Implications of Probabilistic Data Modeling for Rule Mining

Michael Hahsler, Kurt Hornik and Thomas Reutterer
Wirtschaftsuniversität Wien

29th Annual Conference of the German Classification Society (GfKl 2005)
Magdeburg, March 9-11, 2005

Motivation

- Mining association rules is an important technique for discovering meaningful patterns in transaction databases.
  - Example: `diapers` $\Rightarrow$ `beer`
  - Applications: product assortment decisions, adapting promotional activities, personalized product recommendations, adaptive user interfaces
- Current literature focuses on the properties of algorithms.
- We will discuss properties of
  - transaction data sets and
  - interest measures

from a probabilistic point of view.

Outline

1. Association rules

2. Probabilistic model for transaction data

3. Simulation with R

4. Implications for confidence and lift

5. New measure: hyperlift

6. Conclusion

Association Rules

An association rule is a rule of the form $X \Rightarrow Y$, where X and Y are two disjoint sets of items (itemsets).

Rule selection with threshold on interest measures:

- *Support:* fraction of transactions containing an itemset
- *Confidence:* probability of seeing Y under the condition that the transactions also contain X

Found rules are often ranked by:

- *Lift:* how many times more often X and Y occur together than expected if they where statistically independent

A simple probabilistic framework for transaction data

Transactions occur following a *Poisson process*

We analyze transactions which are recorded in a fixed time interval of length t.

The number of transactions m in the time interval is then poisson distributed with parameter θt:

$$P(M = m) = \frac{e^{-\theta t}(\theta t)^m}{m!} \tag{1}$$

A simple probabilistic framework (cont'd)

- n *independent* items $L = \{l_1, l_2, \dots, l_n\}$,
- with each having a *fixed success probabilities* to occur in a transaction given by the vector $p = (p_1, p_2, \dots, p_n)$.

Following the framework: c_i, the observed number of transactions item l_i is contained in, can be interpreted as a realization of a random variable C_i.

Under the condition of a fixed number of transactions m this random variable has a *binomial distribution:*

$$P(C_i = c_i | M = m) = \binom{m}{c_i} p_i^{c_i} (1 - p_i)^{m - c_i} \tag{2}$$

A simple probabilistic framework (cont'd)

Since for a fixed time interval t the number of transactions m is not fixed, the unconditional distribution gives:

$$
\begin{aligned}
P(C_i = c_i) &= \sum_{m=c_i}^{\infty} P(C_i = c_i | M = m) \cdot P(M = m) \\
&= \sum_{m=c_i}^{\infty} \binom{m}{c_i} p_i^{c_i} (1 - p_i)^{m-c_i} \, \frac{e^{-\theta t} (\theta t)^m}{m!} \\
&= \frac{e^{-\theta t} (p_i \theta t)^{c_i}}{c_i!} \sum_{m=c_i}^{\infty} \frac{((1-p)\theta t)^{m-c_i}}{(m - c_i)!} \\
&= \frac{e^{-p_i \theta t} (p_i \theta t)^{c_i}}{c_i!}
\end{aligned}
\tag{3}
$$

which has a *Poisson distribution* with parameter $\lambda_i = p_i \theta t$.

A simple probabilistic framework (cont'd)

Representation of transaction data as a binary incidence matrix:

items

transactions

	I_1	I_2	I_3	...	I_n
p	0.005	0.01	0.0003	...	0.025
Tr_1	0	1	0	...	1
Tr_2	0	1	0	...	1
Tr_3	0	1	0	...	0
Tr_4	0	0	0	...	0
.	.	.	.		.
.	.	.	.	.	.
.	.	.	.		.
Tr_{m-1}	1	0	0	...	1
Tr_m	0	0	1	...	1
c	99	201	7	...	411

Simulation

For simplicity we will assume for the following simulation that the parameters in λ are chosen from a single gamma distribution with parameters $k = 0.75$ and $a = 250$.

We will simulate the counts c_i, for $n = 200$ different items over a $t = 30$ day period with transaction intensity $\theta = 300$ transactions per day.

```
> m <- rpois(1, theta * t)
[1] 8885
> p <- sort(rgamma(n, shape = k, scale = a)/m,
+   decreasing = TRUE)
```

