UVA CS 6316: Machine Learning : 2019 Fall
Course Project: Deep2Reproduce @
https://github.com/qiyanjun/deep2reproduce/tree/master/2019Fall

Scalable Deletion-Robust Submodular Maximization: Data Summarization with Privacy and Fairness Constraints

Reproduced By: Rohit Musti (on my own)

December 5, 2019

Motivation

- We now have huge amounts of data generated by users
- Stakeholders analyze data for insights
- Insights incur the cost of user privacy
- Algorithms amplify bias in data

Motivation

- Not only an ethical obligation, a legal one as well
  - GDPR outlines the right to be forgotten
  - Title VII of the Civil Rights Act (USA) bans discriminative hiring
- Challenges:
  - Expensive to train bespoke models for protected groups
  - We might not know sensitive features!

Motivation

- What if we could remove sensitive features?
  - Geo-location data
  - Skin colour
  - Gender
  - Age
- Without compromising on accuracy!

Background

- Submodular Optimization:
  - Formalizes the idea of diminishing returns

$$A\ function\ is\ submodular\ if\quad Given\ A \epsilon B \epsilon V,\ \mathrm{j}\ \notin A \notin B, j \epsilon V$$

$$f(A \cup j) - f(A) \geq f(B \cup j) - f(B)$$

  - Problems like ML, web search, social network, crowd sourcing, user modeling

Bertsimas, D., Brown, D. B., and Caramanis, C. Theory andapplications of robust optimization.SIAM review, 53(3):464–501, 2011.

Related Work

Nemhauser et al (1978) demonstrated a simple greedy algorithm that starts with an empty set and simply adds elements with the highest marginal utility provides a $\{1 - {}^{1}/_{e}\}$ approximation guarantee

Kraus et al (2008) introduced classic cardinality constrained submodular maximization for the first time, however returned a set that was logarithmically larger than k, the cardinality

Related Work

Orlin et al (2016) could output a set of size k in polynomial time however it was only robust to $o(\sqrt{k})$ elements

Mirzasoleiman et al (2017) developed a streaming algorithm that was robust to any d elements, however it required massive amounts of memory for k and d

