Quiz 4

- There are 6 problems total worth 35 points as shown in each question.
- You must not communicate with other students during this test.
- No books, notes allowed.
- No other electronic device except calculators are allowed. You cannot use your mobile as calculators.
- This is a 45 minute exam.
- Do not turn this page until instructed to.
- There are several different versions of this exam.

1. Fill in your information:

Full Name: _____

NetID: _____

Zone 1

1/1. (5 points) Why is the runtime comparable for large support in the Apriori and FP-growth algorithms?

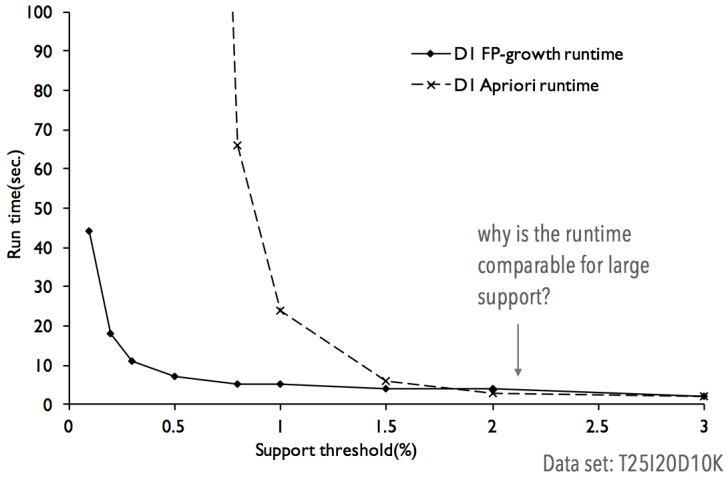

Figure 1: FP-growth v.s. Apriori: scalability with the support threshold.

Solution. When the support is raised, even at a small number of percentage, the number of candidates generated shrinks exponentially. FP-growth works better than Apriori only under the condition that the number of candicates generated grows exponentially. When the support is raised, this condition is removed. So there is no difference between FP-growth and Apriori performance for large support.

1/2. (5 points) Why is the runtime comparable for large support in the Apriori and FP-growth algorithms?

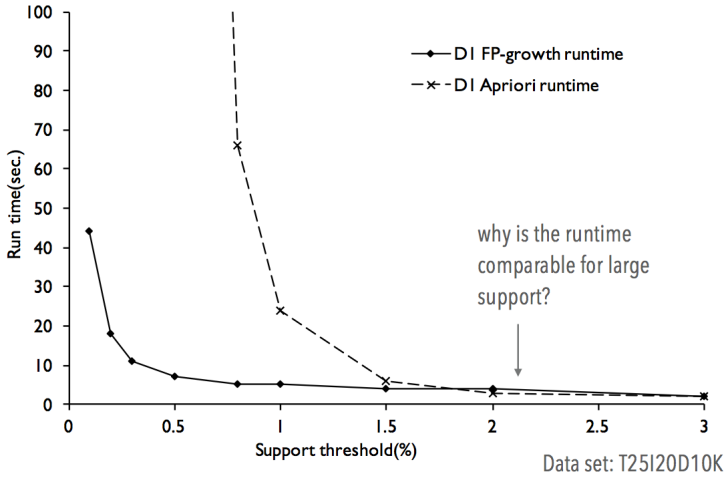

Figure 2: FP-growth v.s. Apriori: scalability with the support threshold.

Solution. When the support is raised, even at a small number of percentage, the number of candidates generated shrinks exponentially. FP-growth works better than Apriori only under the condition that the number of candicates generated grows exponentially. When the support is raised, this condition is removed. So there is no difference between FP-growth and Apriori performance for large support.

2/1. (5 points) Below is the contingency table that summarizes a supermarket transaction data of milk and cereal sell.

	$Cereal$	$\overline{Cereal}$	$\sum_{col}$
$Milk$	800	90	890
$\overline{Milk}$	100	10	110
$\sum_{row}$	900	100	1000

(a) (3 points) Calculate two measures of correlation: *Lift* and *Kulczynski*.
(Hint: $Lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$, $Kulc(A,B) = \frac{1}{2}(P(A|B) + P(B|A))$.)

(b) (1 point) What kind of correlation exists between milk and cereal sell? Is there a positive, negative or no correlation?

(c) (1 point) What is the main difference between the *Lift* and *Kulczynski* measures?

Solution.

(a)

$$Lift = (800/1000)/((900/1000) \times (890/1000)) = 0.9988,$$
$$Kulczynski = 0.5 \times ((800/890) + (800/900)) = 0.8939.$$

(b) Positive.

(c) *Kulczynski* is null-invariant.

2/2. (5 points) Below is the contingency table that summarizes a supermarket transaction data of milk and cereal sell.

	$Cereal$	$\overline{Cereal}$	$\sum_{col}$
$Milk$	500	150	650
$\overline{Milk}$	300	50	350
$\sum_{row}$	800	200	1000

(a) (3 points) Calculate two measures of correlation: *Lift* and *Kulczynski*.
(Hint: $Lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$, $Kulc(A,B) = \frac{1}{2}(P(A|B) + P(B|A))$.)