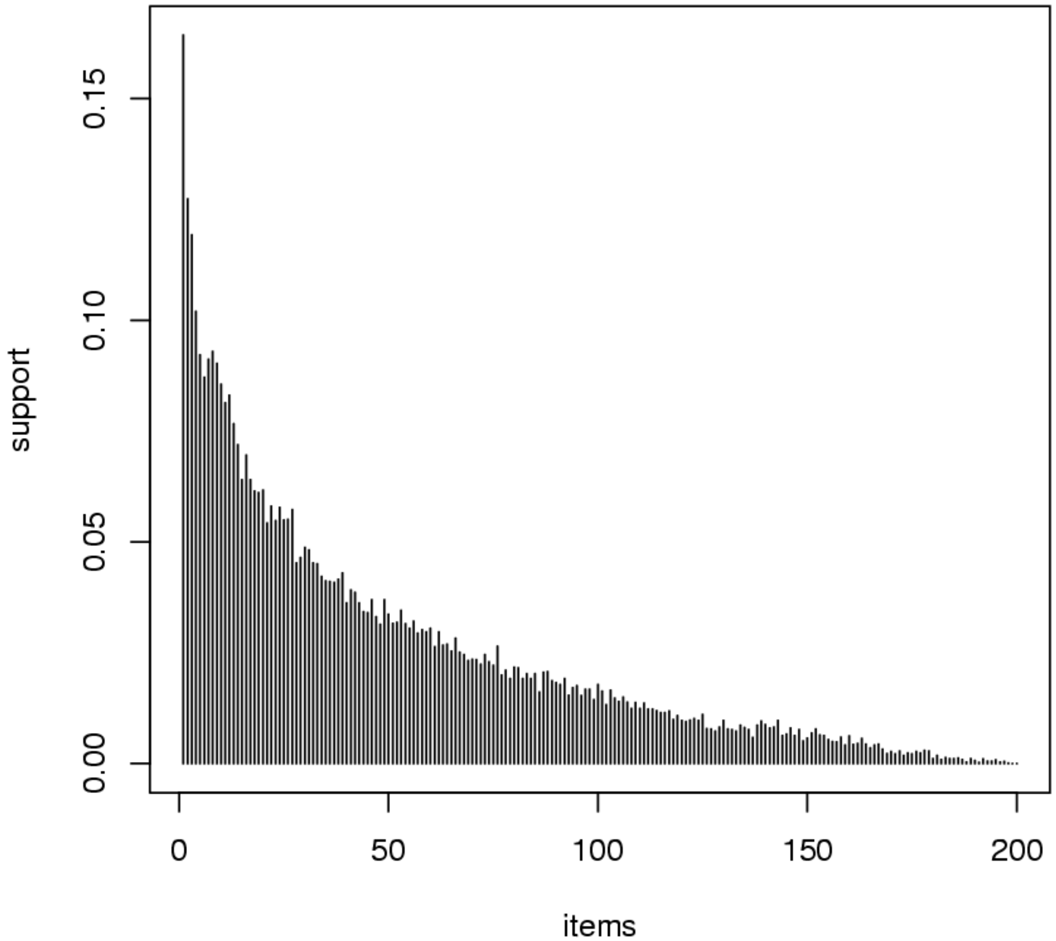
Now we can simulate the transactions in the database by m *Bernoulli trials* for each of the n items and calculate the count vector c.

```
> Tr <- matrix(rbinom(m * n, 1, p), ncol = n, byrow = TRUE)
> c <- (apply(Tr, 2, sum))
```

Simulation (cont'd)

We can directly calculate the *support* of each item from the transaction counts.

```
> supp1 <- c/m
> plot(supp1, type = "h", xlab = "items",
+  ylab = "support")
```


0.15
0.10
0.05
0.00
support
0
50
100
150
200
items

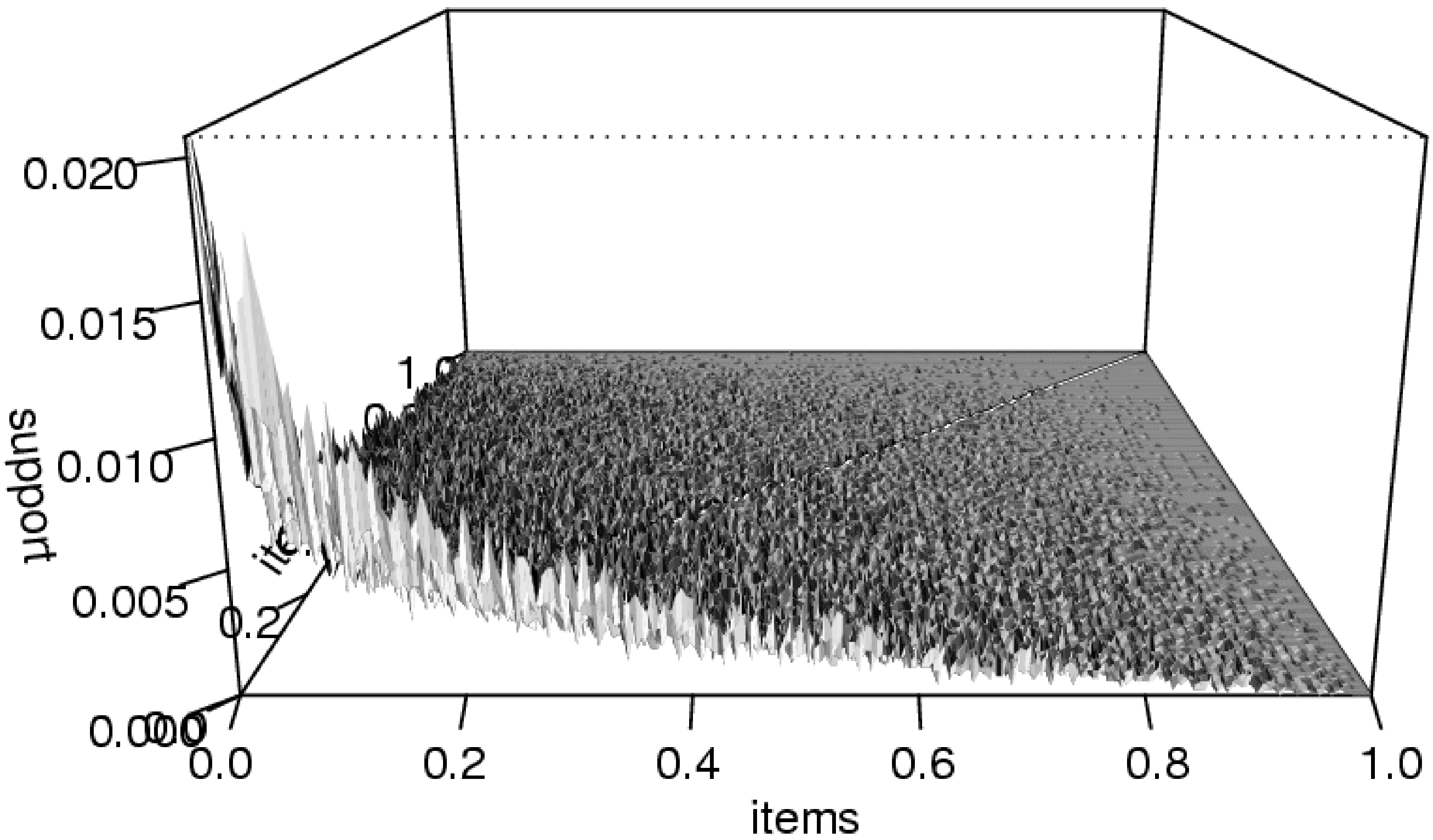
Simulation (cont'd)

