Problem Description

Objective

- *review set selection* problem where given a set of reviews for a specific item, we want to select a comprehensive subset of small size
- comprehensiveness is defined with respect to the attributes of the product and the viewpoints of the reviews.

Results

- top quality reviews

Dataset details

- Source : http://jmcauley.ucsd.edu/data/amazon/links.html
- Musical Instruments reviews dataset with a total of 500,176 reviews
- Sample review :
  - **reviewerID** - ID of the reviewer, e.g. A2SUAM1J3GNN3B
  - **asin** - ID of the product, e.g. 0000013714
  - **reviewerName** - name of the reviewer
  - **helpful** - helpfulness rating of the review, e.g. 2/3
  - **reviewText** - text of the review
  - **overall** - rating of the product
  - **summary** - summary of the review
  - **unixReviewTime** - time of the review (unix time)
  - **reviewTime** - time of the review (raw)

Data Preprocessing

- Removal of columns which are not required; useful columns: **asin**, **reviewText**, **helpful** and **overall.**
- Prune items for which reviews are less than 20 or total number of votes are less than 10.
- Quality score calculated from helpful column
- Total number of reviews after preprocessing: **23972**
- Extraction of **Attributes:**
  - Performed Sentence-wise POS tagging to obtain noun and noun phrase
  - Stop word removal (using mallet stopwords)
  - Frequent Item set mining using apriori algorithm with min support 4.2% of total number of reviews after preprocessing

Data Preprocessing

- Removed duplicate attributes among same sentence
- Some attributes:
  - [['performance'], ['instrument'], ['amp'], ['band'], ['line'], ['pedal'], ['part'], ['bass'], ['tune'], ['year'], ['software'],, ['tuner'], ['model'], ['quality'], ['noise'], ['power], ['price'], ['input'], ['battery'], ['microphone'].....]
- Total number of attributes extracted: **101**

Problem (Coverage(f))

- **coverage scoring function *f(S, a):*** It assigns a score to an attribute a given the subset ***S***.
- Given a set of attributes ***A*** and a set of reviews ***R*** for an item ***x***, an integer budget value ***k***, and a coverage scoring function ***f***, find a subset of reviews $S \subseteq R$ of size $|S| = k$ that maximizes,

$$F(S) = \sum_{a \in A} f(S, a)$$

Where, ***F(S)*** is the *cumulative coverage scoring function* with respect to the *coverage scoring function* **f(S, a).**

Coverage functions

Unit Coverage function

$f_U(S, a) = 0, \forall\ a \notin A_S$

$= 1$, otherwise

Gives equal importance to all the attributes

Quality Coverage function

Quality function: $q : R \rightarrow [0, 1]$

$f_q(S, a) = max_{r \in Sa}\ q(r)$

Where, $\boldsymbol{S_a}$ denote the set of reviews in $\boldsymbol{S}$ that cover $\boldsymbol{a}$, that is, $\boldsymbol{S_a = S \cap R_a}$

Group and Soft Coverage function

- partition the reviews into ***g*** groups, $R = \{R^1, \ldots, R^g\}$
- let $S^i = S \cap R^i$ denote the set of reviews in ***S*** that belong to group R^i

$$f_g(S, a) = min_{i=1\ldots g} f(S^i, a)$$

Where, $f(S^i, a)$ can be either $f_u(S, a)$ or $f_q(S, a)$

  - So, Group coverage function is defined as:
    - *group-unit coverage function:* $f_{gu}(S, a)$
    - *group-quality coverage function:* $f_{gq}(S, a)$

- The group coverage function $f_g(S, a)$ requires that a given attribute must be covered by all different groups, failing which it results in a min score of **0**

$$f_s(S, a) = \sum_i f(S^i, a)$$

Where, $f(S^i, a)$ can be either $f_u(S, a)$ or $f_q(S, a)$

- So, Soft coverage function is defined as:
  - *soft-unit coverage function:* $f_{su}(S, a)$
  - *soft-quality coverage function:* $f_{sq}(S, a)$

Terms and Notations

- Incremental gain of a review ***r :*** $\Delta_s(r) = F(S \cup \{r\}) - F(S)$
- Submodularity : *incremental gain* of adding an element to a set decreases as the size of the set increases.
- F_u, F_q, F_{su} and F_{sq} are submodular and hence follows first Algorithm

Approximation Algorithm1

Algorithm 1 The GREEDY algorithm

Input: Set of reviews $\mathcal{R} = \{r_1, ..., r_n\}$; Set of attributes $\mathcal{A} = \{a_1, ..., a_m\}$; Integer budget value k; Scoring function f.

Output: A set of reviews $\mathcal{S} \subseteq \mathcal{R}$ of size k.

1: $\mathcal{S}_0 = \emptyset$
2: **for all** $i = 1, ..., k$ **do**
3: **for all** $r \in \mathcal{R} \setminus \mathcal{S}_{i-1}$ **do**
4: Compute $\Delta_{\mathcal{S}_{i-1}}(r)$
5: **end for**
6: $r_i = \arg\max_{r \in \mathcal{R} \setminus \mathcal{S}_{i-1}} \Delta_{\mathcal{S}_{i-1}}(r)$
7: $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{r_i\}$
8: **end for**
9: return $\mathcal{S}_k$

Metrics used in Algorithm 2

- GROUP-COVERAGE problem is not submodular, and we use second Algorithm for F_{gu} and F_{gq}
- $T = R^1 \times \cdot \cdot \cdot \times R^g$ denote the set of all possible tuples.
- *Cost* : $C_S(t) = |t \setminus S|$ denote the number of reviews contained in tuple t that are not in S
- Tuple with maximum gain to cost ratio is added to the output set S
- Potential gain of a tuple: $P_S(t) = |U_t \setminus A_S|$ denote the number of attributes partially covered by t that are not already partially or fully covered by S.
- Potential gain is used to break the ties between tuples with the same gain-to-cost ratio
- DENSEST *k*-SUBGRAPH (DKS) problem is a special case of GROUP-COVERAGE problem

Approximate Algorithm2

Algorithm 2 The t-GREEDY algorithm.

