Lecture 5: Document Summarization

Pilsung Kang

School of Industrial Management Engineering

Korea University

AGENDA

01 Word Cloud

02 Association Rules

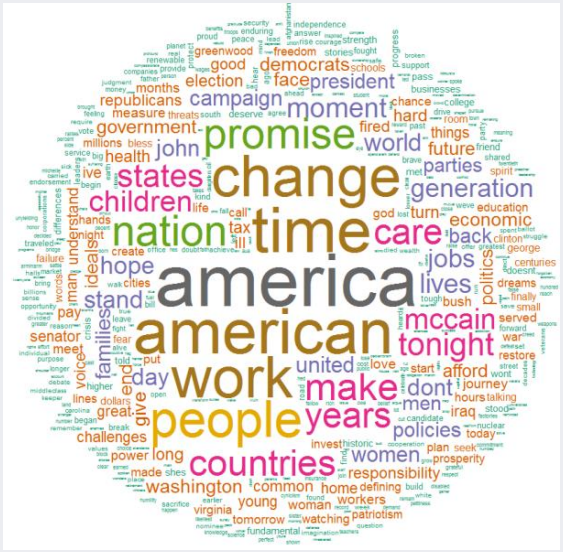
03 Network Representation

04 R Exercise

Background

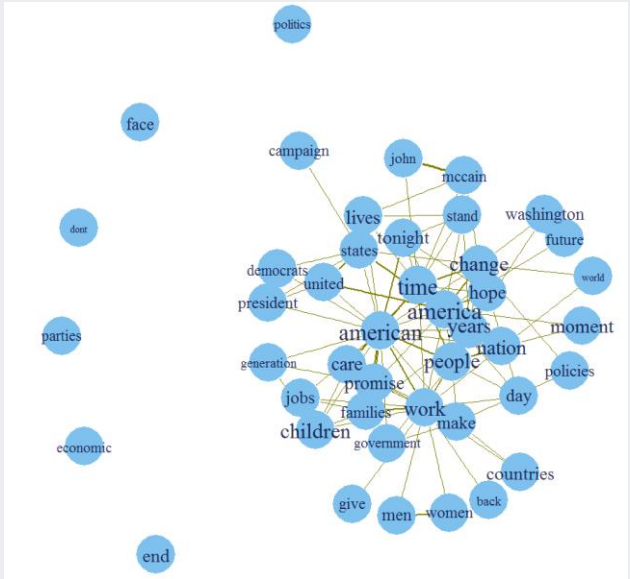
- How to visualize a large amount of text data at a glance?

Wordcloud



Association Rules

Keywords Network



	Frequency	Sentence/Phrase	Co-occurrence	Statistical significance
Wordcloud	O	X	Δ	X
Association Rules	O	O	O	O
Keyword Network	O	O	O	X

Wordcloud (Tagcloud)

- Wordcloud
  - ✓ A tool used to visually show the popularity of words in a collection of documents and how often they have been used
  - ✓ Conceptually resemble histograms, but can represent more items
- The way it works
  - ✓ The more a word is presented, the larger it will appear within the cloud
  - ✓ Words that are similar appear next to each other in the cloud
- Various algorithms/designs exist

http://www.edudemic.com/9-word-cloud-generators-that-arent-wordle/

Wordcloud

- Creation of wordcloud
  - ✓ The font size of a word in a wordcloud is determined by its incidence.
  - ✓ For smaller frequencies, one can specify font size directly, from one to whatever the maximum font size.
  - ✓ For larger values, a scaling should be made
  - ✓ An example of font size computation

$$font\ size\ (w(i)) = \frac{freq(w(i)) - min.freq(w, D)}{max.freq(w, D) - min.freq(w, D)} + constant$$

AGENDA

01 Word Cloud

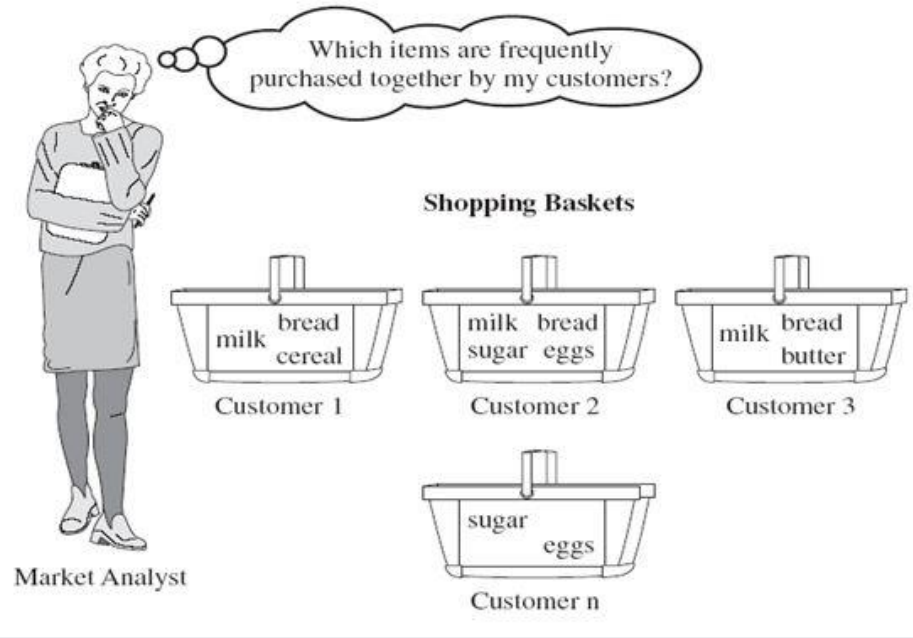
02 Association Rules

03 Network Representation

04 R Exercise

Association rules

- Goal:
  - ✓ Produce rules that define "what goes with what"
  - ✓ "If X was purchased, then Y was also purchased" (in Market Basket Analysis; MBA)
  - ✓ "If a word X is presented in a sentence (phrase), then a word Y is also presented in the same sentence (phrase)" (in Text Mining; TM)




```
> inspect(ares.pruned)
    lhs                rhs          support confidence      lift
1   {기적}          => {한강}    0.04494382  1.0000000 22.250000
2   {기술}          => {과학}    0.04494382  1.0000000 17.800000
3   {경제,
     기술}          => {창조}    0.03370787  1.0000000 11.125000
4   {경제,
     과학}          => {창조}    0.04494382  1.0000000 11.125000
5   {국민,
     신뢰}          => {정부}    0.03370787  1.0000000 11.125000
6   {융성}          => {문화}    0.03370787  1.0000000  8.900000
7   {과학}          => {창조}    0.04494382  0.8000000  8.900000
8   {민국}          => {대한}    0.13483146  1.0000000  7.416667
9   {창조}          => {경제}    0.08988764  1.0000000  6.846154
10  {오늘}          => {민국}    0.04494382  0.8000000  5.933333
11  {오늘}          => {대한}    0.04494382  0.8000000  5.933333
12  {과학}          => {경제}    0.04494382  0.8000000  5.476923
13  {존경}          => {여러분}  0.04494382  0.8000000  5.085714
14  {경제부흥,
     국민}          => {행복}    0.03370787  1.0000000  4.944444
15  {여러분,
     희망}          => {시대}    0.03370787  1.0000000  4.944444
```

Association rules

- Features
  - ✓ Rows are transactions (in MBA) or sentences/phrase (in TM)
  - ✓ A Synthetic Example
    - 6 keywords in 10 sentences

Sentence	Word 1	Word 2	Word 3	Word 4
S1	Love	Movie	Football	
S2	Movie	Watch		
S3	Movie	Sleep		
S4	Love	Movie	Watch	
S5	Love	Sleep		
S6	Movie	Sleep		
S7	Movie	Watch		
S8	Love	Movie	Sleep	Football
S9	Love	Movie	Sleep	
S10	Party			

Association rules

- Terminology
  - ✓ Antecedent – "IF" part
  - ✓ Consequent – "THEN" part
  - ✓ Item set – the items comprising the antecedent or consequent
  - ✓ Antecedent and consequent are disjoint (have no items in common)

- Generating rules
  - ✓ Many rules are possible (e.g., for sentence 1)
    - If Love is presented, then Movie is also presented.
    - If Love and Movie are presented, then Football is also presented.
    - If Football is presented, then Love is also presented.
    - etc.

Association rules: Performance measures

For the rule A → B

- Support

$$Support\ (A) = P(A)\ or\ Support\ (A \rightarrow B) = P(A, B)$$

  ✓ Used to find the frequent item sets

- Confidence

$$Confidence\ (A \rightarrow B) = \frac{P(A, B)}{P(A)}$$

  ✓ Used to generate meaningful rules

- Lift

$$Lift\ (A \rightarrow B) = \frac{P(A, B)}{P(A) \times P(B)}$$

  ✓ Used to evaluate the statistical significance of the generated rules

    - If lift = 1, then the antecedent and the consequents are statistically independent
    - If lift > 1, then the rule is useful in finding consequent item sets

Association rules

- How to generate effective association rules?
  - ✓ Ideally, create all possible combinations of items and see what rules are effective and what rules are not.
  - ✓ Computation time grows exponentially as the number of items increases.

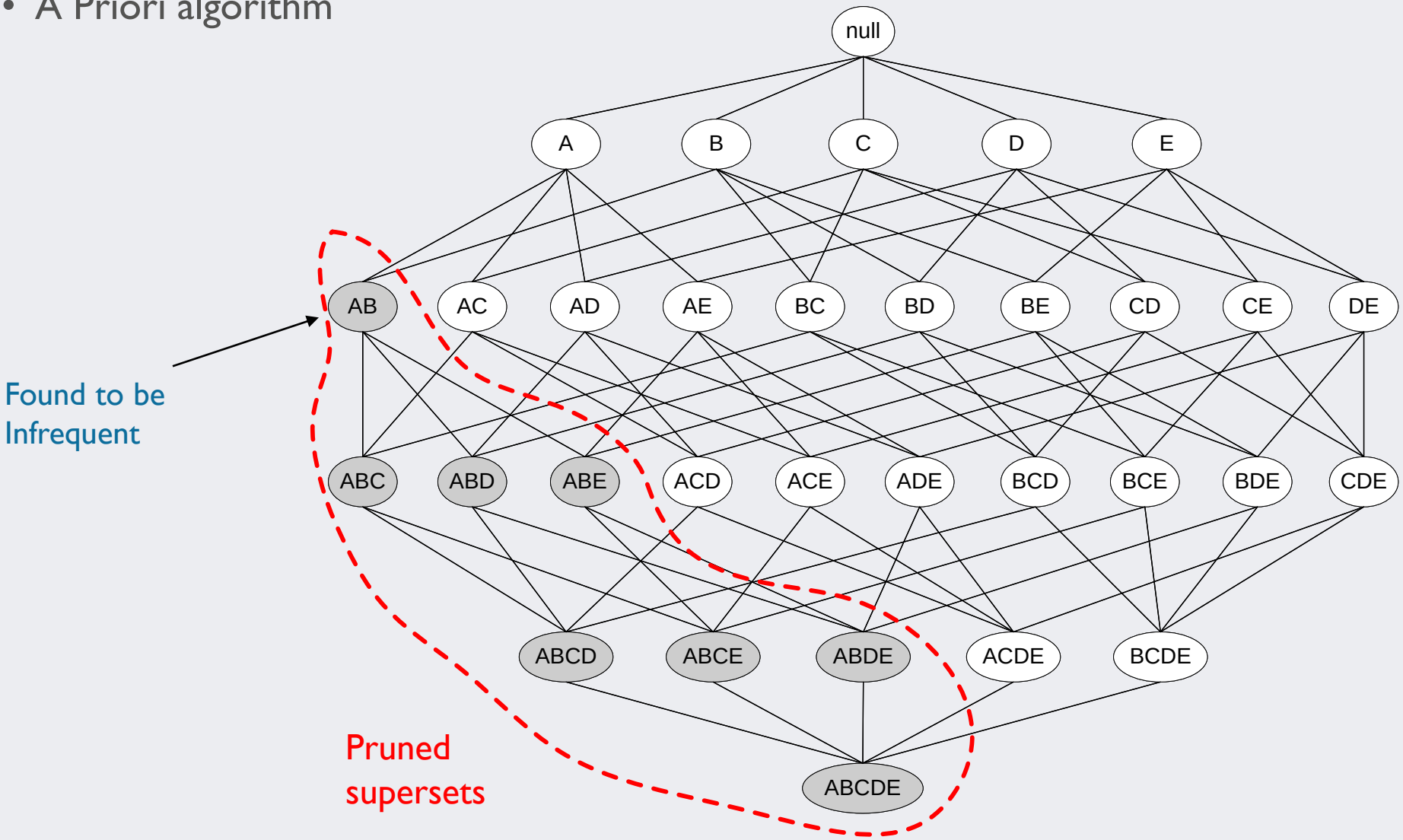
- Brute-force approach
  - ✓ List all possible association rules
  - ✓ Compute the support and confidence for each rule
  - ✓ Prune rules that fail the minimum support and minimum confidence threshold
  - ✓ Computationally prohibitive!

Association rules

- A priori algorithm
  - ✓ Consider only "frequent item sets"
  - ✓ Support
    - Criterion for item set frequency P(A)
    - #(%) of sentences that include the antecedent (both the antecedent and the consequent)
    - Support for the item set {Love, Movie} is 4 out of 10 sentences, or 40%
  - ✓ Support of an itemset never exceeds the support of its subsets, which is known as anti-monotone property of support.

Association rules

- A Priori algorithm



Association Rules: Generating Frequent Item Sets

I

- Set a minimum support criterion

- Set the minimum support to 2 sentences or 20%

Sentence	Word 1	Word 2	Word 3	Word 4
S1	Love	Movie	Football	
S2	Movie	Watch		
S3	Movie	Sleep		
S4	Love	Movie	Watch	
S5	Love	Sleep		
S6	Movie	Sleep		
S7	Movie	Watch		
S8	Love	Movie	Sleep	Football
S9	Love	Movie	Sleep	
S10	Party			

Association Rules: Generating Frequent Item Sets

2

- Generate the list of one-item sets that meets the support criterion

- Support {Movie} = 8/10 = 80%
- Support {Love} = 5/10 = 50%
- Support {Sleep} = 5/10 = 50%
- Support {Watch} = 3/10 = 30%
- Support {Football} = 2/10 = 20%
- Support {Party} = 1/10 = 10%

Party is removed because it does not meet the minimum support criterion

Association Rules: Generating Frequent Item Sets

3

- Use the life of one-item sets to generate list of two-item sets that meet the support criterion

	Movie	Love	Sleep	Watch	Football
Movie		**40%**	**40%**	**20%**	**20%**
Love			**30%**	0%	**20%**
Sleep				0%	10%
Watch					0%
Football					

- Among the 10 possible item sets, six of them are still found to be frequent

Association Rules: Generating Frequent Item Sets

- Use the list of two-item sets to generate the three-item sets.
- Continue up through k-item sets.

