–297
limitations, 290–291
regularized, 290–296
related work, 299–300
unified framework, 286–289
- Kernel methods, 253–282
commentary, 279–280
graph classification, 254–265
cycle-based graph kernels, 257–262
empirical evaluation, 263–265
generally, 254–255
walk-based graph kernels, 255–257
overview, 253–254
vertex classification, 266–278
Bayesian kernel methods, 268–271
empirical evaluation, 277–278
generally, 266–268
multiclass transduction, 271–276
string kernels, 276–277
- Knowledge discovery, defined, 3
- Labeled graph:
defined, 118
formal concept analysis, 228
- Language equivalence, graph mining, formal
concept analysis (FCA), 242–243
- Laplacian kernels:
evaluation, 305–306
relatedness measure, 290–297
limitations, 290–291
regularized, 290–296
- Lattice, defined, 228. *See also* specific lattices
- Lattice properties, formal concept analysis
(FCA), graph mining, 241–242
- Layer assignment, hierarchical graph
visualization methods, 43
- Learning edit costs (graph matching), 24–28
learning probabilistic edit costs, 25–26
self-organizing edit costs, 27–28
- Learning probabilistic edit costs, graph matching,
25–26
- Linear fragments, chemical graphs, 362–366
- Link analysis. *See* Kernel-based link analysis
- Locally injective digraph, Galois lattice, graph
description, 238
- Local resolution, global resolution and,
graph-based entity resolution, 320
- Low diameter, graph generators, 73
- Markush structures, SMILES and, molecular
representation, 354–355
- Maximum common subgraph (MCS), defined,
20–21
- Maximum common subgraph (MCS) distance,
defined, 21
- Maximum independent set, frequent topological
patterns, 121
- Memory requirements reduction, frequency
counting, FSG algorithm, 130–131
- Merging Matches algorithm, Terrorist Modus
Operandi Detection System (TMODS),
459–461
- Meta-strategies, Grew algorithm, 146–148
- Metrics, defined, 447–448
- Min-cut plots:
NetMine, 79–80, 86
R-MAT generator experiments, 83
- Minimum description length (MDL):
graph grammar learning, 188
rooted tree mining, 385
variant substructure patterns, discriminative
frequent substructure, 108
- Mining graph data, defined, 2
- Modified Laplacian kernels, evaluation, 306
- Molecular representation, 347–355
connection table, 350–351
SMILES and Markush structures, 354–355
stereochemistry, 352–353
structure and structure diagrams, 347–350
structure and substructure search, 353
- Molfea, chemical graphs, 364
- MultiCASE, chemical graphs, 367–369
- Multiclass transduction, kernel methods, vertex
classification, 271–276
- Multiedge collapsing, Grew algorithm, 146
- Multiple choices, graph matching, Terrorist
Modus Operandi Detection System
(TMODS), 456–457
- Mutagenesis dataset:
described, 6–8
Subdue system/inductive logic programming
(ILP) compared, 172–174, 176–179

- Multitype entity resolution, graph-based, 319–320
- NetMine system, 79–82
 - components of, 66
 - min-cut plots, 79–80
 - properties, 80–82
- Neumann kernels:
 - evaluation, 304
 - unified framework, 287–289
- Ontologies, graph matching, Terrorist Modus Operandi Detection System (TMODS), 458–459
- Optimization, multiclass transduction, kernel methods, vertex classification, 275–276
- Optimization-based models, graph generators, 78
- Optimization heuristics, dense subgraph extraction, connection subgraphs, 435–437
- Order, defined, 228
- Orthogonal drawing:
 - graph visualization, 36
 - topology-shape-metrics graph visualization, 44–46
- Orthogonalization, topology-shape-metrics graph visualization, 46
- Parameter sensitivity estimation, kernel-based link analysis, 307
- Parameter tuning, kernel-based link analysis, 297–298
- Parametric optimization problem, multiclass transduction, kernel methods, vertex classification, 272–273
- Parent identification generation, vertical SiGraM algorithm, 137
- Partial order, defined, 228
- Pattern growth approach, frequent substructures discovery, 103–107
- Pattern mining, irreducible elements and, graph mining, formal concept analysis (FCA), 246–247
- Peering session, autonomous systems, Internet analysis, 53
- Performance evaluation, rooted tree mining, 402–405
- Phylogenetic trees, rooted tree mining, bioinformatics applications, 407–408
- Planarization, topology-shape-metrics graph visualization, 46
- Polyline drawing, graph visualization, 36