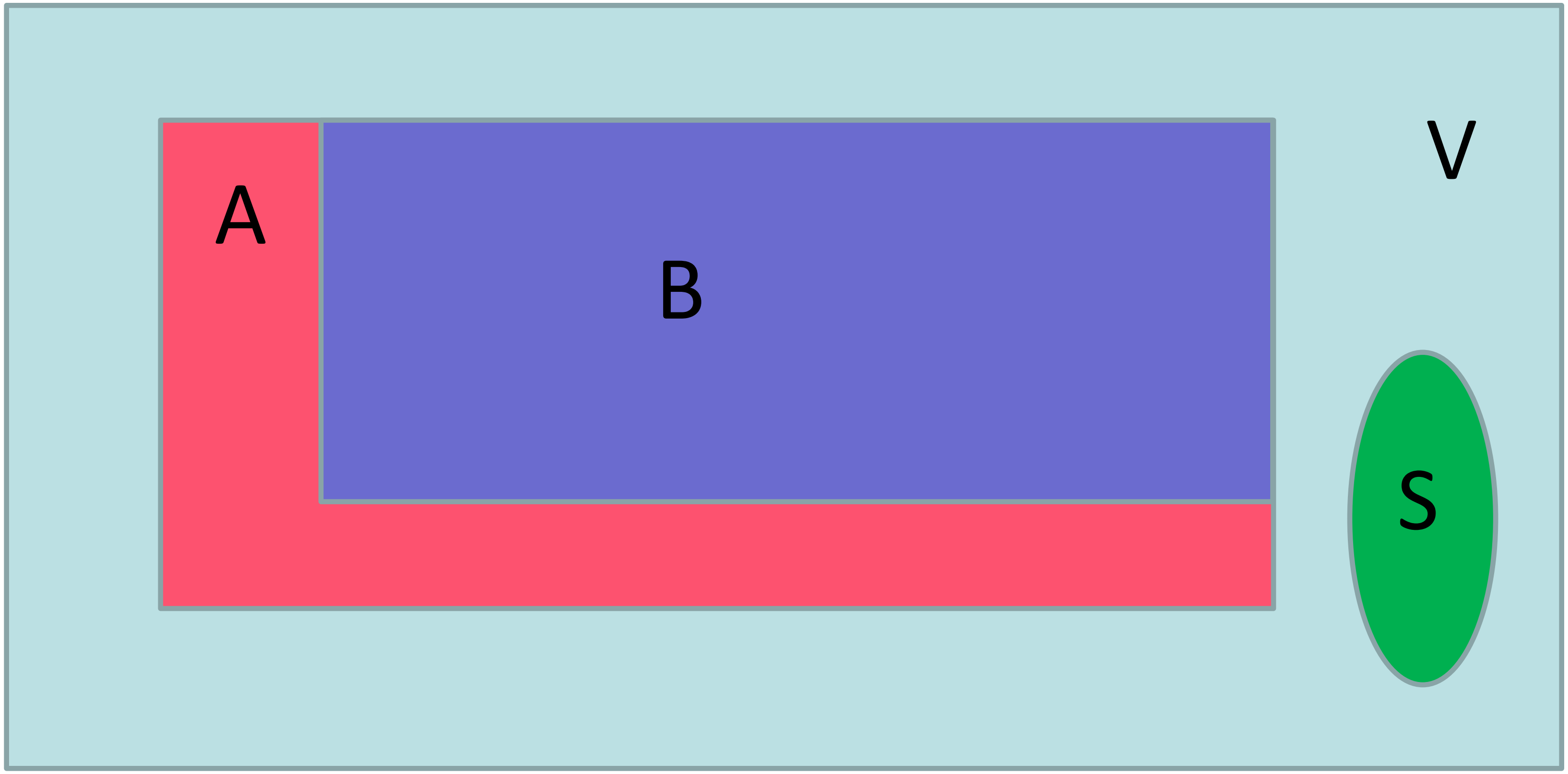
Claim / Target Task

To identify an (α, d) robust randomized core set for a $set\ V$

A randomized robust core set is a random set $A \subseteq V$ such that for any $D \subseteq V$ of size $|D| \leq d$, there exists a $B \subseteq A \backslash D, |B| \leq k$ such that

$$\mathbb{E}[f(B)] \geq \alpha * \max f\ (\{S \mid S \subseteq V \backslash \mathrm{D}, |S| \leq k\})$$

An Intuitive Figure Showing WHY Claim



A – B = D, f(B) >= alpha * f(S)

Proposed Solution

Robust-CoreSet-Centralized

Robust-CoreSet-Streaming

Robust-Distributed

Implementation: Robust-CoreSet-Centralized

1. Select the d + 1 largest element in V and set aside the d+1 largest elements in V into Vt.
2. Set T to the set of (1+e_i) such that (1 + sigma)^i is less than the change in utility of d and greater than the change in utility of d divided by (2 (1 + sigma)k)
3. Set A_t and B_t to the null set for all t in T
4. For all t in T,
    1. while the size of B is greater than d/sigma, add a random element to At from B
        - defining Bt as all e in V such that the value gained by adding e to At is less than (1+e)t but greater than t
5. Set aside all the elements in V not in Bt or At
6. Union Bt with Vt and return it along with A_t as the core set

Implementation: Robust-CoreSet-Streaming

1. Create two sets, A_t and B_t
2. All of the elements in A_t having greater than t marginal gains
3. Good enough elements are in B_t, which only accepts elements within a certain range of utility. When B_t exceeds a certain size and becomes too big, we pick a random element and add it to A_t
    1. This guarantees that the elements being added to A have a similar gain
4. We must then re-compute the marginal gain of the elements in B_t
5. This continues until we have k elements in A or until the data stream ends
6. There are at most d elements with marginal gains within the range acceptable to B_t
7. The core set is the union of B_t and A_t