Set-size	Word 1	Word 2	Word 3	..	Word 6
1	Movie				
1	Love				
1	Sleep				
1	Watch				
1	Football				
2	Movie	Love			
2	Movie	Sleep			
2	Movie	Watch			
…	…	…			

4

Association Rules

- A priori algorithm
  - ✓ Let k=1
  - ✓ Generate frequent itemsets of length 1
  - ✓ Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Prune candidate itemsets containing subsets of length k that are infrequent
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

Association Rules: Result

- Generated Rules

Rule: If all Antecedent items are purchased, then with Confidence percentage Consequent items will also be purchased.

Row ID	Confidence %	Antecedent (A)	Consequent (C)	Support for A	Support for C	Support for A & C	Lift Ratio
6	100	Football	Love & Movie	2	4	2	2.5
2	100	Football	Love	2	5	2	2
4	100	Movie & Football	Love	2	5	2	2
3	100	Football	Movie	2	8	2	1.25
5	100	Love & Football	Movie	2	8	2	1.25
7	100	Watch	Movie	3	8	3	1.25
1	80	Love	Movie	5	8	4	1
8	80	Sleep	Movie	5	8	4	1

✓ Interpretation

- Support for A & C = 2 → There are two sentences that the words Football, Love, and Movie are presented together.
- 100% Confidence → If Football is presented in a sentence, then it is always that Love and Movie are also presented.
- Lift Ratio 2.5 → The association between the two item sets are 2.5 stronger than when they are assumed to be statistically independent.

AGENDA

01 Word Cloud

02 Association Rules

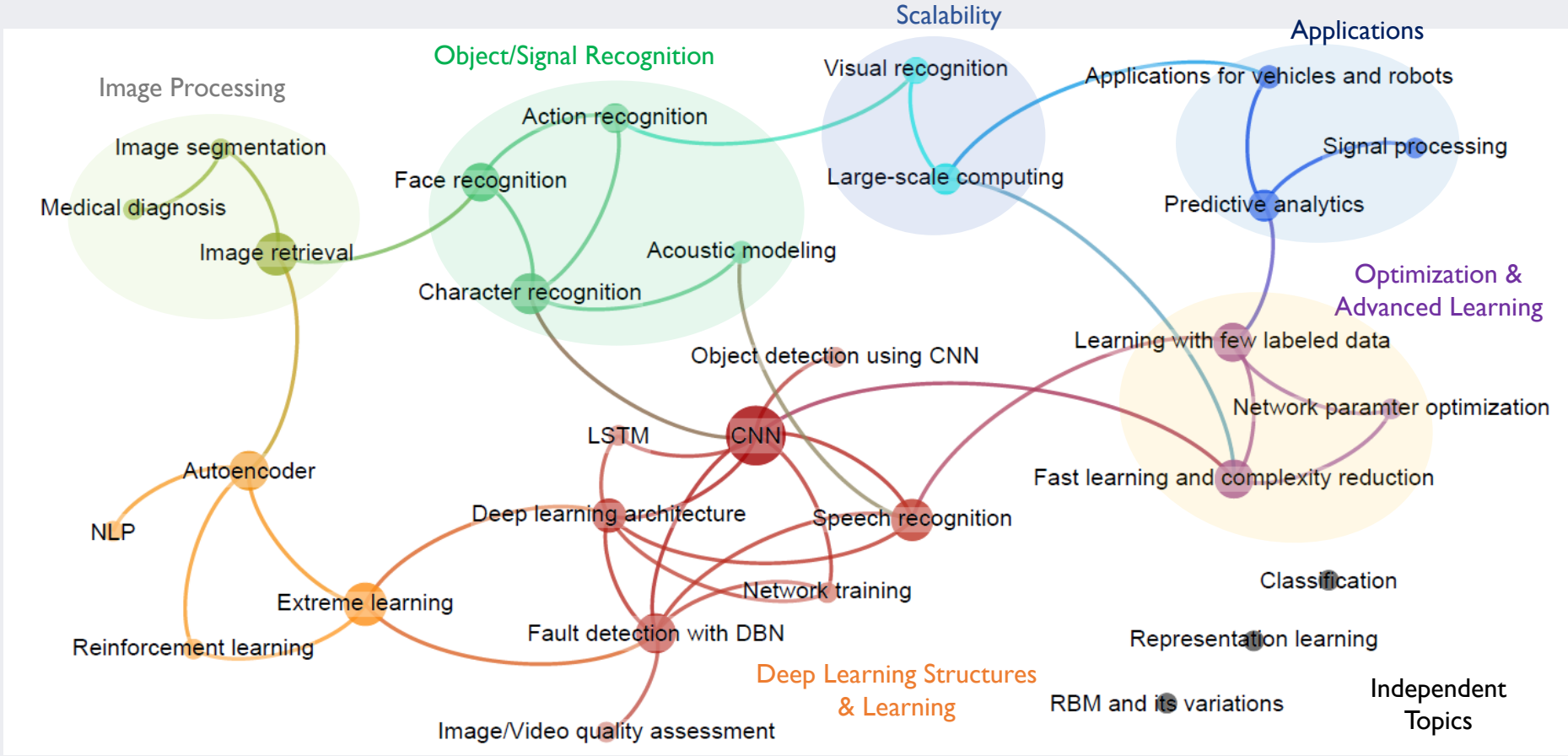
03 Network Representation

04 R Exercise

Network Representation

Kim et al. (2016)

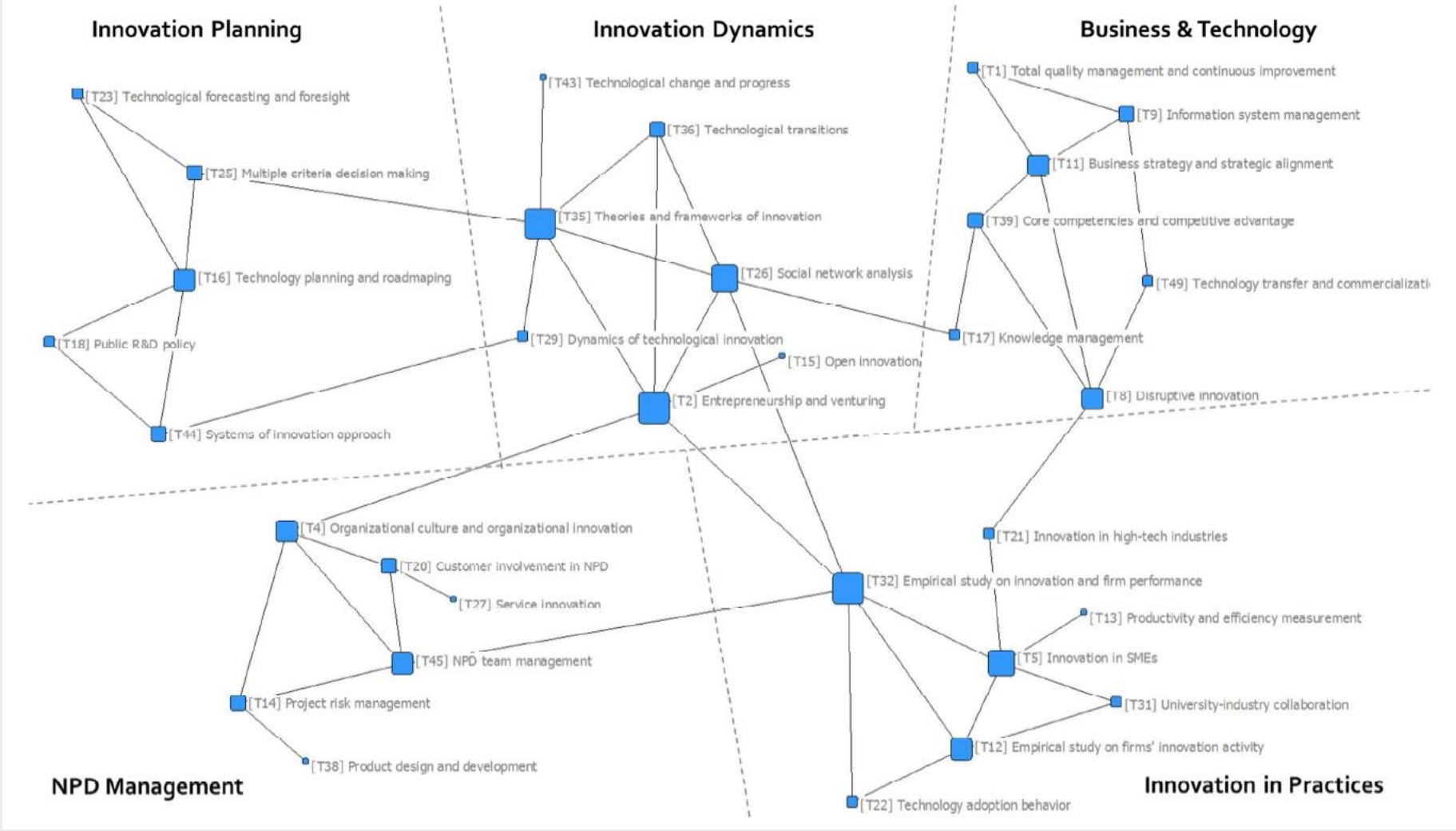
- Network of Deep Learning Topics Until Feb. 2016



Network Representation

Lee and Kang (2017)

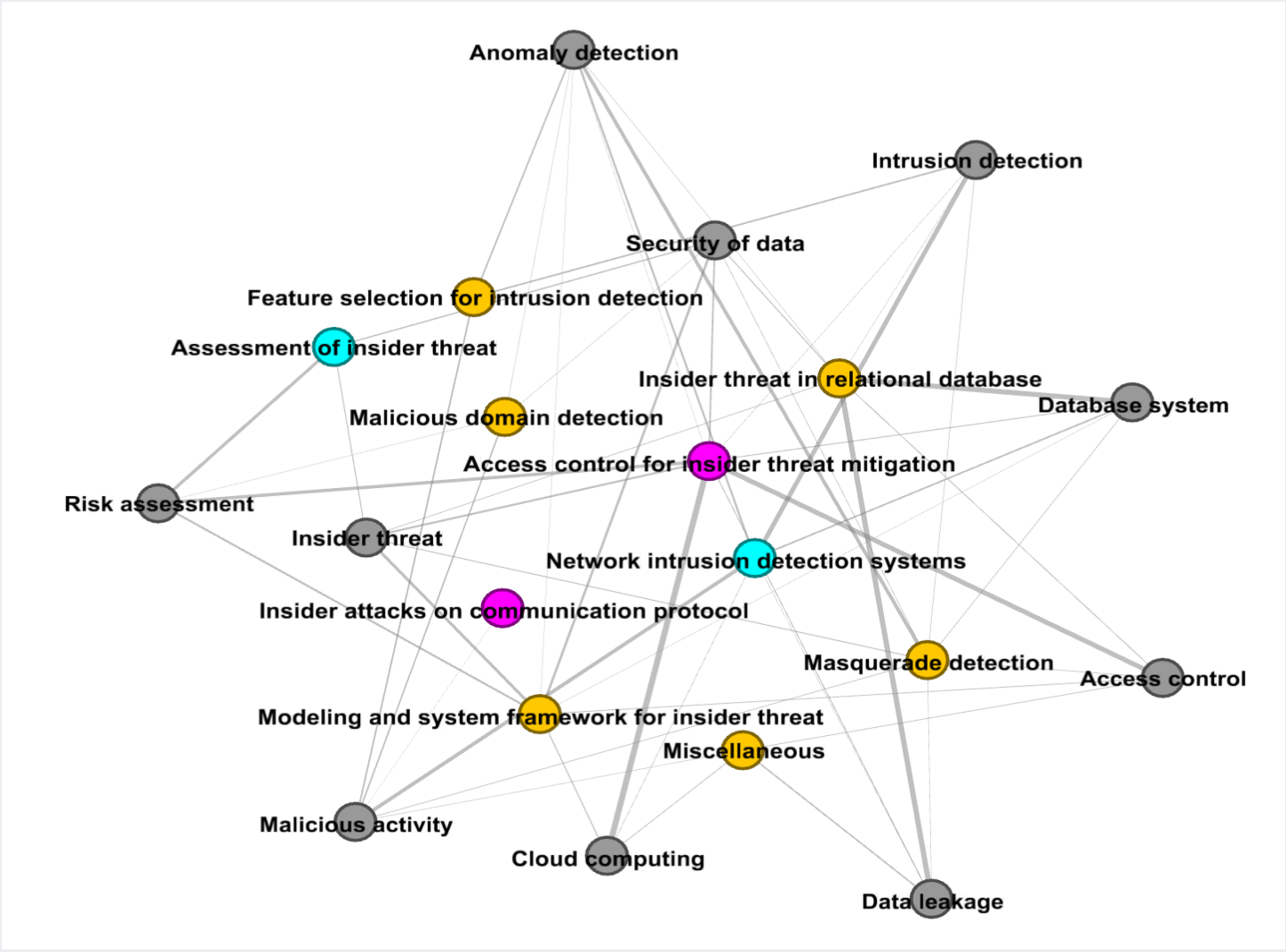
- Network of research topics for "technology and innovation management"



Network Representation

Kim et al. (2016)