Next, we extend the framework to the occurrences of 2-itemsets with a symmetric $n \times n$ count matrix c2 and a *support matrix* (supp2):

```
> c2 <- sapply(1:n, function(i) {
+     apply(Tr[, i] & Tr[, 1:n], 2, sum)})
> diag(c2) <- NA

> supp2 <- c2/m

> persp(supp2, expand = 0.5, ticktype = "detailed",
+  border = 0, shade = 1, zlab = "support",
+  xlab = "items", ylab = "items")
```


0.020
0.015
0.010
0.005
0.000
support
0.0
0.2
0.4
0.6
0.8
1.0
items

Implications for confidence

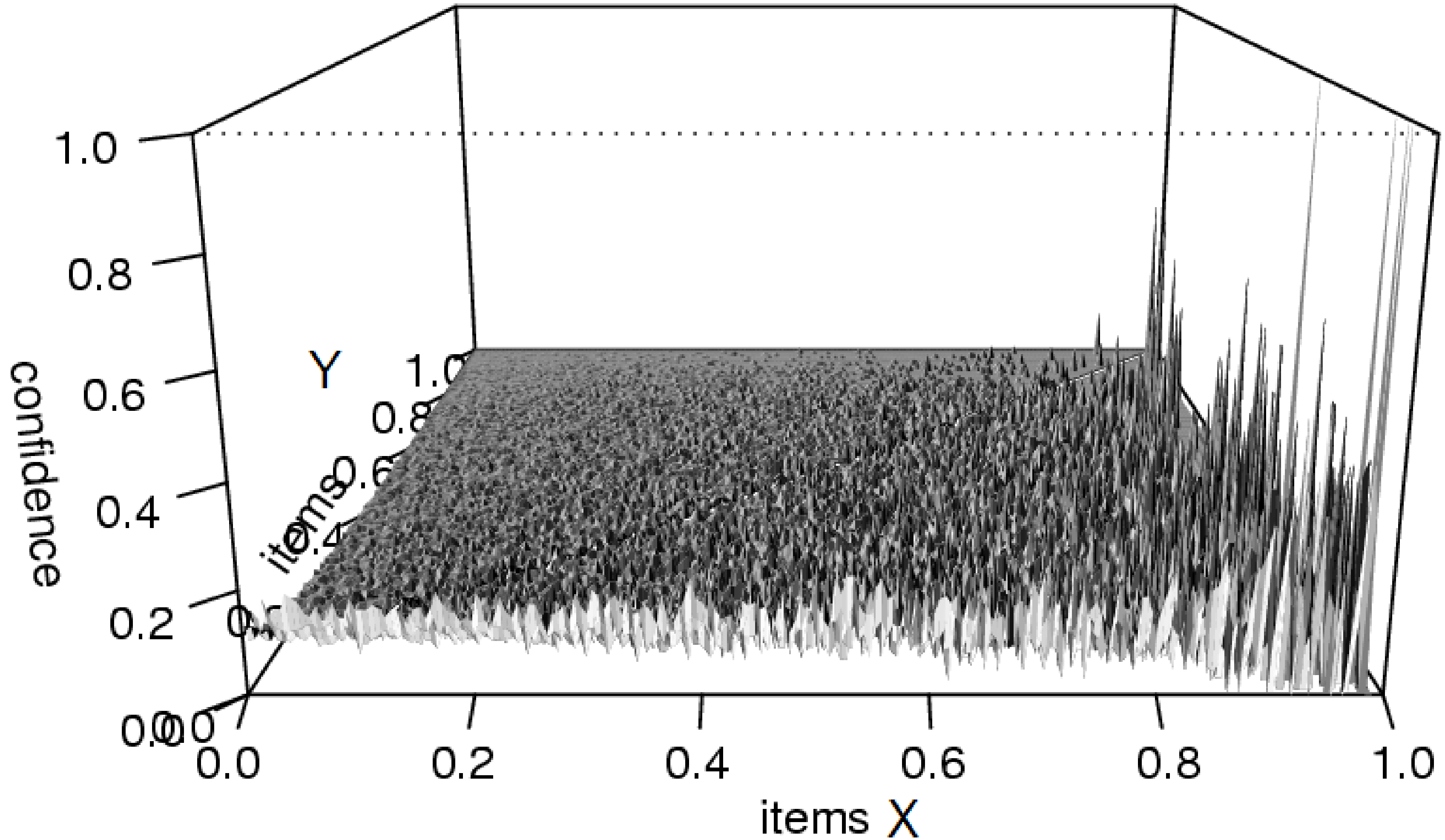
Confidence is defined by

$$\operatorname{conf}(X \Rightarrow Y) = \frac{\operatorname{supp}(X+Y)}{\operatorname{supp}(X)}. \tag{4}$$