Implementation: Robust-Distributed

1. First, randomly distribute data onto m machines
2. Each machine runs Robust-Coreset-Centralized as described earlier on its local data
3. After the deletion of set D, the central machine runs m instances of Robust-Centralized to find the Solutions S_i
4. It also runs the classic greedy on the union of the sets from all the machines to find a solution T
5. The best answer is contained in the sets S and T

6. BONUS: you can run Robust-CoreSet-Centralized on the output of Robust-Distributed to get an ultra-compact set

Data Summary

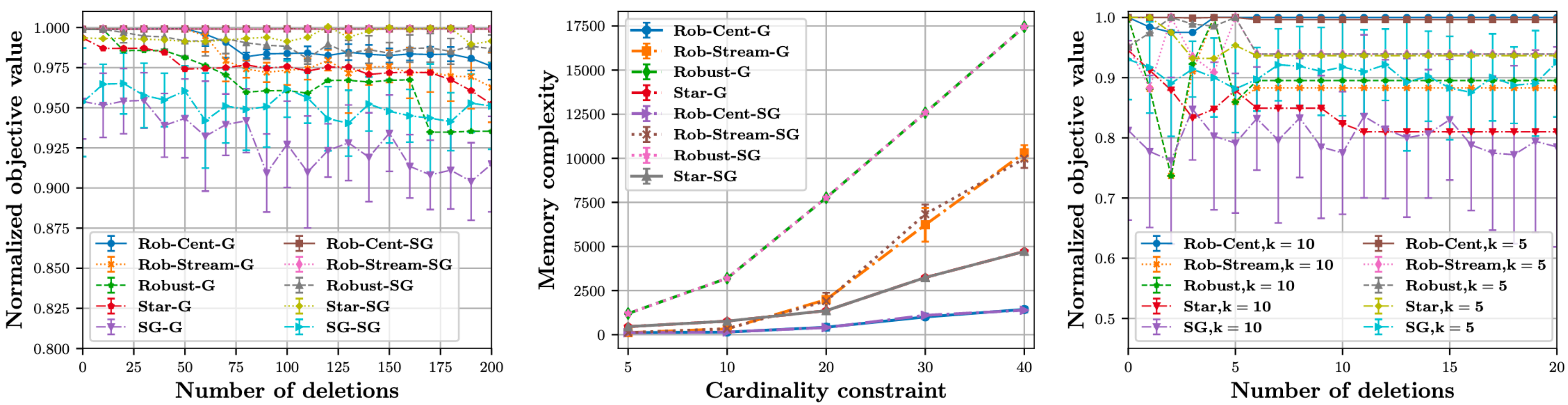
- Location Data from publicly available data sets
  - The goal is to find k representative samples from manhatten latitude longitude data
- The Adult Income Dataset
  - Used to test feature deletion for submodular feature selection
- Census1990
  - Used as a large dataset to understand Robust-Distributed performance

Experimental Results

- For experiment 1, the Manhattan location representation experiment, the proposed set of algorithms came up with better representational values and used less memory
- For experiment 2, predicting adult income data with missing features, the SVM classifier with greedy selected features, had an accuracy of 83%, after deleting race and class sensitive features, the accuracy drops to 79%, when trained on the features found by Robust-Centralized and Robust-Streaming, the performance only dropped to 83.3%
- Robust-Distributed allows for summarizing a data set of almost 2.5 million into just 4,500 points, robust up until deleting 80% of the items

Experimental Analysis

- These are the results from the original paper, the data used to generate the graphs was not publicly available



(a) Uber dataset: we set $d = 5$ and $k = 20$. (b) Uber dataset: we set $d = 5$ and $r = 100$. (c) Adult Income: we set $d = 3$

Figure 1. (a) The effect of deletion on the performance of algorithms with respect two different deletion strategies; (b) memory complexity of robust algorithms for different cardinality constraints; (c) The effect of deletion on the performance for feature selection.