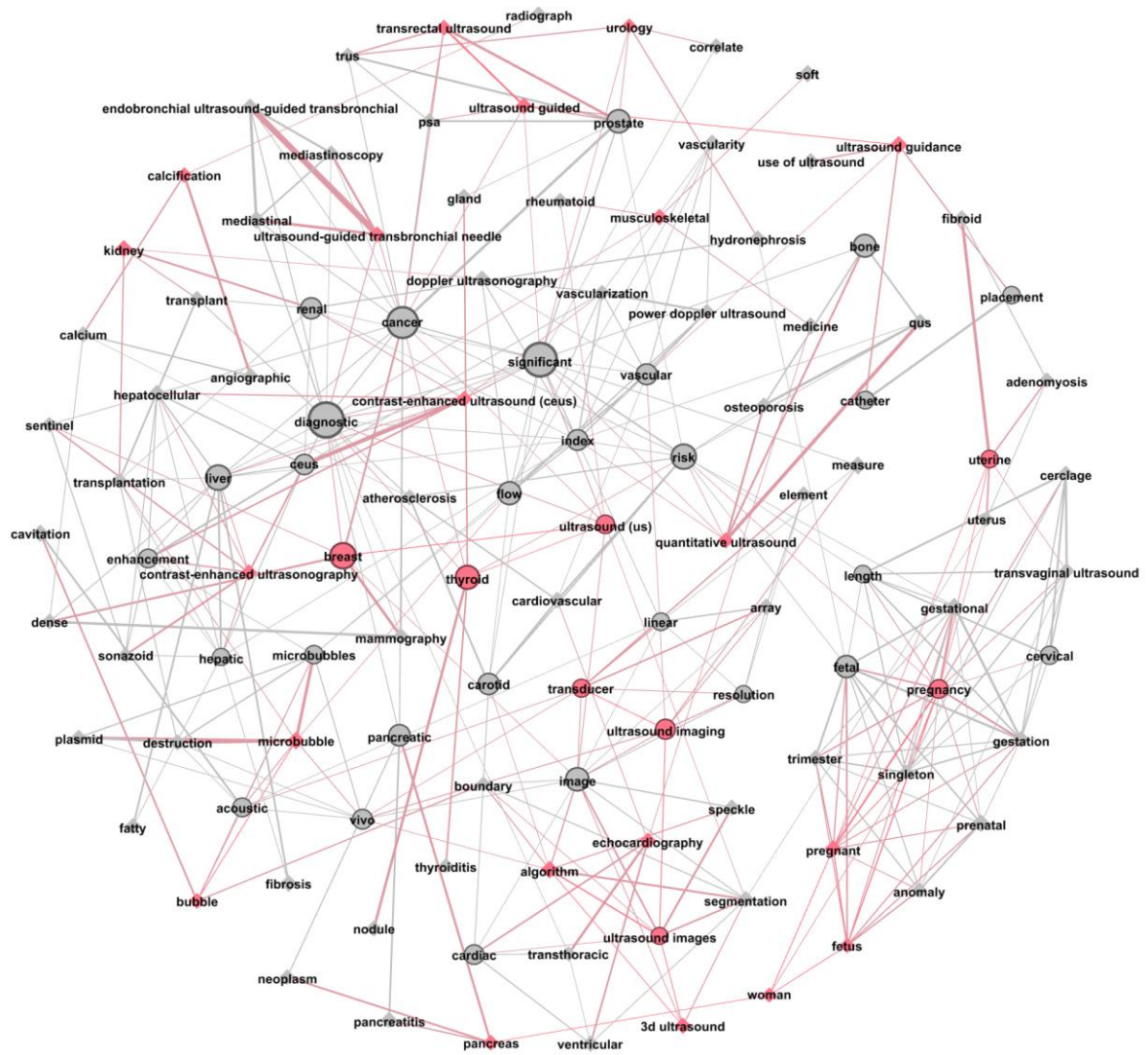
- Network of research topics for "Insider threats"



Network Representation

Kim et al. (2017)

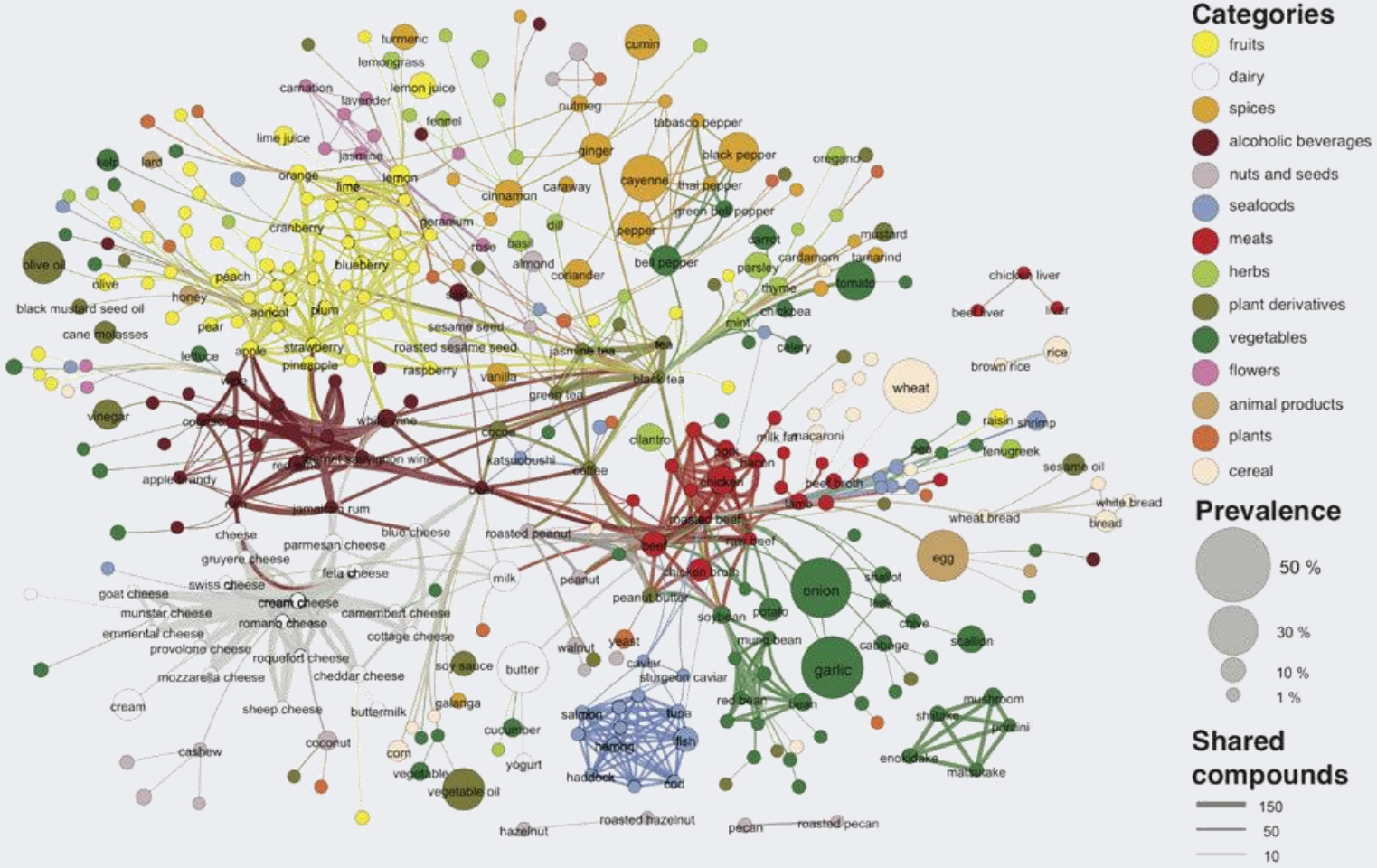
- Network of research topics for "Ultrasound and Ultrasonography"



Network Representation

Ahn et al. (2011)

- Summarize a collection of text documents using a network
  - ✓ The flavor network of the recipes



Network Representation

Ahn et al. (2011)

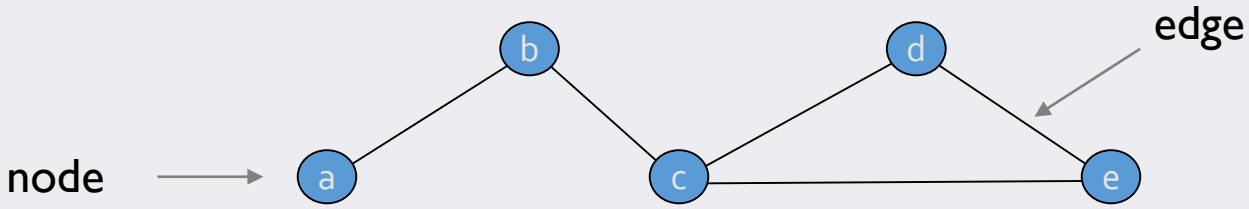
- Summarize a collection of text documents using a network
  - ✓ The flavor network of the recipes

Table S2: Number of recipes and the detailed cuisines in each regional cuisine in the recipe dataset. Five groups have reasonably large size. We use all cuisine data when calculating the relative prevalence and flavor principles.

Cuisine set	Number of recipes	Cuisines included
North American	41525	American, Canada, Cajun, Creole, Southern soul food, Southwestern U.S.
Southern European	4180	Greek, Italian, Mediterranean, Spanish, Portuguese
Latin American	2917	Caribbean, Central American, South American, Mexican
Western European	2659	French, Austrian, Belgian, English, Scottish, Dutch, Swiss, German, Irish
East Asian	2512	Korean, Chinese, Japanese
Middle Eastern	645	Iranian, Jewish, Lebanese, Turkish
South Asian	621	Bangladeshian, Indian, Pakistani
Southeast Asian	457	Indonesian, Malaysian, Filipino, Thai, Vietnamese
Eastern European	381	Eastern European, Russian
African	352	Moroccan, East African, North African, South African, West African
Northern European	250	Scandinavian

Network Representation

- What is a Network?
  - ✓ A network is a combined set of nodes connected by edges
  - ✓ Network ≡ Graph

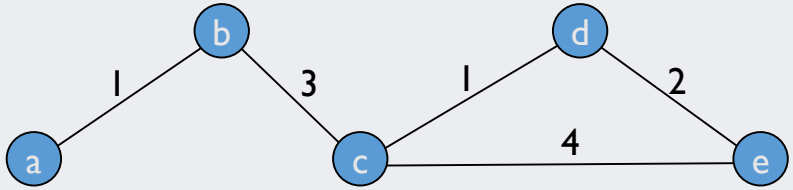


Points	Lines	Domain
Vertices	Edges, arcs	Math
Nodes	Links	Computer Science
Sites	Bonds	Physics
Actors	Ties, relations	Sociology
Keyword	Co-occurrence	Text Mining

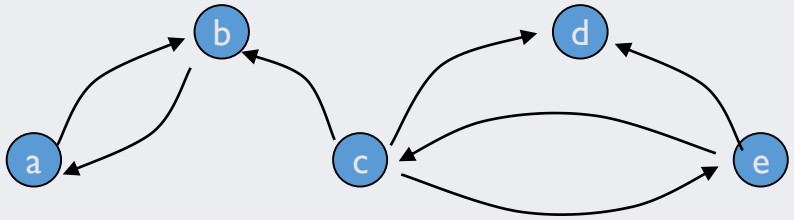
Network Representation: Connections

- Type of Connections
  - ✓ A relation can be binary or valued, directed or undirected

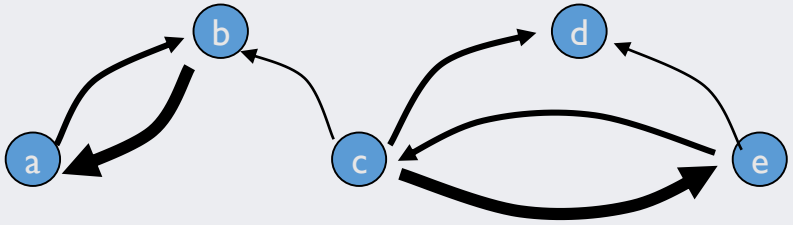
Undirected, binary



Undirected, Valued



Directed, binary

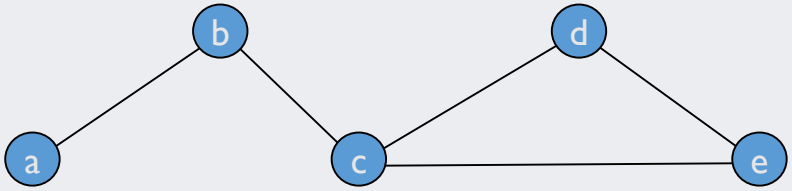


Directed, Valued

"Undirected networks are commonly used for Text Mining"

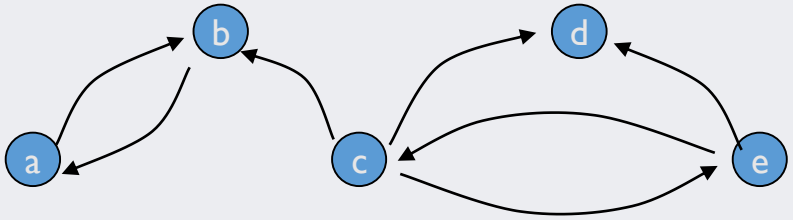
Network Representation: Data Structure

- Basic Data Structure
  - ✓ From pictures to matrices



Undirected, binary

	a	b	c	d	e
a		1			
b	1		1		
c		1		1	1
d			1		1
e			1	1	



Directed, binary

	a	b	c	d	e
a		1			
b	1				
c		1		1	1
d					
e			1	1	

Network Representation: Data Structure

- Basic Data Structure
  - ✓ From matrices to lists to save memory space

	a	b	c	d	e
a		1			
b	1		1		
c		1		1	1
d			1		1
e			1	1	

Adjacency List

a	b
b	a c
c	b d e
d	c e
e	c d

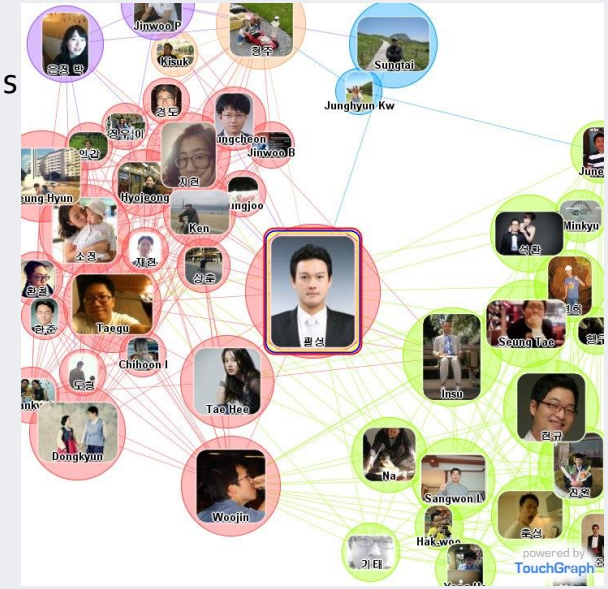
Arc List

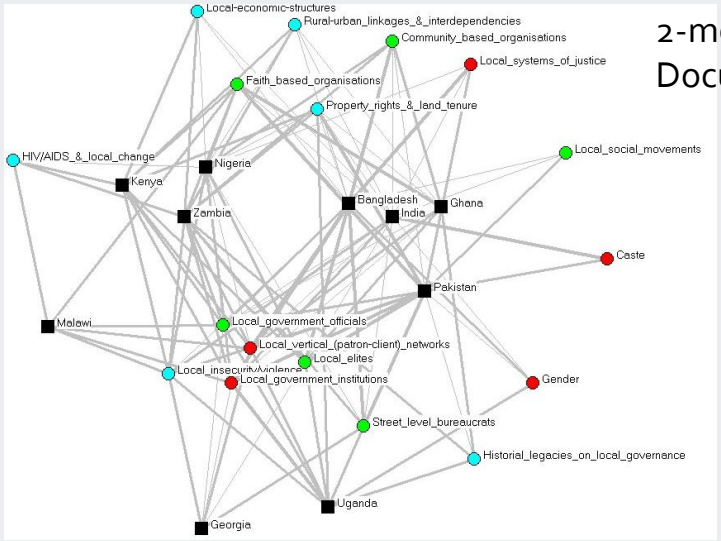
a b
b a
b c
c b
c d
c e
d c
d e
e c
e d

Network Representation: Data Structure

- N-mode data
  - ✓ 1-mode data represent edges based on direct contact between actors in the network
  - ✓ 2-mode data represent nodes from two separate classes, where all ties are across classes
    - People as author on papers, events in the life history of people