From our 2-itemsets we can generate rules of the from $l_i \Rightarrow l_j$, where $i, j = 1, 2, \ldots, n$ and $i \neq j$. We calculate confidence for the $n(n-1)$ possible rules in the data set.

```
> conf2 <- supp2/supp1
```

```
> persp(conf2, expand = 0.5, ticktype = "detailed",
+    border = 0, shade = 1, zlab = "confidence",
+    xlab = "items", ylab = "items")
```



1.0
0.8
0.6
0.4
0.2
0.0
confidence
Y
items
0.0
0.2
0.4
0.6
0.8
1.0
items X

Implications for confidence (cont'd)

- Confidence values are generally very low which reflect the fact that there are no associations in the data.

- Some rules with confidence of one. However, left-hand-sides (X) have low support.

- Confidence increases with the item in the right-hand-side Y of the rule getting more frequent.

The fact that *confidence systematically favors some rules* makes the measure problematic when it comes to ranking rules.

Implications for lift

Typically, rules mined using minimum support (and confidence) are filtered or ordered using their lift value. The measure lift is defined as:

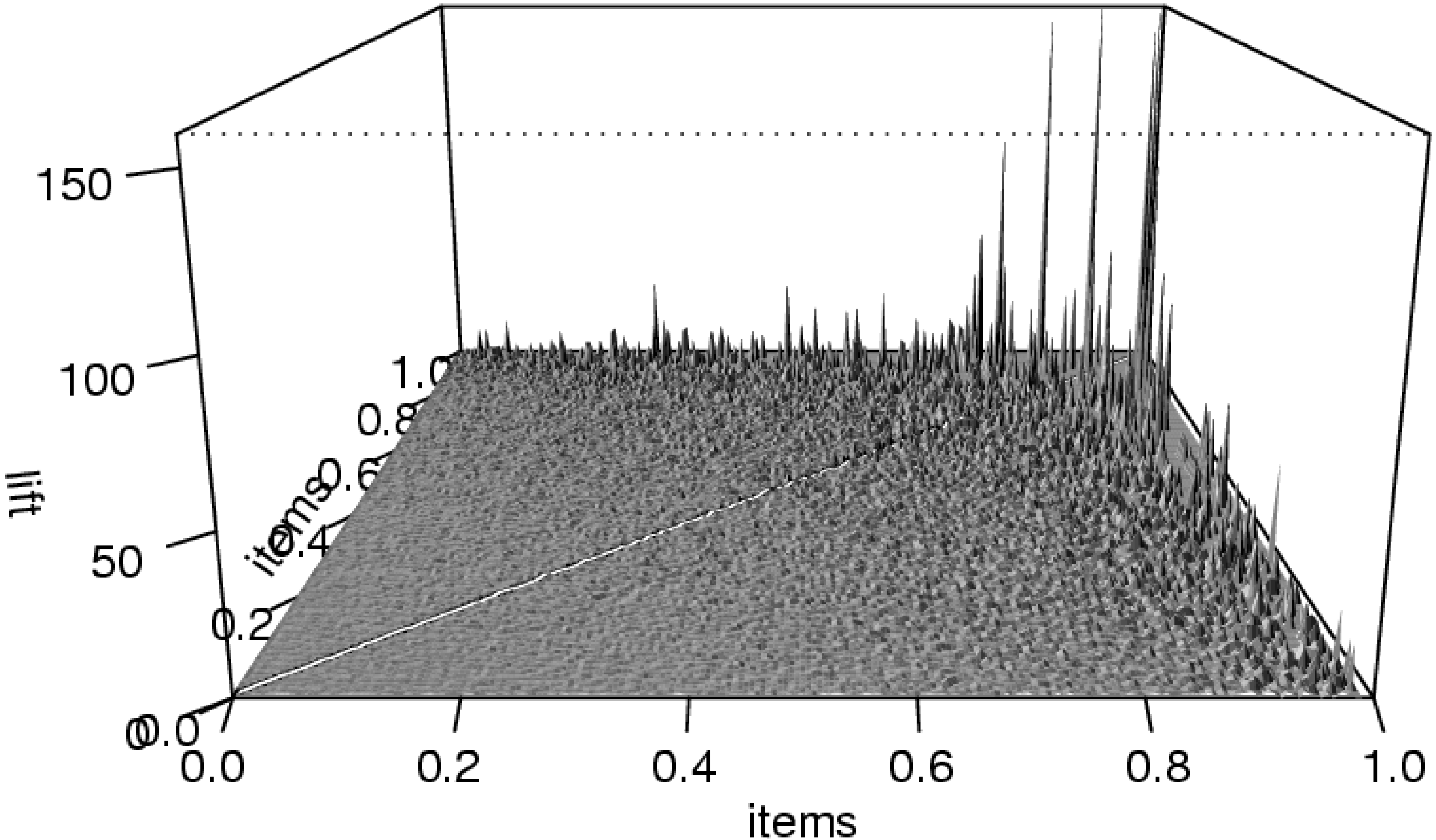
$$\text{lift}(X \Rightarrow Y) = \frac{\text{conf}(X \Rightarrow Y)}{\text{supp}(Y)} \tag{5}$$