Experimental Analysis

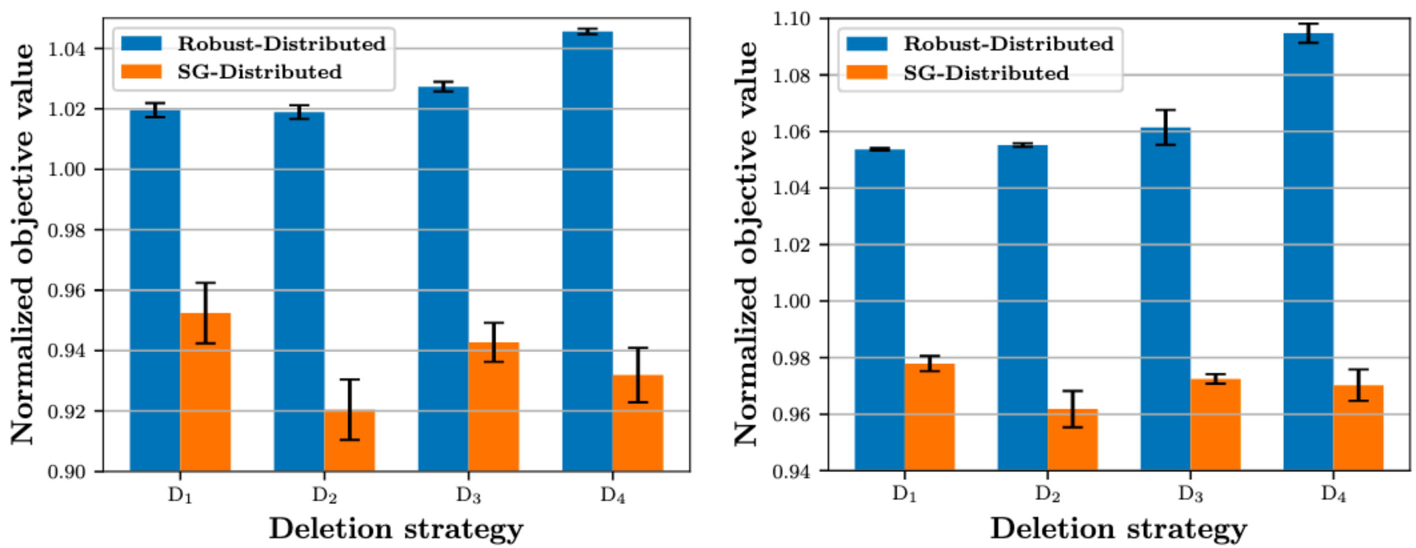
- These are the results from the original paper, the data used to generate the graphs was not publicly available

Table 2. The comparison of Naive Bayes and SVM classifiers for Adult Income dataset. Ten sensitive features are deleted. The number of stored features is reported in parenthesis.

Algorithm	Naive Bayes (Acc.)	SVM (Acc.)
All features	0.798	0.830
GREEDY	0.788	0.796
GREEDY$_D$	0.781	0.793
Rob-Cent	0.781 (22)	0.791
Rob-Stream	0.781 (29)	0.791
ROBUST	0.779 (39)	0.788
STAR-T-GREEDY	0.779 (50)	0.787

Experimental Analysis

- These are the results from the original paper, the data used to generate the graphs was not publicly available



(a) $k = 50$ (b) $k = 100$

Figure 2. Census1990 dataset: ROBUST-DISTRIBUTED versus SG-DISTRIBUTED for four different deleting strategies.