1-mode
Facebook friends

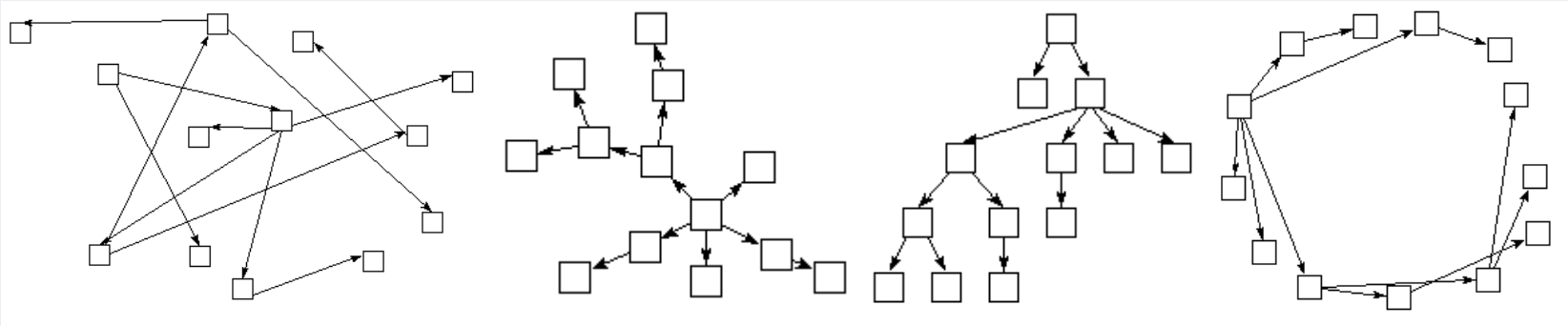




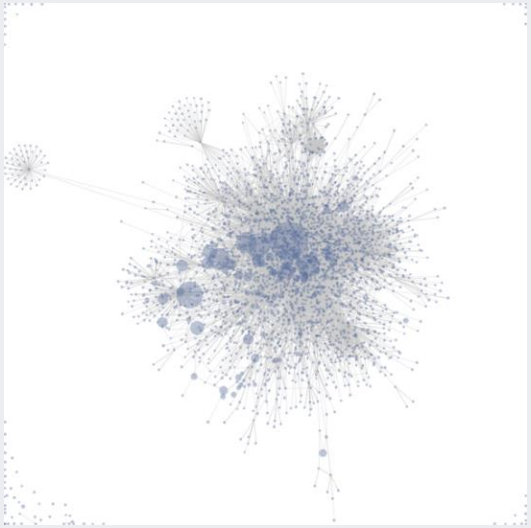
2-mode
Documents & nation

Network Representation: Layout

- Graphical Representation
  - ✓ No standard way to draw a sociogram
    - Each of these are equal



  - ✓ Force-directed layout (Fruchterman and Reingold, 1991)
    - Distribute the vertices evenly in the frame
    - Minimize edge crossings
    - Make edge lengths uniform
    - Reflect inherent symmetry
    - Conform to the frame



Network Representation: Measuring Networks

- Network Level
  - ✓ Size
    - Number of nodes
  - ✓ Density
    - Number of ties that are present
  - ✓ Out-degree (directed network)
    - Sum of connections from an actor to other
  - ✓ In-degree (undirected network)
    - Sum of connections to an actor

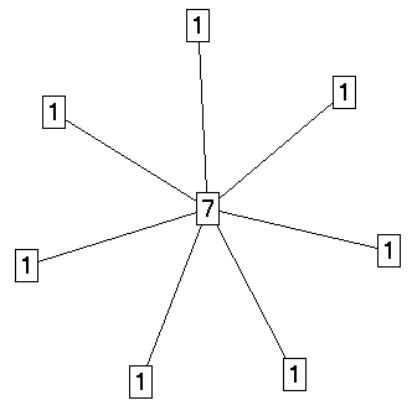
Network Representation: Measuring Networks

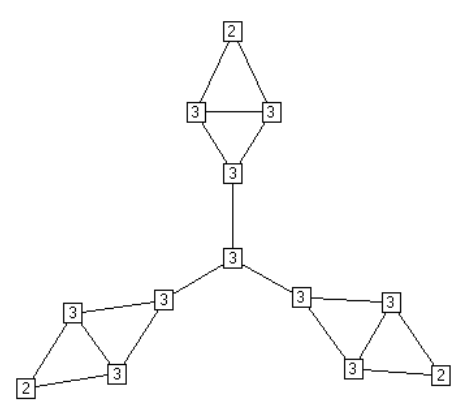
- Individual Node Level
  - ✓ Connectivity
    - refers to how actors in one part of the network are connected to actors in another part of the network
    - Reachability
      - Is it possible for actor i to reach actor j?
      - True only if there is a chain of contact from one actor to another
    - Distance
      - Given they can be reached, how many steps are they from each other?
    - Number of paths
      - How may different paths connect each pair?

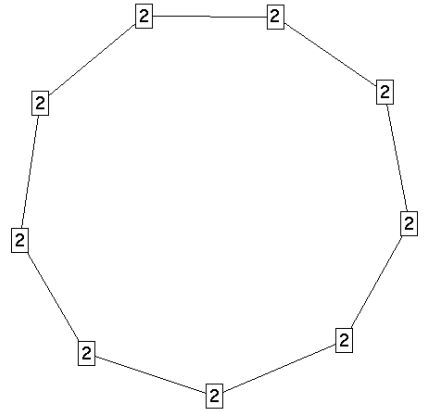
Network Representation: Measuring Networks

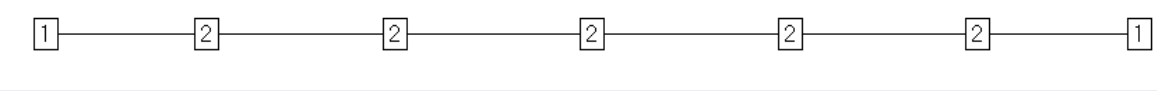
- Individual Node Level
  - ✓ Centrality
    - Refers to location, identifying where an actor resides in a network
    - Commonly believed that actors in the "center" of the network are "important"
    - Degree
      - the number of edges connected to the actor

$$C_D = d(n_i) = X_{i+} = \sum_j X_{ij}$$





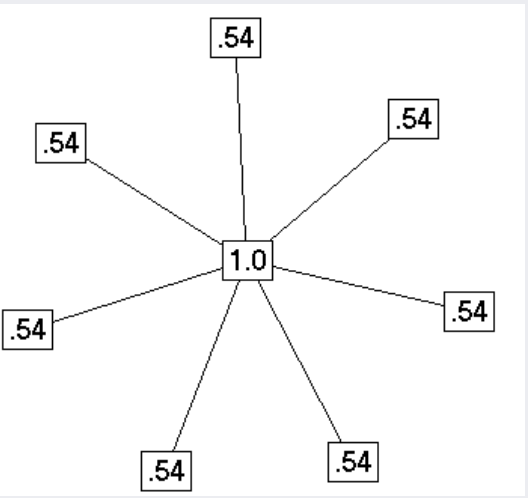


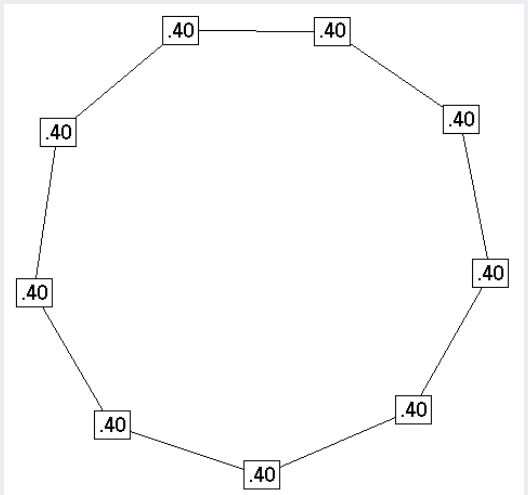


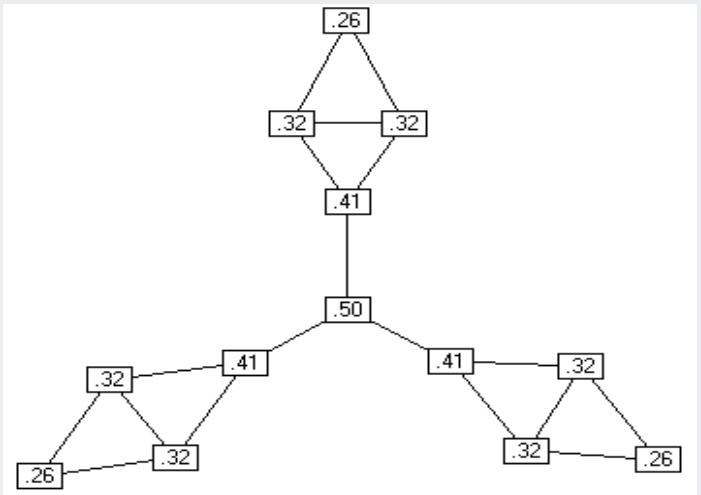
Network Representation: Measuring Networks

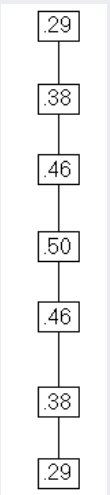
- Individual Node Level
  - ✓ Centrality
    - Closeness
      - An actor is considered important if it is relatively close to all other actors
      - based on the inverse of the distance of each actor to every other actor in the network, often normalized

$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j)\right]^{-1} \qquad C'_c(n_i) = C_c(n_i)(g-1)$$



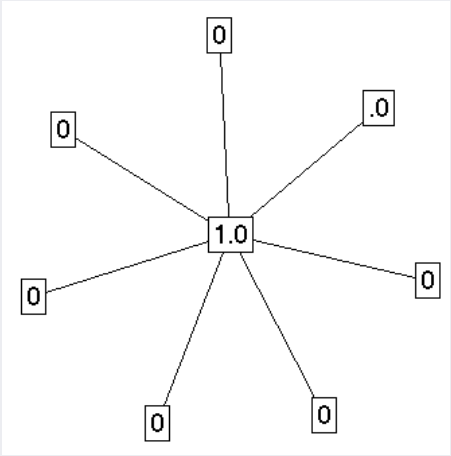


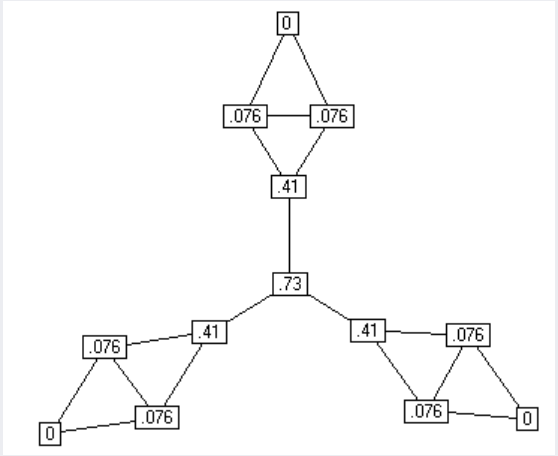


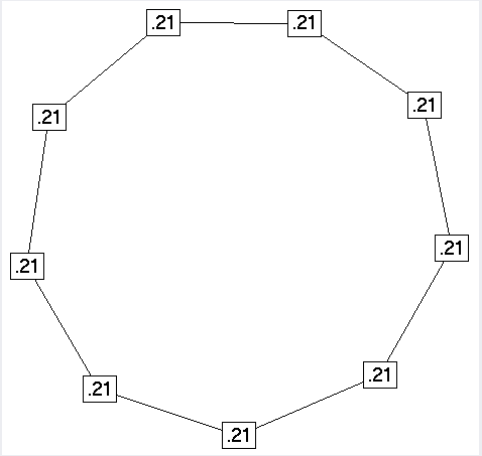


Network Representation: Measuring Networks

- Individual Node Level
  - ✓ Centrality
    - Betweenness
      - The number of shortest paths between i and k that actor j resides on