- Power law(s), R-MAT generator, 68–69
- Power law degree distributions, graph generators, 72
- Preferential attachment models, graph generators, 72–73
- Preorder, defined, 228
- Probabilistic inference, entity resolution, 316–317
- Probabilistic modeling, entity resolution, 315
- Procedural method, graph generators, 73
- Product operator, homomorphism and, Galois lattice, graph description, 237
- Quantitative estimation, chemical graphs, 358–362
- Radiality, defined, 451
- Radius, defined, 451
- Random graph models, graph generators, 72
- Randomized scheme, Grew algorithm, 146–148
- Random walks, connection strength measurement, dense subgraph extraction, connection subgraphs, 430–431
- Readability, graph visualization, 36
- Real dataset experiments, Subdue system/frequent substructure approaches compared, 168–170
- Realism, R-MAT generator detection, 66
- Real-world graph comparisons, R-MAT generator experiments, 83–86
- Relatedness measure (kernel-based link analysis), 290–297
 - diffusion kernels, 296–297
 - limitations, 290–291
 - regularized, 290–296
- Resilience:
 - graph generators, 73
 - R-MAT generator, 71
- R-MAT generator, 65–95
 - commentary, 86–89
 - community effects, 69–70
 - degree distribution, 90–91
 - edge betweenness (stress), 71
 - experiments, 82–86
 - min-cut plots, 83
 - real-world graph comparisons, 83–86
 - graph generators, 71–78
 - generally, 71–72
 - geographical models, 73
 - optimization-based models, 78
 - preferential attachment models, 72–73
 - random graph models, 72
 - taxonomy of, summary table, 74–77
 - NetMine, 79–82
 - min-cut plots, 79–80
 - properties, 80–82

- overview, 65–67
- power laws, 68–69
- relational learning, 78
- resilience, 71
- singular vector value, 71
- small diameters, 69
- terminology, 65
- RNA structure, rooted tree mining,
 - bioinformatics applications, 405–407
- Rooted tree mining, 381–410
 - bioinformatics applications, 405–408
 - phylogenetic trees, 407–408
 - RNA structure, 405–407
 - commentary, 409
 - described, 382–384
 - distinct occurrences counting, 397–399
 - experimental results, 401–405
 - CSLOGS dataset experiments, 402
 - performance evaluation, 402–405
 - synthetic dataset experiments, 401–402
 - synthetic datasets, 401–402
 - frequency computation, 392–397
 - embedded subtrees, 392–395
 - induced subtrees, 395–397
 - overview, 381–382
 - related work, 384–385
 - SLEUTH algorithm, 399–401
 - subtree generation, 385–392
 - canonical extension, 388–389
 - commentary, 391–392
 - equivalence class-based extension, 389–391
- Scree plot, defined, 68
- Self-organizing edit costs, graph matching, 27–28
- Self-organizing map (SOM), graph matching, 27–28
- Shingling, graph shingling, dense subgraph extraction, 422–424
- SiGraM algorithm, 131–141
 - generally, 131–132
 - horizontal, 132–135
 - interactions between, 139–141
 - vertical, 135–139
- Similarity measures, graph-based entity resolution, 325–329
- Simulation studies, R-MAT generator detection, 66
- Single-edge collapsing, Grew algorithm, 144–145
- Singular vector value, R-MAT generator, 71
- SLEUTH algorithm, 382, 384, 391, 392, 397, 399–405, 409. *See also* Rooted tree mining
- Small diameters, R-MAT generator, 69
- SMILES, Markush structures and, molecular representation, 354–355
- Social network(s):
 - community effects, dense subgraph extraction, 416
 - connection strength measurement, dense subgraph extraction, connection subgraphs, 429–434
- Social network analysis (SNA), 443–468
 - Best Friends Group Detection algorithm, 452
 - commentary, 466–467
 - defined, 443–444
 - experiments, 465–466
 - group detection, 452
 - metrics, 444–451
 - average cycle length (ACL), 446
 - average path length (APL), 446
 - brandes betweenness centrality (BR), 445
 - characteristic path length (CPL), 447
 - circumference, 451
 - cliqueishness, 448–449
 - closeness centrality, 446
 - clustering coefficient (CC), 447–448
 - degree, 450
 - density, 450
 - diameter, 450
 - External-to-internal (E-to-I) ratio, 449–450
 - girth, 450
 - global efficiency (GE), 447
 - homogeneity, 449
 - radiality and integration, 451
 - radius, 451
 - overview, 443
 - problems in, 444
- Terrorist Modus Operandi Detection System (TMODS), 452–464
 - algorithms, 459–462
 - distributed genetic search algorithm, 462
 - event detection, 463–464