Input: Set of reviews $\mathcal{R} = \{r_1, ..., r_n\}$ and groups $\{\mathcal{R}^1, ..., \mathcal{R}^g\}$; Set of attributes $\mathcal{A} = \{a_1, ..., a_m\}$; Integer budget value k; Scoring function f_g

Output: A set of reviews $\mathcal{S} \subseteq \mathcal{R}$ of size k.

Compute $\mathcal{T} = \mathcal{R}^1 \times \cdots \times \mathcal{R}^g$
$\mathcal{S} = \emptyset$
while $|\mathcal{S}| < k$ **do**
  $b = k - |\mathcal{S}|$
  for all $t \in \mathcal{T}$ **do**
    Compute $\Delta_{\mathcal{S}}(t), C_{\mathcal{S}}(t), P_{\mathcal{S}}(t)$
  end for
  $T = \arg\max_{t:C_{\mathcal{S}}(t) \leq b} \Delta_{\mathcal{S}}(t)/C_{\mathcal{S}}(t)$
  $t = \arg\max_{t \in T} P_{\mathcal{S}}(t)$
  $\mathcal{S} = \mathcal{S} \cup t$
end while
return $\mathcal{S}$

Other Baseline Metrics:

- TOPQLTY: Sort the reviews according the their quality and select the top-*k* reviews
- TOPLEN: Sort the reviews according to length, and select the top-*k* reviews
- RANDOM: Randomly select *k* reviews

Mean and standard deviation of the performance measures for the different algorithms

	UCov	QCov	GUCov	GQCov	SUCov	SQCov	Qlty
Greedy-U	**0.98** (0.04)	0.90 (0.09)	0.27 (0.24)	0.26 (0.24)	0.73 (0.11)	0.70 (0.12)	0.83 (0.11)
Greedy-Q	0.97 (0.05)	**0.96** (0.06)	0.21 (0.26)	0.21 (0.27)	0.70 (0.12)	0.73 (0.12)	0.92 (0.07)
Greedy-GU	0.72 (0.27)	0.66 (0.26)	**0.84** (0.14)	0.77 (0.17)	0.62 (0.15)	0.55 (0.16)	0.80 (0.12)
Greedy-GQ	0.71 (0.27)	0.70 (0.28)	0.82 (0.15)	**0.83** (0.15)	0.61 (0.15)	0.58 (0.16)	0.86 (0.11)
Greedy-SU	0.95 (0.07)	0.87 (0.11)	0.77 (0.17)	0.70 (0.18)	**0.86** (0.10)	0.79 (0.13)	0.80 (0.11)
Greedy-SQ	0.95 (0.07)	0.93 (0.08)	0.71 (0.20)	0.72 (0.19)	0.84 (0.10)	**0.84** (0.11)	0.89 (0.08)
TopQlty	0.74 (0.19)	0.77 (0.17)	0.14 (0.23)	0.15 (0.25)	0.52 (0.18)	0.58 (0.18)	**1.00** (0.00)
TopLen	0.88 (0.11)	0.84 (0.13)	0.31 (0.30)	0.31 (0.30)	0.68 (0.14)	0.68 (0.15)	0.85 (0.11)
Random	0.61 (0.11)	0.54 (0.12)	0.16 (0.08)	0.14 (0.07)	0.43 (0.11)	0.40 (0.11)	0.77 (0.08)

Table 1: Mean and standard deviation of the performance measures for the different algorithms

Musical Instrument Review Dataset

	UCOv	QCOv	GUCOv	GQCOv	SUCOv	SQCOV	QLTY
GREEDY-U	**0.99**	0.96	0.18	0.17	0.58	0.57	0.95
GREEDY-Q	0.98	**0.97**	0	0	0.49	0.48	0.98
GREEDY-GU	0.67	0.51	**0.15**	0.12	0.59	0.58	0.91
GREEDY-GQ	0.61	0.63	0	**0**	0.52	0.64	0.96
GREEDY-SU	0.95	0.89	0.74	0.56	**0.84**	0.73	0.97
GREEDY-SQ	0.92	0.91	0.70	0.68	0.81	**0.80**	0.85
TOPQLTY	0.35	0.35	0	0	0.17	0.17	**1.00**
TOPLEN	0.83	0.78	0	0	0.41	0.39	0.89
RANDOM	0.36	0.33	0.04	0.03	0.20	0.18	0.85

Conclusion

- Studied the problem of selecting a small subset of reviews from a large collection of reviews for a product, such that it covers the different attributes of the product with high quality content that represents different viewpoints
- Each greedy algorithm achieves the best value on the target metric, but not always optimal
- All algorithms perform on average better than the random baseline (with the exception of TOPQLTY , TOPLEN, GREEDY-Q on GUCov & GQCov)
- The results are data dependent