A lift value close to 1 indicates that the items are co-occurring in the database as expected under independence.

```
> lift <- conf2/matrix(supp1, ncol = n, nrow = n,
+   byrow = TRUE)

> persp(lift, expand = 0.5, ticktype = "detailed",
+  border = 0, shade = 1, zlab = "lift",
+  xlab = "items", ylab = "items")

> length(which(lift > 2))
[1] 3424
```


150
100
50
0
lift
0.0
0.2
0.4
0.6
0.8
1.0
items
0.0
0.2
0.4
0.6
0.8
1.0
items

Implications for lift (cont'd)

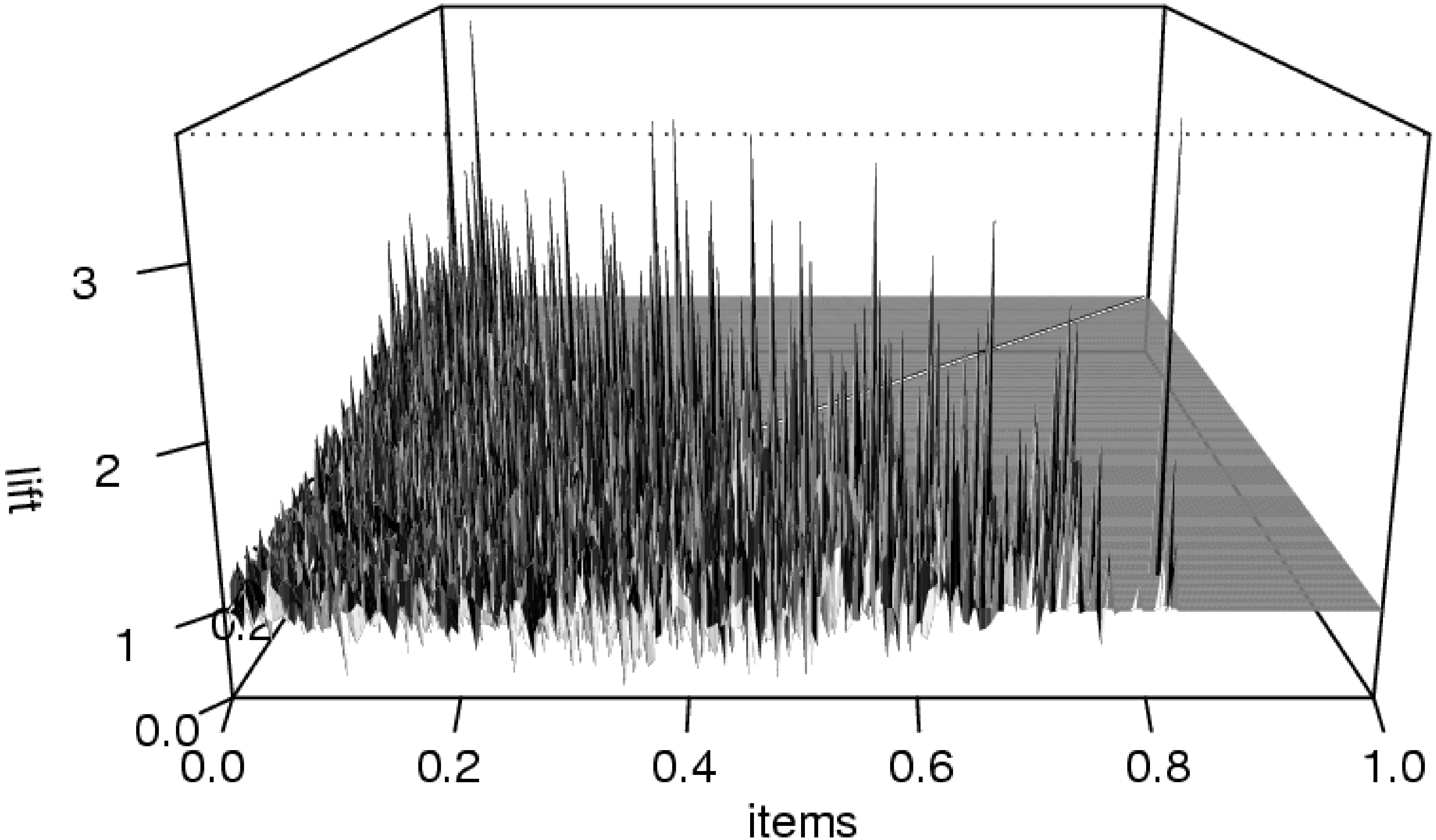
To counter the problem with extremely high lift values, we discard all 2-itemsets which do not satisfy a minimum support of 0.1%.

```
> min_supp <- 0.001
> length(lift[supp2 >= min_supp])
[1] 7096

> lift[supp2 < min_supp] <- 1

> persp(lift, expand = 0.5, ticktype = "detailed",
+  border = 0, shade = 1, zlab = "lift",
+  xlab = "items", ylab = "items")

> length(which(lift > 2))
[1] 130
```


3
2
1
lift
0.0
0.2
0.0
0.2
0.4
0.6
0.8
1.0
items

Implications for lift (cont'd)

- Lift performs poorly to filter random noise in transaction data especially if for relatively rare items.