 - generally, 452–455
 - genetic search algorithm, 461
 - Merging Matches algorithm, 459–461
 - social network analysis capability, 462–463
 - standard graph matching extension, 455–459
 - constraints, 458
 - hierarchical patterns, 457–458
 - inexact matching, 455–456
 - multiple choices and abstractions, 456–457
 - ontologies, 458–459
- Social sciences, graph visualization applications, 36–37

- Software engineering, graph visualization applications, 37
- Stereochemistry, molecular representation, 352–353
- Straight-line drawing, graph visualization, 36
- Stress plot, R-MAT generator, 71
- String kernels, vertex classification, 276–277
- String similarity, entity resolution, 314
- Structure-activity relationship (SAR), Mutagenesis dataset, 6–8
- Subdue system, 159–181
 - approaches, 160–165
 - Subgraph clustering, 162
 - substructure discovery, 160–162
 - supervised learning, 162–165
 - chemical graphs, 363–364
 - commentary, 179
 - frequent substructure approaches compared, 165–170
 - real dataset experiments, 168–170
 - synthetic dataset experiments, 167–168
 - frequent topological patterns, 149, 150–151
 - graph-based algorithms compared, 165
 - graph grammar learning, 194, 197
 - inductive logic programming (ILP) compared, 170–179
 - artificial data experiments, 174–176
 - CProgol, 172
 - generally, 170–172
 - Mutagenesis dataset experiments, 172–174, 176–179
 - overview, 159–160
- Subgraph, defined, 19. *See also* Connection subgraphs
- Subgraph clustering, Subdue system, 162
- Subgraph extension, efficient, vertical SiGraM algorithm, frequent topological patterns, 138–139
- Subgraph isomorphism, defined, 118
- Substructure discovery, Subdue system, 160–162
- Subtree generation (rooted tree mining), 385–392
 - canonical extension, 388–389
 - commentary, 391–392
 - equivalence class-based extension, 389–391
- Supervised learning:
 - kernel methods, 253–254
 - Subdue system, 162–165
- SV-list joins, rooted tree mining, distinct occurrences counting, 398–399
- Synthetic dataset experiments:
 - rooted tree mining, 401–402
 - Subdue system/frequent substructure approaches compared, 167–168
- Terrorist Modus Operandi Detection System (TMODS), 452–464
 - algorithms, 459–462
 - distributed genetic search algorithm, 462
 - event detection, 463–464
 - generally, 452–455
 - genetic search algorithm, 461
 - Merging Matches algorithm, 459–461
 - social network analysis capability, 462–463
 - standard graph matching extension, 455–459
 - constraints, 458
 - hierarchical patterns, 457–458
 - inexact matching, 455–456
 - multiple choices and abstractions, 456–457
 - ontologies, 458–459
- Theoretical bounds, entity resolution, 314
- Topological frequent patterns. *See* Frequent topological patterns
- Topology-shape-metrics method, graph visualization techniques, 44–48
- Transaction identifier (TID) lists, frequency counting, FSG algorithm, 130–131
- Transductive inference, multiclass transduction, kernel methods, vertex classification, 273–275
- Trawling, dense subgraph extraction, 418–421
- Tree mining. *See* Rooted tree mining
- Undirected graphs, R-MAT generator experiments, 86
- Unsupervised learning, defined, 3
- Variant substructure patterns (frequent substructures), 107–109
 - approximate substructure, 107–108
 - closed frequent substructure, 107
 - coherent substructure, 108
 - contrast substructure, 108
 - dense substructure, 109
 - discriminative frequent substructure, 108–109
- Vertex classification:
 - commentary, 279–280
 - kernel methods, 266–278
 - Bayesian kernel methods, 268–271
 - generally, 266–268
 - multiclass transduction, 271–276
 - string kernels, 276–277
- Vertical SiGraM algorithm:
 - frequent topological patterns, 135–139
 - problem formulations, 139–141
- Voltage algorithms, connection strength measurement, dense subgraph extraction, 432

- Walk-based graph kernels:
 - empirical evaluation, 263–265
 - graph classification, 255–257
- Waxman model, graph generators, 73
- Web-based community finding, dense subgraph extraction, 414–415
- Web graph searching, dense subgraph extraction, 415–416
- Web searching:
 - graph visualization applications, 37
 - graph visualization example, 48–50
- World Wide Web, graph databases, 8–10. *See also* Dense subgraph extraction; Internet analysis
- XML data, rooted tree mining, 384