Conclusion and Future Work

- Provided the first scalable and memory efficient algorithms for deletion robust submodular maximization
- They showcased how much powerful the algorithms were in real world scenario for preserving privacy

Challenges in Reproducing Results

1. There were many references to variables in the formulas that were not explained, instantiated, or clarified in other parts of the formula
2. The dataset preprocessing steps loosely described in the paper.
    1. More later: they claimed to produce 101 binary features from the data, unclear how this is actually possible given the data
3. I personally didn't have a lot of background and terminology that the general audience for this paper has
4. While the paper did offer some intuition for how the concepts worked, they weren't fully flushed out
5. Being solo, I didn't have a team to bounce ideas off of
6. There is very little to no existing open code or data available for related/similar papers for this particular research problem. So I had to do a lot of the work from complete scratch

What I did

I always able to reproduce experiment 2. Experiment 3 required hardware beyond my means and Experiment 1 and 2 tested the effectiveness of the same algorithms, just on an unwieldy + inconvenient dataset.

Experiment 1 also required implementing several other algorithms that have only been written in research papers. This would mean reproducing two other papers so I didn’t pursue that particular experiment.

MY RESULTS: Robust Centralized Core - Adult Income Dataset

I implemented the Robust-Coreset-Centralized (by definition also the Robust Coreset) algorithms

- SVM Result
  - lazy greedy feature selection: 79.08%
  - Submodular detection feature selection: 83.39%
  - no feature selection: 83.85%
- Naïve Bayes Results
  - lazy greedy feature selection: 78.91%
  - Submodular detection feature selection: 78.89%
  - no feature selection: 78.06%

MY RESULTS: Robust Core Stream - Adult Income Dataset

I implemented the Robust-Coreset-Centralized (by definition also the Robust Coreset) algorithms

- SVM Result
  - lazy greedy feature selection: 78.96%
  - fancy research feature selection: 81.79%
  - no feature selection: 83.92%
- Naïve Bayes Results
  - lazy greedy feature selection: 78.91%
  - Submodular detection feature selection: 77.86%
  - no feature selection: 78.23%

References

Acknowledgements

Amin Karbasi was supported by a DARPA Young Faculty Award (D16AP00046) and a AFOSR Young Investigator Award (FA9550-18-1-0160). Ehsan Kazemi was supported by the Swiss National Science Foundation (Early Postdoc.Mobility) under grant number 168574.

References

Bach, F. et al. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A., and Krause, A. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 671–680. ACM, 2014.

Barbosa, R., Ene, A., Nguyen, H., and Ward, J. The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning*, pp. 1236–1244, 2015.

Discrete Algorithms, pp. 1202–1216. Society for Industrial and Applied Mathematics, 2015.

Chayes, J. T. How Machine Learning Advances Will Improve the Fairness of Algorithms, 2017. URL https://www.huffingtonpost.com/entry/how-to-shorten-a-website-link_us_579bb7aee4b07066ba1ea7dd.

Chekuri, C., Gupta, S., and Quanrud, K. Streaming algorithms for submodular function maximization. In *International Colloquium on Automata, Languages, and Programming*, pp. 318–330. Springer, 2015.

Feldman, M., Harshaw, C., and Karbasi, A. Greed Is Good: Near-Optimal Submodular Maximization via Greedy Optimization. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 758–784, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

Feldman, M., Karbasi, A., and Kazemi, E. Do Less, Get More: Streaming Submodular Maximization with Subsampling. *CoRR*, abs/1802.07098, 2018. URL http://arxiv.org/abs/1802.07098.

References

Barbosa, R., Ene, A., Nguyen, H., and Ward, J. The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning*, pp. 1236–1244, 2015.

Bertsimas, D., Brown, D. B., and Caramanis, C. Theory and applications of robust optimization. *SIAM review*, 53(3): 464–501, 2011.

Beutel, A., Chen, J., Zhao, Z., and Chi, E. H. Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations. *arXiv preprint arXiv:1707.00075*, 2017.