- Lift has a tendency to produce higher values for rules with items close to minimum support.

This makes using lift *problematic for ranking* discovered rules.

New measure: hyperlift

- The $n \times n$ co-occurrence matrix can be modeled by n^2 random variables $C_{i,j}$.

- The framework results in hypergeometric distributions for the $C_{i,j}$s (urn model).

- Using the expected value of $C_{i,j}$ lift can be rewritten as:

$$\mathrm{lift}(l_i \Rightarrow l_j) = \frac{P(l_i + l_j)}{P(l_i)P(l_j)} = \frac{c_{i,j}}{E[C_{i,j}]} \tag{6}$$

- As a more conservative approach we use quantile $Q_\delta[C_{i,j}]$ instead of the expected value.

$$\mathrm{hyperlift}(l_i \Rightarrow l_j) = \frac{c_{i,j}}{Q_\delta[C_{i,j}]}. \tag{7}$$

New measure: hyperlift (cont'd)

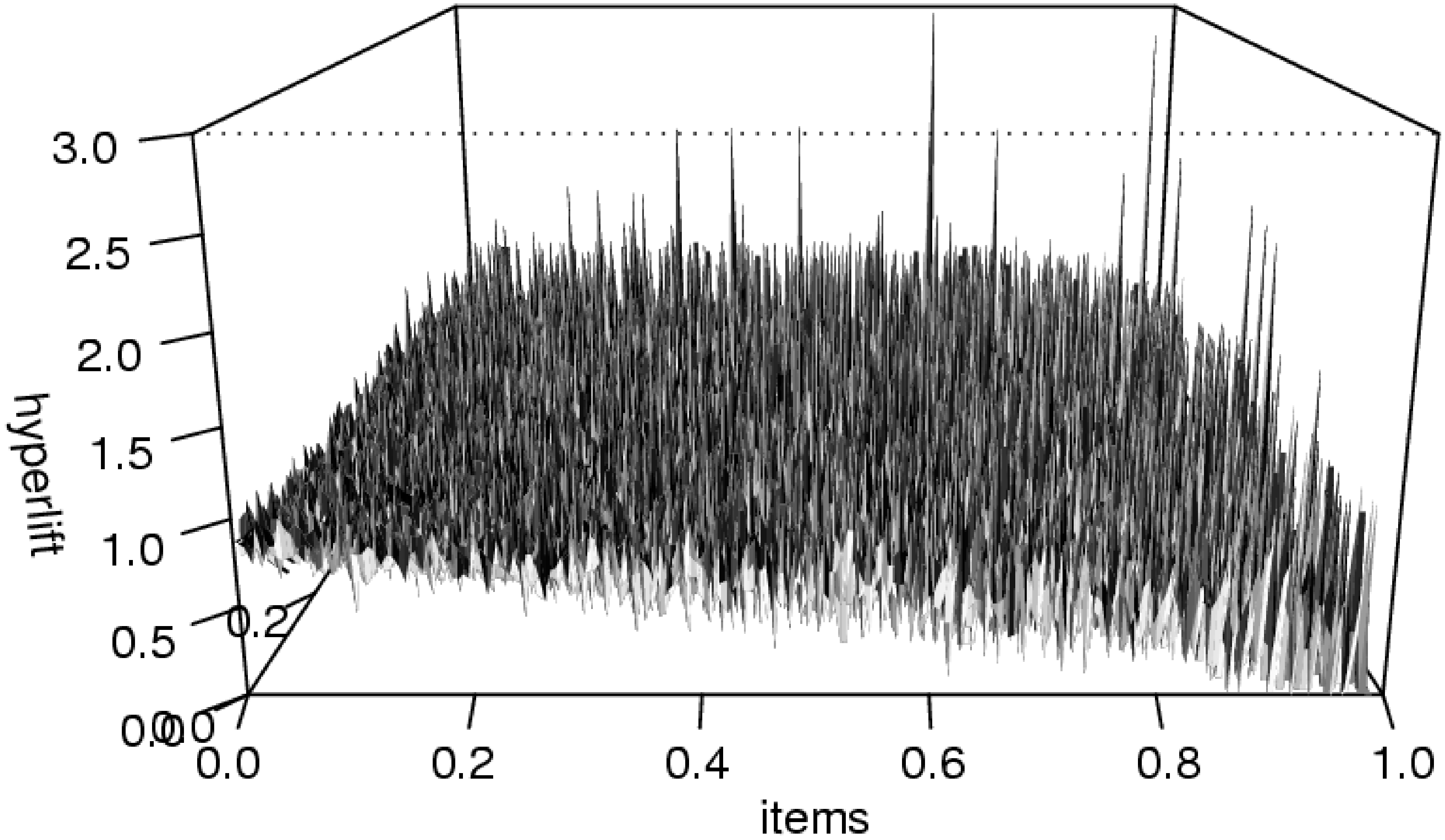
Calculating hyperlift for $\delta = 0.99$:

```
> calc_hyperbase <- function(ci, cj) {
+     qhyper(0.99, m = cj, n = m - cj, k = ci)}

> hyperlift <- c2/outer(c, c, FUN = calc_hyperbase)
> hyperlift[is.infinite(hyperlift)] <- NA

> persp(hyperlift, shade = 1, ticktype = "detailed",
+   border = 0, expand = 0.5, zlab = "hyperlift",
+   xlab = "items", ylab = "items")

> length(which(hyperlift > 2))
[1] 2
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
hyperlift
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
items

New measure: hyperlift (cont'd)