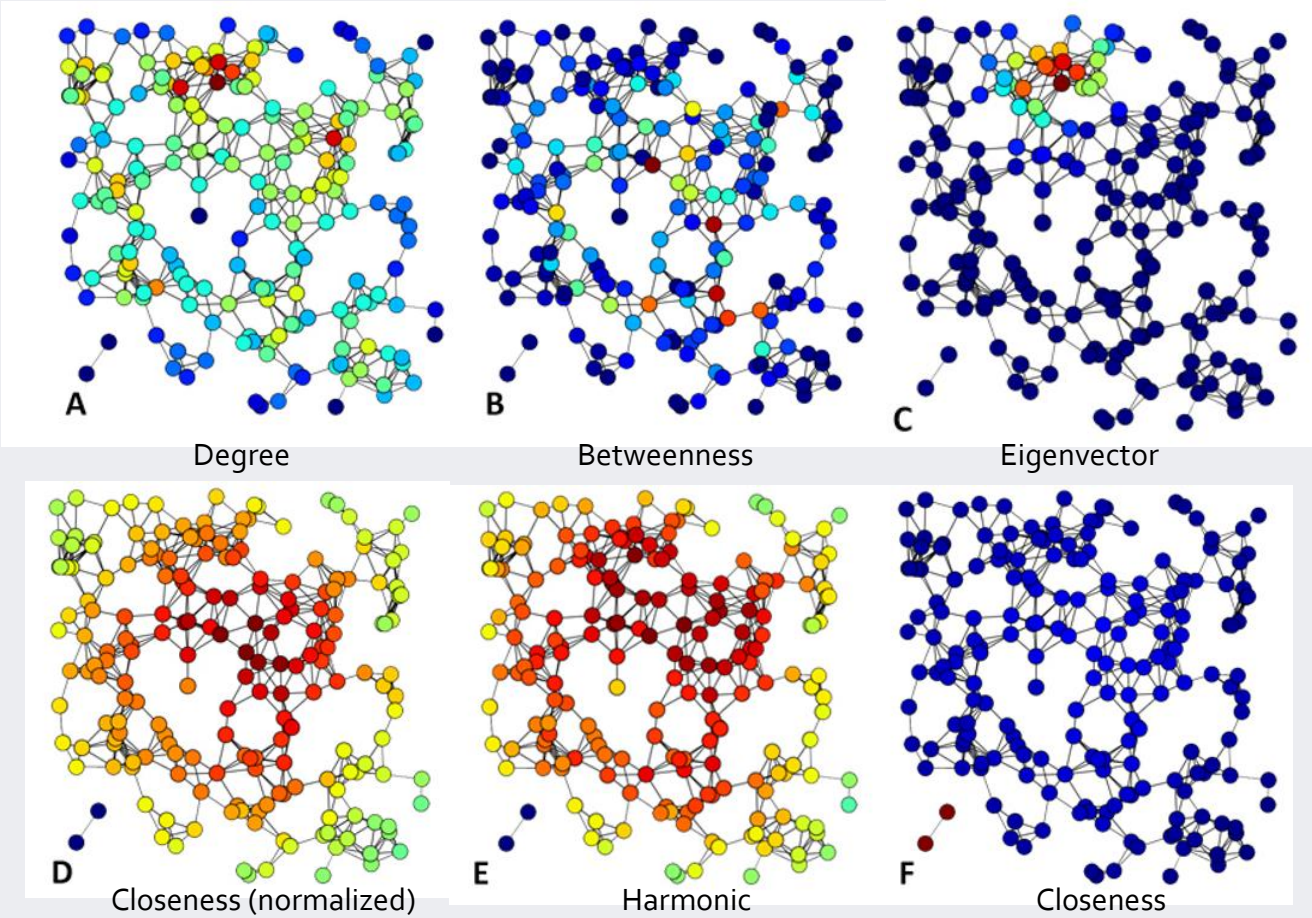






Network Representation: Measuring Networks

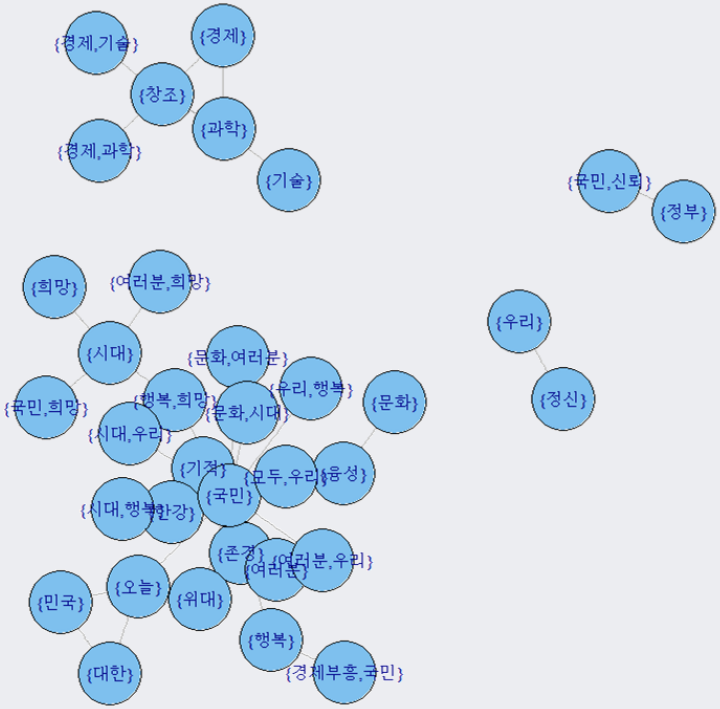
- Individual Node Level
  - ✓ Centrality
    - Eigenvector centrality, Harmonic centrality



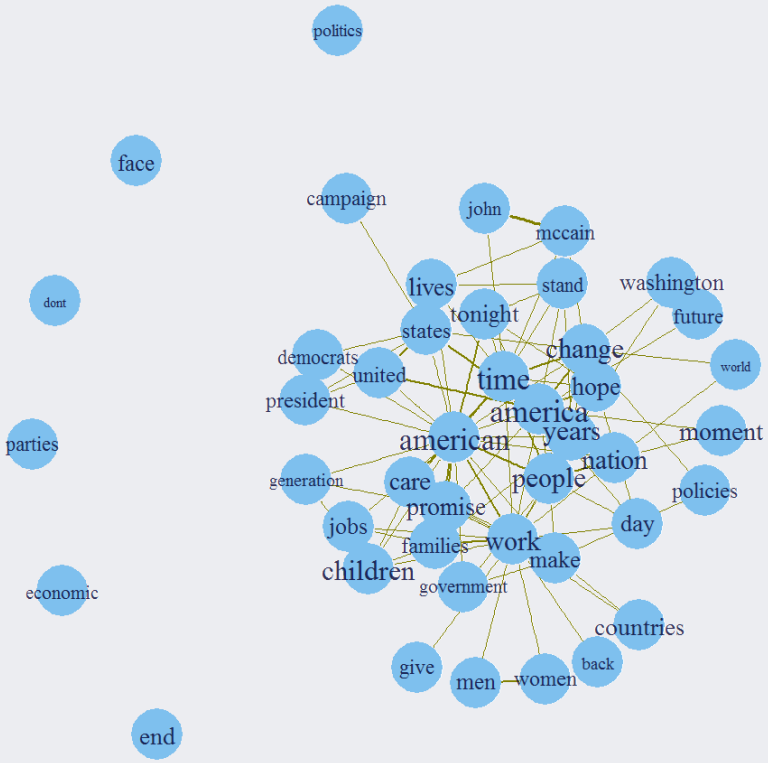
Network Representation: Community Structure

- Community Structure
  - ✓ Nodes of a network can be grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally.

Four non-overlapping communities



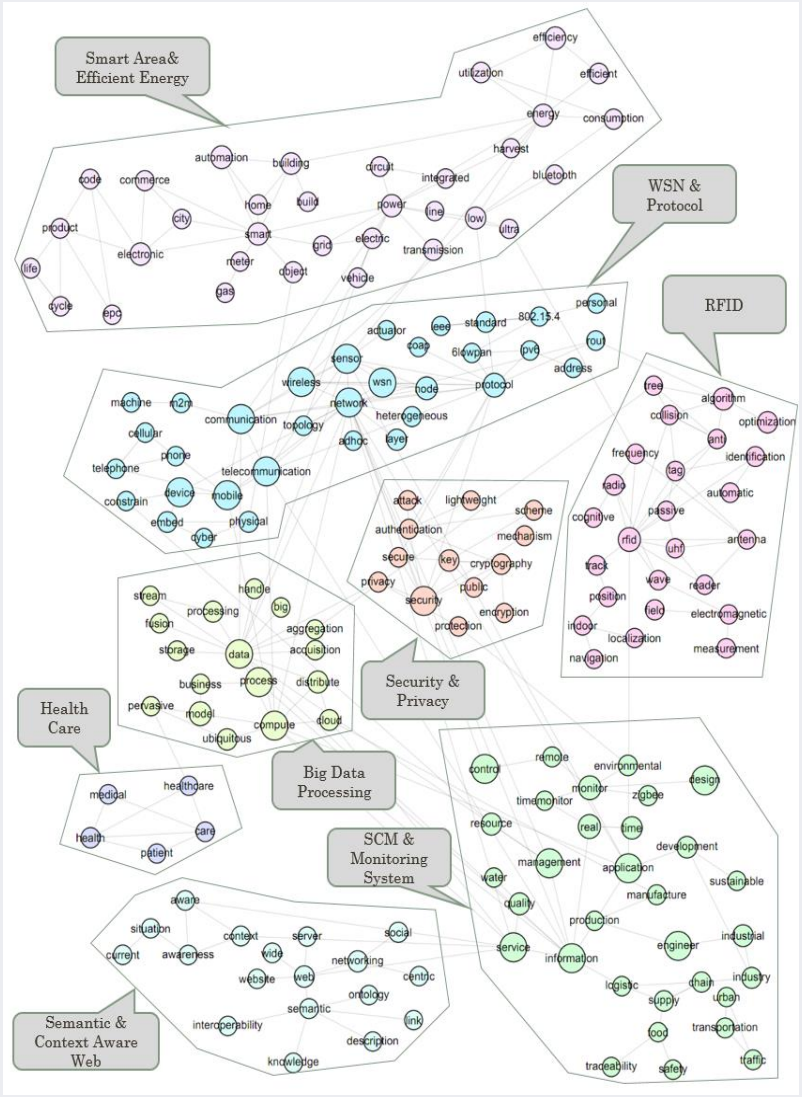
Difficult to identify the community structure



Network Representation: Community Structure

Kim & Kang (2016)

- Keyword network of Research Papers for "Internet of Things"



AGENDA

01 Word Cloud

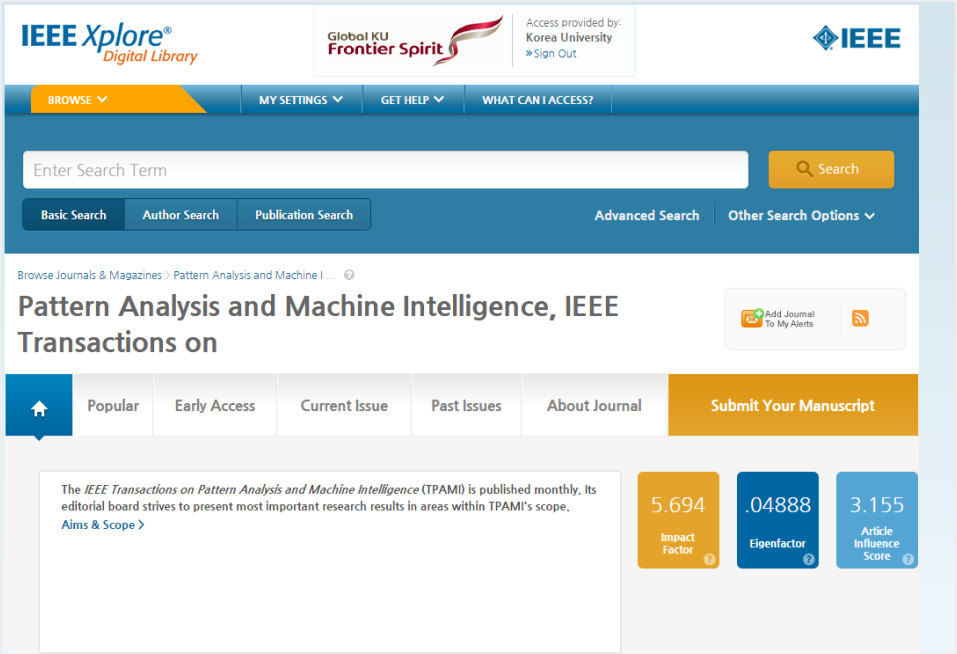
02 Association Rules

03 Network Representation

04 R Exercise

R Exercise: Word Cloud

- 100 abstracts from two journals
  - ✓ Journal 1: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
  - ✓ Journal 2: Journal of Finance (JoF)



R Exercise: Word Cloud

- Construct Corpuses & Preprocessing
  - ✓ To lower case, remove punctuations/numbers/stopwords, stemming

```
# Load the data
TPAMI <- read.csv("IEEE_TPAMI.csv", encoding = "UTF-8", stringsAsFactors = FALSE)
JoF <- read.csv("Journal of Finance.csv", encoding = "UTF-8", stringsAsFactors = FALSE)

# 1. Wordcloud -----------------------------------------------------

# Construct the corpus for each journal with the abstracts
TPAMI.Corpus <- Corpus(VectorSource(TPAMI$Abstract))
JoF.Corpus <- Corpus(VectorSource(JoF$Abstract))

# Preprocessing
# 1: to lower case
TPAMI.Corpus <- tm_map(TPAMI.Corpus, content_transformer(stri_trans_tolower))
JoF.Corpus <- tm_map(JoF.Corpus, content_transformer(stri_trans_tolower))

# 2: remove puntuations
TPAMI.Corpus <- tm_map(TPAMI.Corpus, content_transformer(removePunctuation))
JoF.Corpus <- tm_map(JoF.Corpus, content_transformer(removePunctuation))

# 3. remove numbers
TPAMI.Corpus <- tm_map(TPAMI.Corpus, content_transformer(removeNumbers))
JoF.Corpus <- tm_map(JoF.Corpus, content_transformer(removeNumbers))

# 4. remove stopwords (SMART stopwords list)
myStopwords <- c(stopwords("SMART"))

TPAMI.Corpus <- tm_map(TPAMI.Corpus, removeWords, myStopwords)
JoF.Corpus <- tm_map(JoF.Corpus, removeWords, myStopwords)

# 5. Stemming
TPAMI.Corpus <- tm_map(TPAMI.Corpus, stemDocument)
JoF.Corpus <- tm_map(JoF.Corpus, stemDocument)

# 4. remove stopwords (with frequently used words)
myStopwords <- c(stopwords("SMART"), "financ", "american", "associ", "firm",
                 "model", "data", "algorithm", "method", "imag")

TPAMI.Corpus <- tm_map(TPAMI.Corpus, removeWords, myStopwords)
JoF.Corpus <- tm_map(JoF.Corpus, removeWords, myStopwords)
```