(b) (1 point) What kind of correlation exists between milk and cereal sell? Is there a positive, negative or no correlation?

(c) (1 point) What is the main difference between the *Lift* and *Kulczynski* measures?

Solution.

(a)

$$Lift = (500/1000)/((650/1000) \times (800/1000)) = 0.9615,$$
$$Kulczynski = 0.5 \times ((500/650) + (500/800)) = 0.6971.$$

(b) Positive.

(c) *Kulczynski* is null-invariant.

2/3. (5 points) Below is the contingency table that summarizes a supermarket transaction data of milk and cereal sell.

	$Cereal$	$\overline{Cereal}$	$\sum_{col}$
$Milk$	80	100	180
$\overline{Milk}$	120	700	820
$\sum_{row}$	200	800	1000

(a) (3 points) Calculate two measures of correlation: *Lift* and *Kulczynski*.
(Hint: $Lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$, $\quad Kulc(A,B) = \frac{1}{2}(P(A|B) + P(B|A))$.)

(b) (1 point) What kind of correlation exists between milk and cereal sell? Is there a positive, negative or no correlation?

(c) (1 point) What is the main difference between the *Lift* and *Kulczynski* measures?

Solution.

(a)

$$Lift = (80/1000)/((180/1000) \times (200/1000)) = 2.2222,$$
$$Kulczynski = 0.5 \times ((80/180) + (80/200)) = 0.4222.$$

(b) Negative.

(c) *Kulczynski* is null-invariant.

2/4. (5 points) Below is the contingency table that summarizes a supermarket transaction data of milk and cereal sell.

	$Cereal$	$\overline{Cereal}$	$\sum_{col}$
$Milk$	50	100	150
$\overline{Milk}$	200	650	850
$\sum_{row}$	250	750	1000

(a) (3 points) Calculate two measures of correlation: *Lift* and *Kulczynski*.
(Hint: $Lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$, $\quad Kulc(A,B) = \frac{1}{2}(P(A|B) + P(B|A))$.)

(b) (1 point) What kind of correlation exists between milk and cereal sell? Is there a positive, negative or no correlation?

(c) (1 point) What is the main difference between the *Lift* and *Kulczynski* measures?

Solution.

(a)

$$Lift = (50/1000)/((150/1000) \times (250/1000)) = 1.3333,$$
$$Kulczynski = 0.5 \times ((50/150) + (50/250)) = 0.2667.$$

(b) Negative.

(c) *Kulczynski* is null-invariant.

3/1. (5 points) Why is it hard to push data anti-monotonicity deep into the Aprori algorithm? Please explain with a simple example of data anti-monotonic constrain.

Solution. In Apriori based algorithm, we are doing level based join. Each item set is considered in multiple set joins. The pattern that we are trying to reject from one join which violates the constrain could be part of another join that satisfies the constrain. So we cannot easily prune the pattern completely. Data anti-monotonicity can only be used for pattern growth based algorithms.

Examples, $sum(S.price) \geq v$, $min(S.price) \leq v$, $range(S.price) \geq v$.

3/2. (5 points) Why is it hard to push data anti-monotonicity deep into the Aprori algorithm? Please explain with a simple example of data anti-monotonic constrain.

Solution. In Apriori based algorithm, we are doing level based join. Each item set is considered in multiple set joins. The pattern that we are trying to reject from one join which violates the constrain could be part of another join that satisfies the constrain. So we cannot easily prune the pattern completely. Data anti-monotonicity can only be used for pattern growth based algorithms.

Examples, $sum(S.price) \geq v$, $min(S.price) \leq v$, $range(S.price) \geq v$.

4/1. (5 points) Below are some transaction patterns with their supports from a transaction dataset. Identify all the core patterns ($\boldsymbol{\tau = 0.5}$) for each given transaction pattern. Please show your calculation steps briefly.

Transaction pattern	Support (number of transactions)
ab	100
abc	90
abe	100
bcde	80

Solution.

Transaction pattern	Support (number of transactions)
ab	
abc	c, ac, bc
abe	e, ae, be
bcde	d, bd, cd, ce, de, bcd, bce, bde, cde

4/2. (5 points) Below are some transaction patterns with their supports from a transaction dataset. Identify all the core patterns ($\boldsymbol{\tau = 0.3}$) for each given transaction pattern. Please show your calculation steps briefly.

Transaction pattern	Support (number of transactions)
ab	100
abc	90
abe	100
bcde	80

Solution.

Transaction pattern	Support (number of transactions)
ab	a
abc	a, c, ab, ac, bc
abe	a, e, ab, ae, be
bcde	c, d, e, bc, bd, be, cd, ce, de, bcd, bce, bde, cde

4/3. (5 points) Below are some transaction patterns with their supports from a transaction dataset. Identify all the core patterns ($\boldsymbol{\tau = 0.5}$) for each given transaction pattern. Please show your calculation steps briefly.