- Generally smaller than 1 and *more evenly distributed* than lift. Indicates that hyperlift filters the random co-occurrences better than lift.

- Hyperlift *shows a weak systematic dependency* to favor rules with more frequent items.

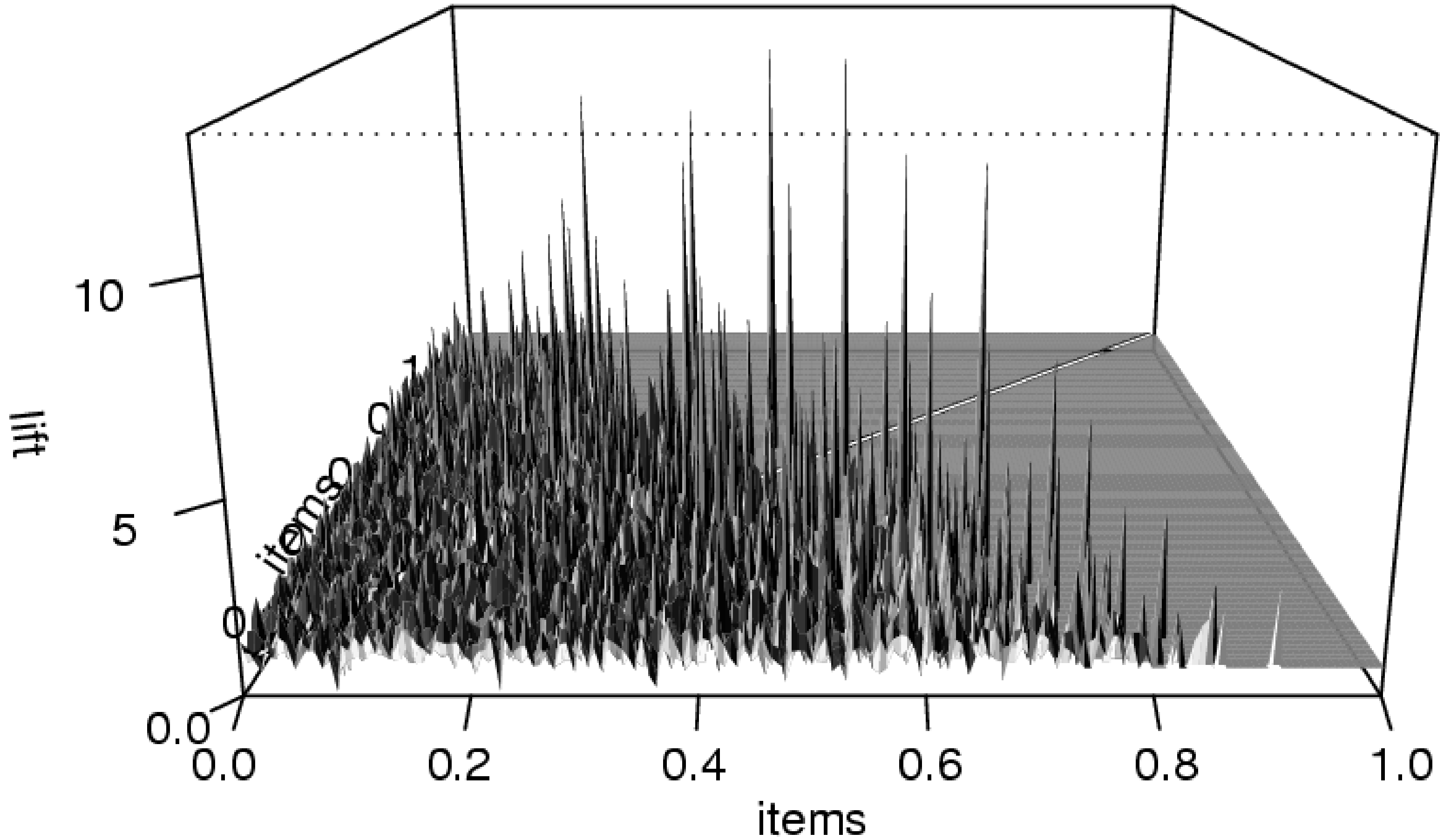
Comparing lift and hyperlift on a grocery database

- 1 month of real-world point-of-sale transaction data from a local grocery outlet with

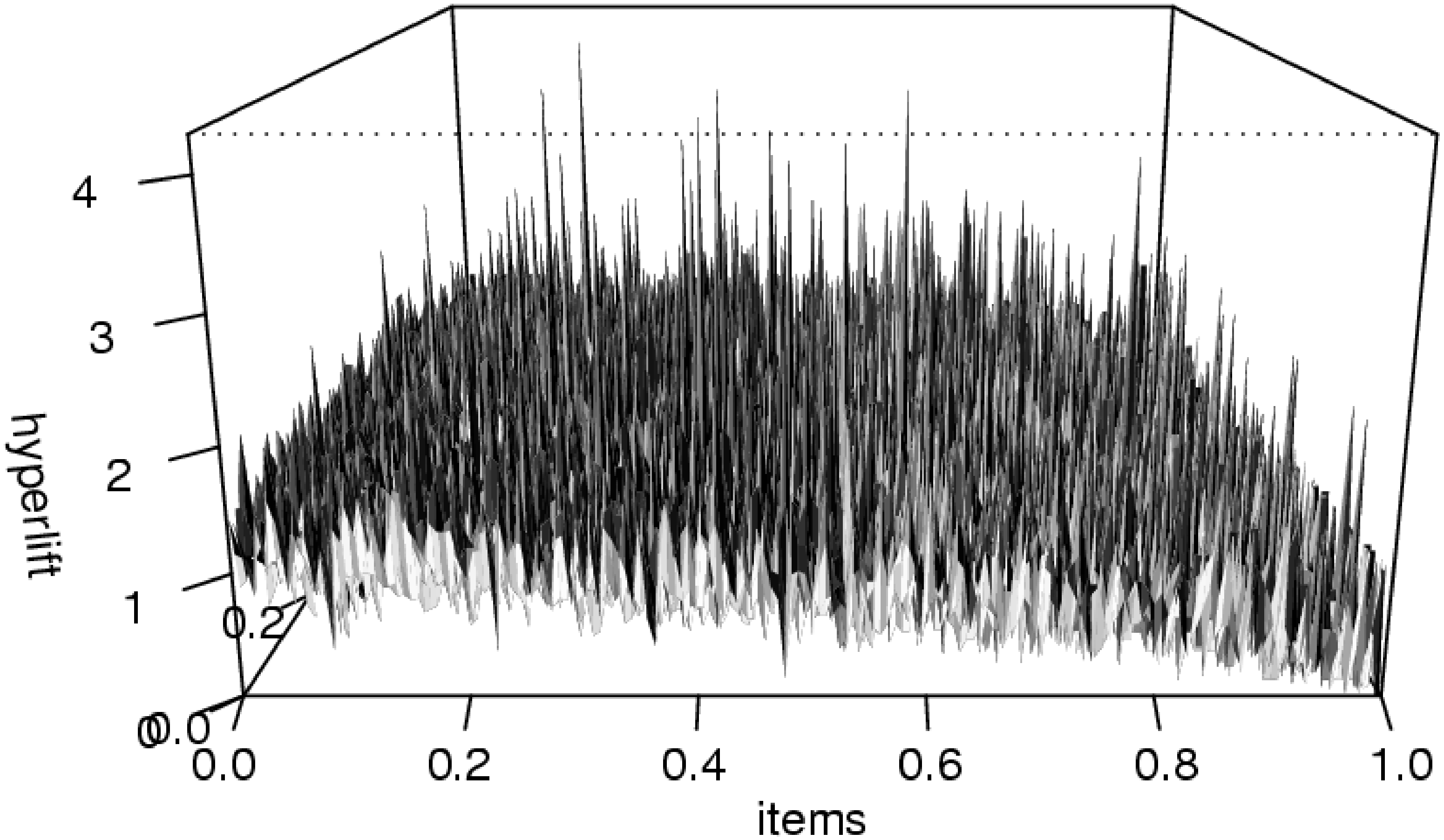
- $m = 9835$ transaction and

- $n = 169$ categories.

- Support, confidence and lift distributions look almost identical to the simulated data.



Lift for 2-itemsets for items with support of 0.1% in the grocery database



Hyperlift for 2-itemsets for items in the grocery database

Comparing lift and hyperlift (cont'd)

Top 10 rules (ordered by lift, support = 0.001)

	l_i	l_j	supp	lift
20	mayonnaise	mustard	0.001423	12.965
8	Instant food products	hamburger meat	0.003050	11.421
15	softener	detergent	0.001118	10.600
16	liquor	red/blush wine	0.002135	10.025
6	flour	sugar	0.004982	8.463
4	popcorn	salty snack	0.002237	8.192
11	processed cheese	ham	0.003050	7.071
9	sauces	hamburger meat	0.001220	6.684
3	meat spreads	cream cheese	0.001118	6.605
14	house keeping products	detergent	0.001017	6.346

Comparing lift and hyperlift (cont'd)

Top 10 rules (ordered by hyperlift, no support)

	l_i	l_j	supp	hyperlift	lift
11	Instant food products	hamburger meat	0.0030	4.286	11.421