R Exercise: Word Cloud

- Construct Term-Document Matrices
  - ✓ TPAMI: 1,827 terms & 100 documents
  - ✓ JoF: 1,354 terms & 100 documents

```
55  # Term-Document Matrix
56  TPAMI.TDM <- TermDocumentMatrix(TPAMI.Corpus, control = list(minWordLength = 1))
57  JoF.TDM <- TermDocumentMatrix(JoF.Corpus, control = list(minWordLength = 1))
58
59  # Term-Document Matrix
60  TPAMI.TDM
61  JoF.TDM
62
63  as.matrix(TPAMI.TDM)[11:30,11:30]
64  as.matrix(JoF.TDM)[11:30,11:30]
```

```
> TPAMI.TDM
<<TermDocumentMatrix (terms: 1827, documents: 100)>>
Non-/sparse entries: 6572/176128
Sparsity           : 96%
Maximal term length: 21
Weighting          : term frequency (tf)
> JoF.TDM
<<TermDocumentMatrix (terms: 1354, documents: 100)>>
Non-/sparse entries: 3962/131438
Sparsity           : 97%
Maximal term length: 19
Weighting          : term frequency (tf)
```

R Exercise: Word Cloud

- Frequently used words for each journal

```
66  # Frequently used words
67  findFreqTerms(TPAMI.TDM, lowfreq=15)
68  findFreqTerms(JoF.TDM, lowfreq=15)
```

```
> findFreqTerms(TPAMI.TDM, lowfreq=15)
  [1] "accuraci"     "achiev"       "action"       "adapt"        "address"      "analysi"      "appli"        "applic"
  [9] "approach"     "approxim"     "base"         "bayesian"     "benchmark"    "call"         "case"         "class"
 [17] "classif"      "classifi"     "cluster"      "code"         "combin"       "compar"       "complet"      "complex"
 [25] "comput"       "consist"      "databas"      "demonstr"     "depend"       "describ"      "descriptor"   "detect"
 [33] "develop"      "dimens"       "dirichlet"    "discrimin"    "distanc"      "distort"      "distribut"    "domain"
 [41] "effect"       "effici"       "error"        "estim"        "evalu"        "exist"        "experi"       "experiment"
 [49] "exploit"      "extens"       "featur"       "filter"       "find"         "fingerprint"  "formul"       "framework"
 [57] "function"     "general"      "generat"      "graph"        "hierarch"     "high"         "ieee"         "improv"
 [65] "includ"       "infer"        "inform"       "introduc"     "kernel"       "label"        "larg"         "latent"
 [73] "learn"        "linear"       "local"        "make"         "map"          "match"        "matrix"       "measur"
 [81] "motion"       "multipl"      "natur"        "nois"         "nonparametr"  "number"       "object"       "observ"
 [89] "obtain"       "optim"        "outperform"   "paper"        "paramet"      "part"         "partit"       "perform"
 [97] "point"        "pose"         "power"        "predict"      "present"      "prior"        "probabl"      "problem"
[105] "process"      "properti"     "propos"       "provid"       "real"         "recognit"     "reconstruct"  "reduc"
[113] "region"       "regress"      "relat"        "repres"       "represent"    "result"       "robust"       "sampl"
[121] "scene"        "search"       "set"          "shape"        "show"         "signific"     "similar"      "sourc"
[129] "space"        "spars"        "spatial"      "specif"       "stateoftheart" "statist"     "strategi"     "structur"
[137] "studi"        "support"      "svms"         "synthet"      "system"       "target"       "task"         "techniqu"
[145] "templat"      "tensor"       "term"         "test"         "time"         "track"        "train"        "trajectori"
[153] "transform"    "tree"         "variat"       "vector"       "video"        "vision"       "visual"       "wide"
[161] "word"         "work"
> findFreqTerms(JoF.TDM, lowfreq=15)
 [1] "account"     "activ"       "affect"      "asset"       "bank"        "bidder"      "capit"       "cash"        "consist"
[10] "correl"      "cost"        "credit"      "crosssect"   "debt"        "default"     "document"    "econom"      "effect"
[19] "equilibrium" "equiti"      "estim"       "evid"        "examin"      "expect"      "factor"      "financi"     "find"
[28] "flow"        "fund"        "govern"      "higher"      "hold"        "import"      "incent"      "increas"     "industri"
[37] "inform"      "invest"      "investor"    "larg"        "leverag"     "liquid"      "lower"       "manag"       "market"
[46] "measur"      "merger"      "option"      "paper"       "perform"     "portfolio"   "predict"     "price"       "product"
[55] "rate"        "relat"       "result"      "return"      "risk"        "sensit"      "shock"       "show"        "signific"
[64] "spread"      "stock"       "structur"    "studi"       "suggest"     "target"      "tax"         "time"        "trade"
[73] "volatil"
```

R Exercise: Word Cloud

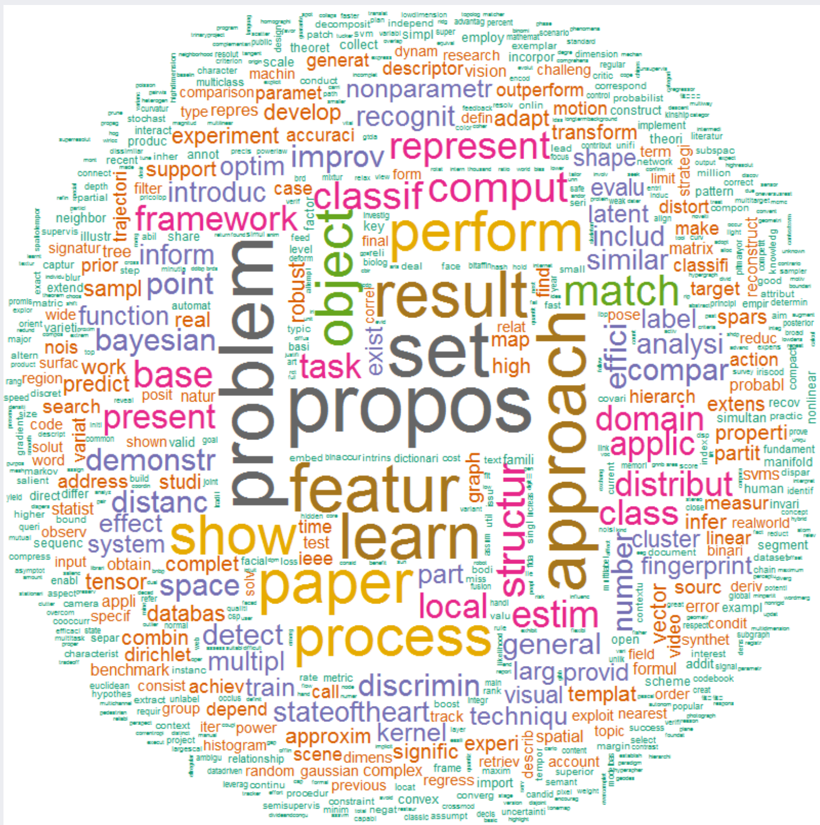
- Construct Word Clouds

```
70  # Construct a Word Cloud with IEEE_TPAMI abstracts
71  TPAMI.wcmat <- as.matrix(TPAMI.TDM)
72
73  # calculate the frequency of words
74  TPAMI.word.freq <- sort(rowSums(TPAMI.wcmat), decreasing=TRUE)
75  TPAMI.keywords <- names(TPAMI.word.freq)
76  TPAMI.wcdat <- data.frame(word = TPAMI.keywords, freq = TPAMI.word.freq)
77
78  pal <- brewer.pal(8, "Dark2")
79  wordcloud(TPAMI.wcdat$word, TPAMI.wcdat$freq, min.freq=3, scale = c(5, 0.2), rot.per = 0.1, col=pal, random.order=F)
80
81  # Construct a Word Cloud with Romney's speeches
82  JoF.wcmat <- as.matrix(JoF.TDM)
83
84  # calculate the frequency of words
85  JoF.word.freq <- sort(rowSums(JoF.wcmat), decreasing=TRUE)
86  JoF.keywords <- names(JoF.word.freq)
87  JoF.wcdat <- data.frame(word = JoF.keywords, freq = JoF.word.freq)
88
89  pal <- brewer.pal(8, "Dark2")
90  wordcloud(JoF.wcdat$word, JoF.wcdat$freq, min.freq=3, scale = c(5, 0.2), rot.per = 0.1, col=pal, random.order=F)
```

R Exercise: Word Cloud

- Construct Word Clouds

TPAMI



JoF

R Exercise: Association Rules

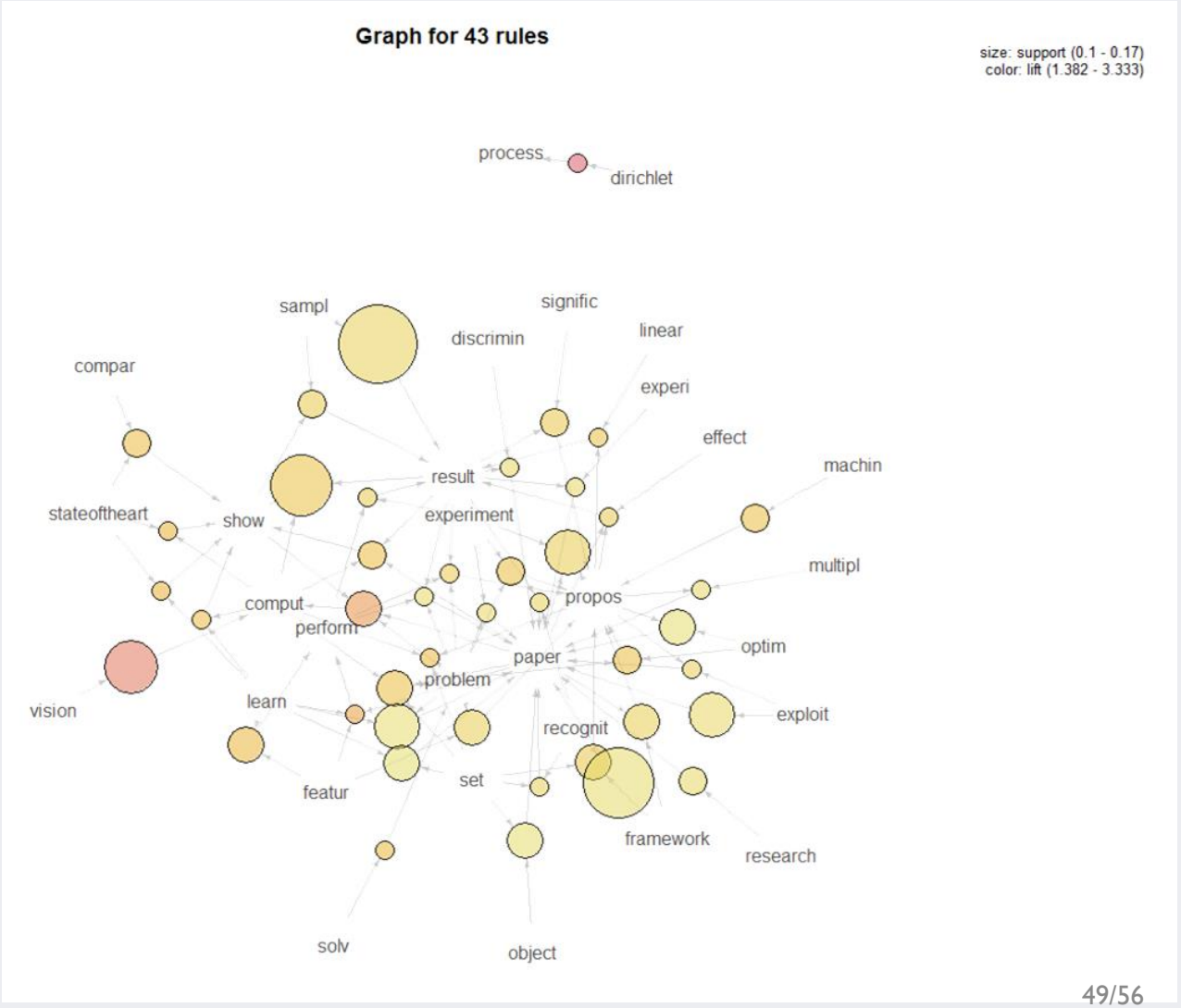
- Parameters for A-priori algorithm

```
# Association Rules for IEEE_TPAMI
TPAMI.tran <- as.matrix(t(TPAMI.TDM))
TPAMI.tran <- as(TPAMI.tran, "transactions")

TPAMI.rules <- apriori(TPAMI.tran, parameter=list(minlen=2,supp=0.1, conf=0.85))
inspect(TPAMI.rules)

TPAMI.rules.sorted <- sort(TPAMI.rules, by="lift")
subset.matrix <- is.subset(TPAMI.rules.sorted, TPAMI.rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
redundant <- colSums(subset.matrix, na.rm=T) >= 1
TPAMI.rules.pruned <- TPAMI.rules.sorted[!redundant]
inspect(TPAMI.rules.pruned)

# Plot the rules
plot(TPAMI.rules.pruned, method="graph")

# Association Rules for Journal of Fiance
JoF.tran <- as.matrix(t(JoF.TDM))
JoF.tran <- as(JoF.tran, "transactions")

JoF.rules <- apriori(JoF.tran, parameter=list(minlen=2, supp=0.06, conf=0.8))
inspect(JoF.rules)

JoF.rules.sorted <- sort(JoF.rules, by="lift")
subset.matrix <- is.subset(JoF.rules.sorted, JoF.rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
redundant <- colSums(subset.matrix, na.rm=T) >= 1
JoF.rules.pruned <- JoF.rules.sorted[!redundant]
inspect(JoF.rules.pruned)

# Plot the rules
plot(JoF.rules.pruned, method="graph")
```