Transaction pattern	Support (number of transactions)
ad	100
abc	80
bd	100
bcde	90

Solution.

Transaction pattern	Support (number of transactions)
ad	a
abc	ab, ac
bd	
bcde	c, e, bc, be, cd, ce, de, bcd, bce, bde, cde

4/4. (5 points) Below are some transaction patterns with their supports from a transaction dataset. Identify all the core patterns ($\boldsymbol{\tau = 0.3}$) for each given transaction pattern. Please show your calculation steps briefly.

Transaction pattern	Support (number of transactions)
ad	100
abc	80
bd	100
bcde	90

Solution.

Transaction pattern	Support (number of transactions)
ad	a, d
abc	a, c, ab, ac, bc
bd	b, d
bcde	b, c, d, e, bc, bd, be, cd, ce, de, bcd, bce, bde, cde

5/1. (5 points) Below is a sequence dataset containing transaction sequences. Note that (bc) means items b and c are purchased at the same time. Use the Generalized Sequential Patterns (GSP) mining algorithm to find all the frequent sequential patterns with **min_sup = 3**. Please show your calculation steps briefly.

SID	Sequence
S1	ab(cd)e
S2	a(bc)bd
S3	(bc)(bd)
S4	bcd

Solution. b:4, c:4, d:4, bd:4, cd:3.

5/2. (5 points) Below is a sequence dataset containing transaction sequences. Note that (bc) means items b and c are purchased at the same time. Use the Generalized Sequential Patterns (GSP) mining algorithm to find all the frequent sequential patterns with **min_sup = 4**. Please show your calculation steps briefly.

SID	Sequence
S1	ab(cd)e
S2	a(bc)bd
S3	(bc)(bd)
S4	bcd

Solution. b:4, c:4, d:4, bd:4.

5/3. (5 points) Below is a sequence dataset containing transaction sequences. Note that (bc) means items b and c are purchased at the same time. Use the Generalized Sequential Patterns (GSP) mining algorithm to find all the frequent sequential patterns with **min_sup = 3**. Please show your calculation steps briefly.

SID	Sequence
S1	(bc)de
S2	a(bc)bd
S3	(bc)(bd)
S4	bd

Solution. b:4, c:3, d:4, bd:4, cd:3, (bc):3.

5/4. (5 points) Below is a sequence dataset containing transaction sequences. Note that (bc) means items b and c are purchased at the same time. Use the Generalized Sequential Patterns (GSP) mining algorithm to find all the frequent sequential patterns with **min_sup = 4**. Please show your calculation steps briefly.

SID	Sequence
S1	(bc)de
S2	a(bc)bd
S3	(bc)(bd)
S4	bd

Solution. b:4, d:4, bd:4.

6/1. (10 points) Below is a transaction dataset and item prices of each transaction. Assume each kind of item has one distinct positive price. Use the Apriori algorithm to find all the frequent patterns that satisfy the requirements **min_sup = 2** and **sum{S.price} < 5**.

TID	Items (Price)
I1	1, 2, 4
I2	2, 3, 5
I3	1, 2, 3, 5
I4	2, 5

Please write out each of the candidate and pattern generation steps and mark those patterns that are either pruned or unnecessarily computed.

Solution.

C1	sup
1	2
2	4
3	2
4	1 (prune)
5	3

L1	sup
1	2
2	4
3	2
5	3 (unnec)

C2	sup
1, 2	2
1, 3	1 (prune)
1, 5	1 (prune)
2, 3	2
2, 5	3
3, 5	2

L2	sup
1, 2	2
2, 3	2 (unnec)
2, 5	3 (unnec)
3, 5	2 (unnec)

C3	sup
1, 2, 3	1 (prune)
1, 2, 5	1 (prune)
2, 3, 5	2

L3	sup
2, 3, 5	2 (unnec)

6/2. (10 points) Below is a transaction dataset and item prices of each transaction. Assume each kind of item has one distinct positive price. Use the Apriori algorithm to find all the frequent patterns that satisfy the requirements **min_sup = 2** and **sum{S.price} < 5**.

TID	Items (Price)
I1	1, 2, 3, 5
I2	2, 4
I3	1, 2, 4, 5
I4	2, 5

Please write out each of the candidate and pattern generation steps and mark those patterns that are either pruned or unnecessarily computed.

Solution.

C1	sup
1	2
2	4
3	1 (prune)
4	2
5	3

L1	sup
1	2
2	4
4	2
5	3 (unnec)

C2	sup
1, 2	2
1, 4	1 (prune)
1, 5	2
2, 4	2
2, 5	3
4, 5	1 (prune)

L2	sup
1, 2	2
1, 5	2 (unnec)
2, 4	2 (unnec)
2, 5	3 (unnec)

C3	sup
1, 2, 4	1 (prune)
1, 2, 5	2
2, 4, 5	1 (prune)

L3	sup
1, 2, 5	2 (unnec)