9	flour	sugar	0.0049	4.083	8.463
15	liquor	red/blush wine	0.0021	3.500	10.025
* 17	cooking chocolate	baking powder	0.0007	3.500	15.826
18	mayonnaise	mustard	0.0014	3.500	12.965
6	processed cheese	white bread	0.0041	3.154	5.975
7	popcorn	salty snack	0.0022	3.143	8.192
13	processed cheese	ham	0.0030	3.000	7.071
3	liquor	bottled beer	0.0046	2.875	5.241
14	softener	detergent	0.0011	2.750	10.600
8	baking powder	sugar	0.0032	2.667	5.432

Comparing lift and hyperlift (cont'd)

- All rules for lift (with support) and hyperlift make intuitively sense.
- Rules with high hyperlift have potentially also high lift.
- Hyperlift selects rules with support varying from very rare to relatively frequent (the tendency of hyperlift to favors rules with more frequent items seems not too strong).
- Hyperlift is also able to deal with very infrequent rules.

Conclusion

- Interest measures are systematically influenced by the frequencies of items in the corresponding itemsets or rules.
- Lift performs poorly to filter random noise.
- The presented framework provides many possibilities for further research:
  - Adapt hyperlift to finding substitutes (instead of complements).
  - Analyze systematic influence of the occurrence frequency of items on the hyperlift measure.
  - Use p-value instead of hyperlift.
  - Expand model to itemsets of size > 2.
  - Model dependencies between items.