R Exercise: Association Rules

- Generated rules from TPAMI Abstracts

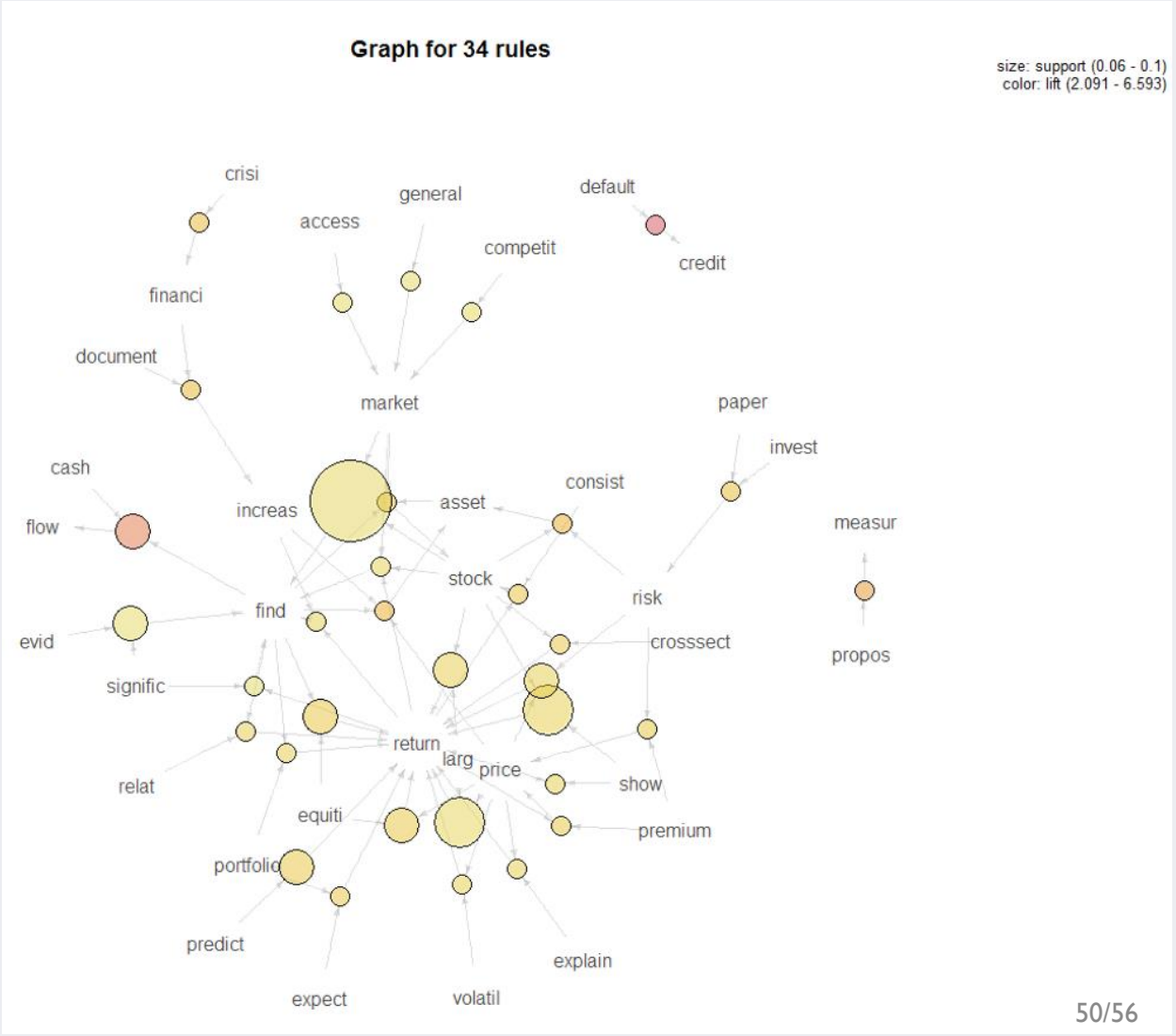
```
> inspect(TPAMI.rules.pruned)
   lhs                    rhs          support confidence     lift
1  {dirichlet}         => {process}       0.10  1.0000000 3.333333
2  {vision}            => {comput}        0.14  1.0000000 3.030303
3  {paper,
    problem,
    show}              => {comput}        0.12  0.8571429 2.597403
4  {featur,
    learn,
    paper}             => {perform}       0.10  1.0000000 2.380952
5  {learn,
    stateoftheart}     => {show}          0.10  1.0000000 2.127660
6  {comput,
    stateoftheart}     => {show}          0.10  1.0000000 2.127660
7  {featur,
    learn}             => {perform}       0.12  0.8571429 2.040816
8  {solv}              => {problem}       0.10  0.9090909 2.020202
9  {comput,
    propos,
    set}               => {problem}       0.10  0.9090909 2.020202
10 {compar,
    stateoftheart}     => {show}          0.11  0.9166667 1.950355
11 {comput,
    paper,
    result}            => {show}          0.11  0.9166667 1.950355
12 {comput,
    learn}             => {show}          0.10  0.9090909 1.934236
13 {machin}            => {propos}        0.11  1.0000000 1.923077
14 {experiment,
    problem}           => {propos}        0.11  1.0000000 1.923077
15 {optim,
    problem}           => {propos}        0.11  1.0000000 1.923077
16 {comput,
    paper,
    set}               => {problem}       0.12  0.8571429 1.904762
17 {comput,
    result}            => {show}          0.15  0.8823529 1.877347
18 {linear,
    propos}            => {result}        0.10  1.0000000 1.851852
19 {framework,
    set}               => {propos}        0.12  0.9230769 1.775148
20 {result,
    signific}          => {propos}        0.11  0.9166667 1.762821
21 {perform,
    problem,
    result}            => {propos}        0.10  0.9090909 1.748252
22 {sampl,
    show}              => {result}        0.11  0.9166667 1.697531
23 {experiment,
    perform}           => {result}        0.10  0.9090909 1.683502
24 {effect,
    paper,
    propos}            => {result}        0.10  0.9090909 1.683502
25 {experiment,
    paper}             => {propos}        0.13  0.8666667 1.666667
26 {framework,
    paper}             => {propos}        0.12  0.8571429 1.648352
27 {featur,
    problem}           => {propos}        0.12  0.8571429 1.648352
28 {exploit,
    propos}            => {paper}         0.10  1.0000000 1.612903
29 {recognit,
    set}               => {paper}         0.10  1.0000000 1.612903
30 {recognit,
    result}            => {paper}         0.10  1.0000000 1.612903
```



R Exercise: Association Rules

- Generated rules from JoF Abstracts

```
> inspect(JoF.rules.pruned)
   lhs             rhs       support confidence     lift
1  {default}    => {credit}     0.06  0.8571429 6.593407
2  {cash,
    find}       => {flow}       0.07  0.8750000 5.468750
3  {propos}     => {measur}     0.06  0.8571429 4.511278
4  {risk,
    stock}      => {asset}      0.06  1.0000000 3.703704
5  {find,
    increas,
    price}      => {asset}      0.06  1.0000000 3.703704
6  {asset,
    find,
    market}     => {stock}      0.06  1.0000000 3.571429
7  {document,
    financi}    => {increas}    0.06  1.0000000 3.448276
8  {invest,
    paper}      => {risk}       0.06  1.0000000 3.333333
9  {crisi}      => {financi}    0.06  0.8571429 3.296703
10 {consist,
    return}     => {stock}      0.06  0.8571429 3.061224
11 {expect,
    portfolio}  => {return}     0.06  1.0000000 2.941176
12 {portfolio,
    predict}    => {return}     0.07  1.0000000 2.941176
13 {equiti,
    price}      => {return}     0.07  1.0000000 2.941176
14 {equiti,
    find}       => {return}     0.07  1.0000000 2.941176
15 {crosssect,
    stock}      => {return}     0.06  1.0000000 2.941176
16 {premium,
    risk}       => {price}      0.06  0.8571429 2.857143
17 {premium,
    return}     => {price}      0.06  0.8571429 2.857143
18 {show,
    stock}      => {return}     0.08  0.8888889 2.614379
19 {larg,
    stock}      => {return}     0.07  0.8750000 2.573529
20 {price,
    risk}       => {return}     0.07  0.8750000 2.573529
21 {explain,
    price}      => {return}     0.06  0.8571429 2.521008
22 {price,
    volatil}    => {return}     0.06  0.8571429 2.521008
23 {find,
    relat}      => {return}     0.06  0.8571429 2.521008
24 {find,
    portfolio}  => {return}     0.06  0.8571429 2.521008
25 {price,
    show}       => {return}     0.06  0.8571429 2.521008
26 {increas,
    return}     => {find}       0.06  1.0000000 2.439024
27 {market,
    return,
    stock}      => {find}       0.06  1.0000000 2.439024
28 {larg,
    price}      => {return}     0.08  0.8000000 2.352941
29 {market,
    stock}      => {find}       0.10  0.9090909 2.217295
30 {general}    => {market}     0.06  0.8571429 2.197802
```



R Exercise: Keyword Network

- Transform the Term-Frequency matrix into Term-Term co-occurrence matrix

```
126 # Section 3: Keyword Network ------------------------------------------
127 # For IEEE_TPAMI Abstracts
128 # Change it to a Boolean matrix
129 TPAMI.wcmat[TPAMI.wcmat >= 1] <- 1
130 # find the words that are used more than 10 times
131 freq.idx <- which(rowSums(TPAMI.wcmat) >= 10)
132 TPAMI.wcmat.freq <- TPAMI.wcmat[freq.idx,]
133
134 # Transform into a term-term adjacency matrix
135 TPAMI.ttmat <- TPAMI.wcmat.freq %*% t(TPAMI.wcmat.freq)
136
137 # inspect terms numbered 1 to 10
138 TPAMI.ttmat[1:10,1:10]
```

```
> TPAMI.ttmat[1:10,1:10]
          Terms
Terms      account accuraci achiev addit address analysi appli applic approach approxim
  account       11        4      0     2       5       2     1      3        3        0
  accuraci       4       12      2     3       3       3     0      3        5        3
  achiev         0        2     18     3       4       3     2      6        9        2
  addit          2        3      3    11       2       2     0      3        3        4
  address        5        3      4     2      18       1     2      4        9        2
  analysi        2        3      3     2       1      19     3      5       10        2
  appli          1        0      2     0       2       3    12      6        3        1
  applic         3        3      6     3       4       5     6     31        9        3
  approach       3        5      9     3       9      10     3      9       40        7
  approxim       0        3      2     4       2       2     1      3        7       13
```

R Exercise: Keyword Network

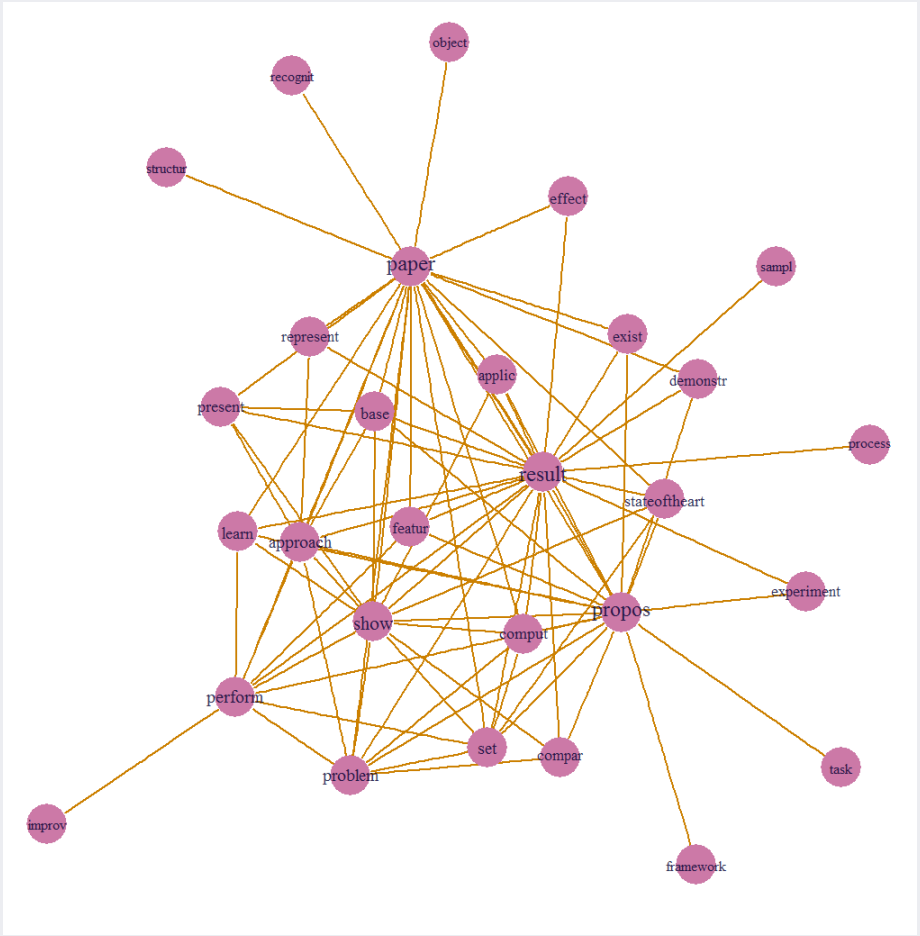
- Build a graph & find communities
  - ✓ Undirected, simplification (remove loops), etc.

```
# Build a graph from the above matrix
TPAMI.graph <- graph.adjacency(TPAMI.ttmat, weighted=T, mode = "undirected")

# remove loops
TPAMI.graph <- simplify(TPAMI.graph)

# set labels and degrees of vertices
V(TPAMI.graph)$label <- V(TPAMI.graph)$name
TPAMI.graph <- delete.edges(TPAMI.graph, which(E(TPAMI.graph)$weight <= 15))

TPAMI.graph <- delete.vertices(TPAMI.graph, which(degree(TPAMI.graph) == 0))
V(TPAMI.graph)$degree <- degree(TPAMI.graph)

# set seed to make the layout reproducible
set.seed(3952)
plot(TPAMI.graph, layout=layout.fruchterman.reingold)
plot(TPAMI.graph, layout=layout.kamada.kawai,
     vertex.size = 5, vertex.color = 8, vertex.label.cex = 1)

# Make the network look better
V(TPAMI.graph)$label.cex <- 0.5*V(TPAMI.graph)$degree/max(V(TPAMI.graph)$degree)+1
V(TPAMI.graph)$label.color <- rgb(0, 0, 0.2, 0.8)
V(TPAMI.graph)$frame.color <- NA
egam <- 3*(log(E(TPAMI.graph)$weight+1))/max(log(E(TPAMI.graph)$weight+1))
E(TPAMI.graph)$color <- rgb(0.8, 0.5, 0)
E(TPAMI.graph)$width <- egam

# plot the graph in layout
plot(TPAMI.graph, layout=layout.kamada.kawai, vertex.size = 10, vertex.color = 7)

# Plot the communities
TPAMI.community <- walktrap.community(TPAMI.graph)
modularity(TPAMI.community)
membership(TPAMI.community)
plot(TPAMI.community, TPAMI.graph)
```