Blake, C. L. and Merz, C. J. Uci repository of machine learning databases [http://www. ics. uci. edu/~ mlearn/mlrepository. html]. irvine, ca: University of california. *Department of Information and Computer Science*, 55, 1998.

Bogunovic, I., Mitrovic, S., Scarlett, J., and Cevher, V. Robust Submodular Maximization: A Non-Uniform Partitioning Approach. pp. 508–516, 2017.

Feldman, M., Karbasi, A., and Kazemi, E. Do Less, Get More: Streaming Submodular Maximization with Subsampling. *CoRR*, abs/1802.07098, 2018. URL http://arxiv.org/abs/1802.07098.

Herbrich, R., Lawrence, N. D., and Seeger, M. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in neural information processing systems*, pp. 625–632, 2003.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM, 2003.

Krause, A. and Gomes, R. G. Budgeted nonparametric learning from data streams. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 391–398, 2010.

References

Bogunovic, I., Zhao, J., and Cevher, V. Robust Maximization of Non-Submodular Objectives. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pp. 890–899. PMLR, 2018.

Borodin, A., Jain, A., Lee, H. C., and Ye, Y. Max-Sum Diversification, Monotone Submodular Functions, and Dynamic Updates. *ACM Transactions on Algorithms (TALG)*, 13(3):41, 2017.

Buchbinder, N., Feldman, M., and Schwartz, R. Online submodular maximization with preemption. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on*

Krause, A. and Guestrin, C. Near-optimal Nonmyopic Value of Information in Graphical Models. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pp. 324–331, 2005.

Krause, A., McMahan, H. B., Guestrin, C., and Gupta, A. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008.

Kumar, R., Moseley, B., Vassilvitskii, S., and Vattani, A. Fast greedy algorithms in mapreduce and streaming. *ACM Transactions on Parallel Computing*, 2(3):14, 2015.

References

Lin, H. and Bilmes, J. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, Stroudsburg, PA, USA, 2011.

Mirrokni, V. and Zadimoghaddam, M. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 153–162. ACM, 2015.

Mirzasoleiman, B., Karbasi, A., Sarkar, R., and Krause, A. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pp. 2049–2057, 2013.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier Than Lazy Greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1812–1818, 2015.

Mirzasoleiman, B., Karbasi, A., and Krause, A. Deletion-Robust Submodular Maximization: Data Summarization with "the Right to be Forgotten". In *International Conference on Machine Learning*, pp. 2449–2458, 2017.

UberDataset. Uber Pickups in New York City. URL https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1954–1963, 2015.

Yue, Y. and Guestrin, C. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pp. 2483–2491, 2011.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 325–333, 2013.

Zhang, H. The Optimality of Naive Bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*, pp. 562–567, 2004.

References

Mitrovic, S., Bogunovic, I., Norouzi-Fard, A., Tarnawski, J. M., and Cevher, V. Streaming Robust Submodular Maximization: A Partitioned Thresholding Approach. In *Advances in Neural Information Processing Systems*, pp. 4560–4569, 2017.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functionsâĂŤi. *Mathematical Programming*, 14(1): 265–294, 1978.

Orlin, J. B., Schulz, A. S., and Udwani, R. Robust monotone submodular function maximization. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 312–324. Springer, 2016.

Seeger, M. Greedy forward selection in the informative vector machine. Technical report, Technical report, University of California at Berkeley, 2004.

Singla, A., Tschiatschek, S., and Krause, A. Noisy Submodular Maximization via Adaptive Sampling with Applications to Crowdsourced Image Collection Summarization. In *AAAI*, pp. 2037–2043, 2016.

Smola, A. J. and Schölkopf, B. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

Tzoumas, V., Jadbabaie, A., and Pappas, G. J. Resilient Non-Submodular Maximization over Matroid Constraints. *arXiv preprint arXiv:1804.01013*, 2018.