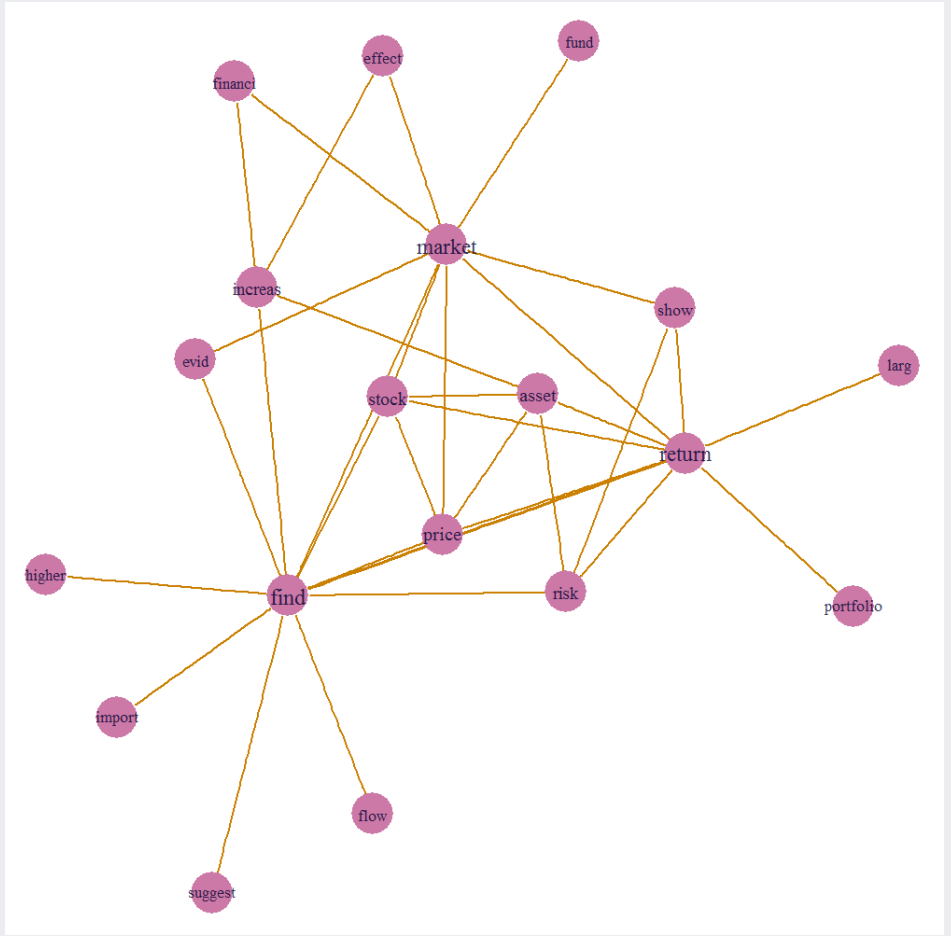
R Exercise: Keyword Network

- Keyword networks

TPAMI



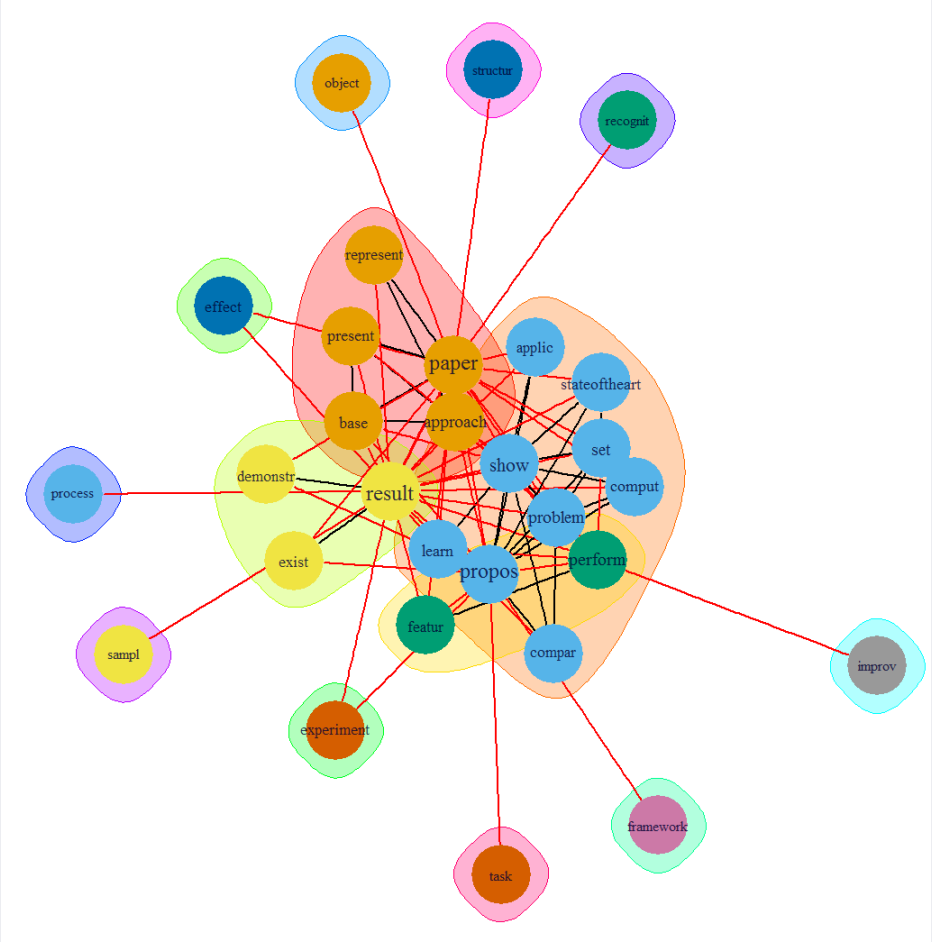
JoF



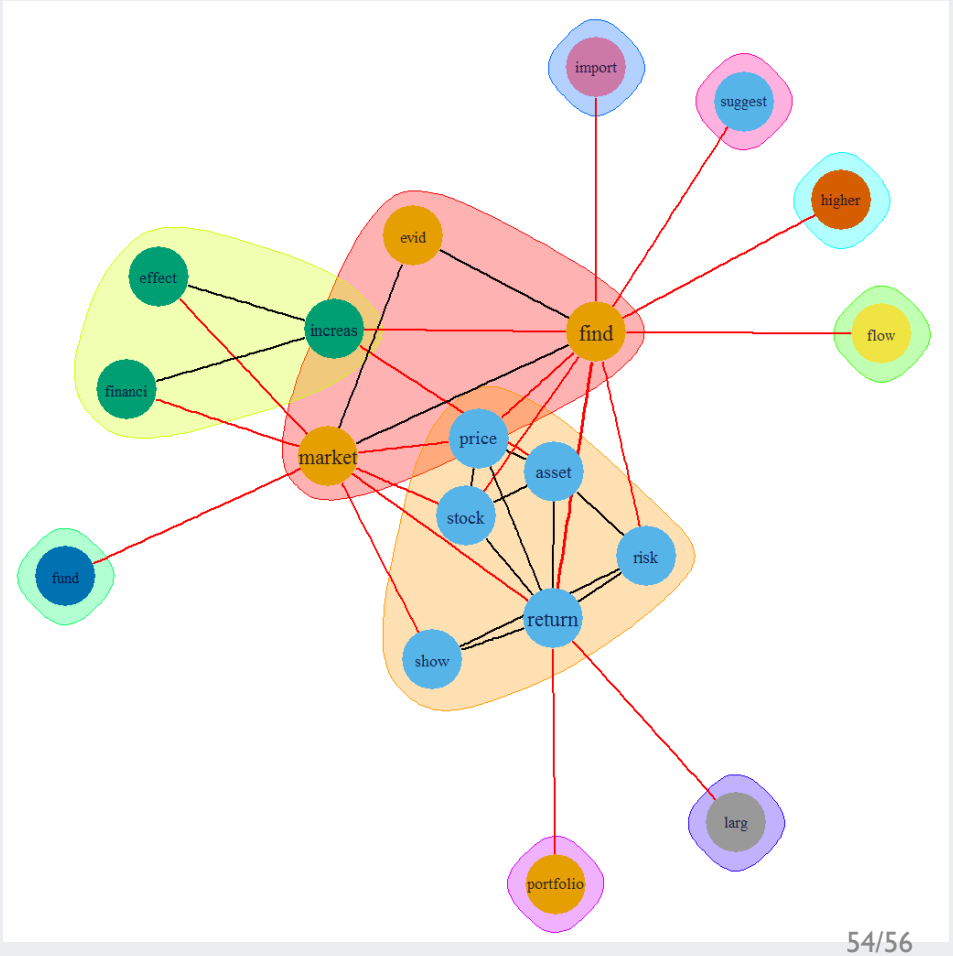
R Exercise: Keyword Network

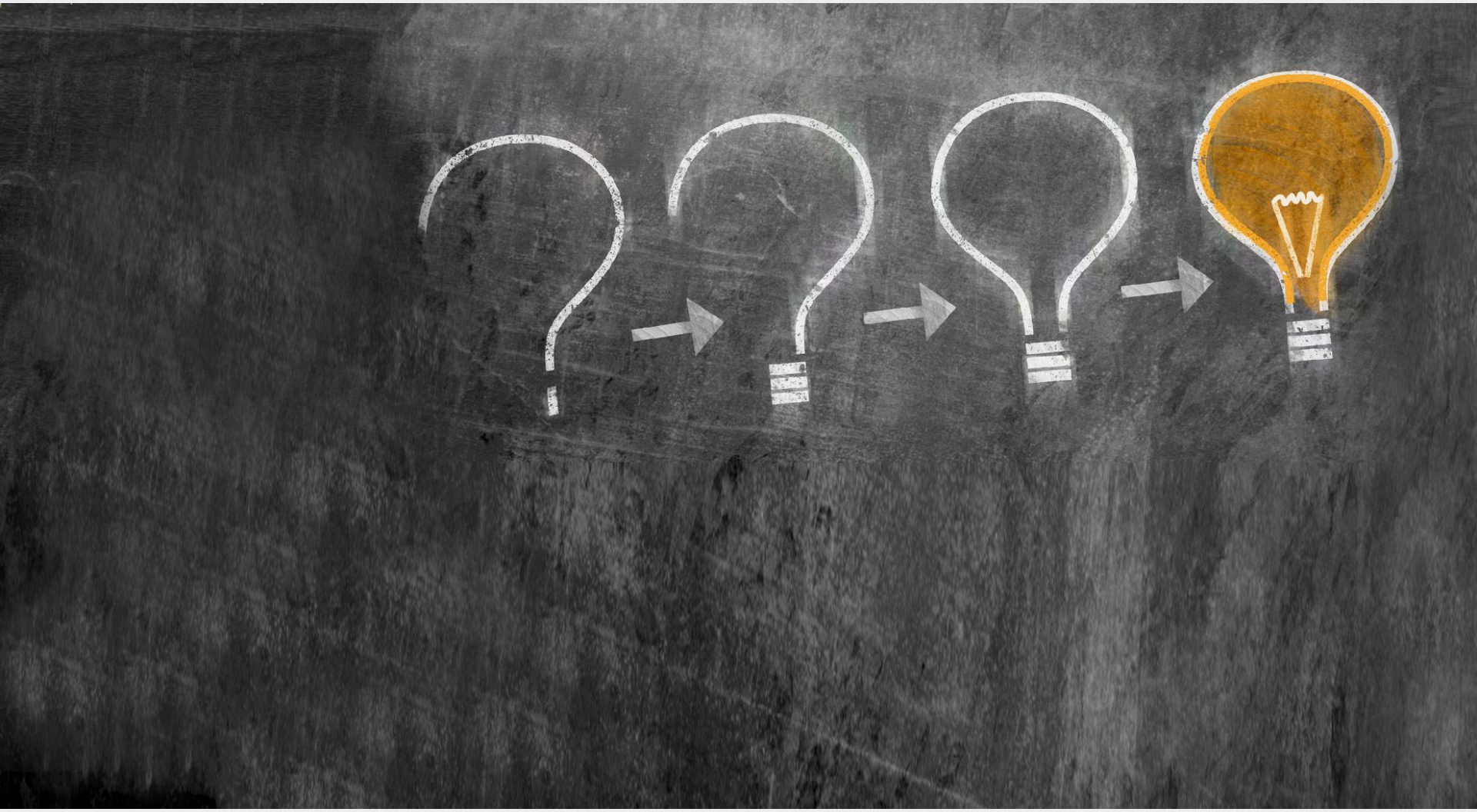
- Keyword communities

TPAMI



JoF





References

Research Papers

- Ahn, Y. Y., Ahnert, S. E., Bagrow, J. P., & Barabási, A. L. (2011). Flavor network and the principles of food pairing. Scientific reports, 1.
- Kim, J. & Kang, P. (2016+). Analyzing international collaboration and identifying core topics for the "Internet of Things" based on network analysis and topic modeling, under review.
- Kim, J., Park, M., Kim, H., Cho, S., Kang, P., Lee, D., Yang, K., & Kim, K. (2016). 이상치 탐지 기법을 활용한 내부자 위협 탐지 방법론 개발. 대한산업공학회 추계학술대회, 서울.
- Kim, H., Park, M., & Kang, P. (2016). 토픽모델링과 사회연경망을 통한 딥러닝 연구동향 분석. 대한산업공학회 춘계공동학술대회, 제주.
- Lee, H. & Kang, P. (2017+). Identifying core topics in technology and innovation management studies: A topic model approach. Journal of Technology Transfer. Online available https://link.springer.com/content/pdf/10.1007%2Fs10961-017-9561-4.pdf

Other Materials

- Kim, C., Kim, H., Cho, S., Kim, J., & Kang, P. (2017). 초음파 관련 임상연구 데이터의 텍스트마이닝 분석 플랫폼 개발, 산학과제 Granted by 삼성메디슨