Feature Cross Search via Submodular Optimization

Lin Chen[1]
Simons Institute for the Theory of Computing, University of California, Berkeley, CA, USA

Hossein Esfandiari
Google Research, New York, NY, USA

Gang Fu
Google Research, New York, NY, USA

Vahab S. Mirrokni
Google Research, New York, NY, USA

Qian Yu
Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, USA

Abstract

In this paper, we study feature cross search as a fundamental primitive in feature engineering. The importance of feature cross search especially for the linear model has been known for a while, with well-known textbook examples. In this problem, the goal is to select a small subset of features, combine them to form a new feature (called the crossed feature) by considering their Cartesian product, and find feature crosses to learn an *accurate* model. In particular, we study the problem of maximizing a normalized Area Under the Curve (AUC) of the linear model trained on the crossed feature column.

First, we show that it is not possible to provide an $n^{1/\log\log n}$-approximation algorithm for this problem unless the exponential time hypothesis fails. This result also rules out the possibility of solving this problem in polynomial time unless $\mathsf{P} = \mathsf{NP}$. On the positive side, by assuming the naïve Bayes assumption, we show that there exists a simple greedy $(1 - 1/e)$-approximation algorithm for this problem. This result is established by relating the AUC to the total variation of the commutator of two probability measures and showing that the total variation of the commutator is monotone and submodular. To show this, we relate the submodularity of this function to the positive semi-definiteness of a corresponding kernel matrix. Then, we use Bochner's theorem to prove the positive semi-definiteness by showing that its inverse Fourier transform is non-negative everywhere. Our techniques and structural results might be of independent interest.

2012 ACM Subject Classification Computing methodologies → Feature selection

Keywords and phrases Feature engineering, feature cross, submodularity

Digital Object Identifier 10.4230/LIPIcs.ESA.2021.31

Related Version *Full Version*: https://arxiv.org/pdf/2107.02139.pdf

1 Introduction

Feature engineering is one of the most fundamental problems in machine learning and it is the key to all supervised learning models. In feature engineering, we start with a collection of features (a.k.a., raw attributes) and turn them into a new set of features, with the purpose of improving the accuracy of the learning model. This is often done by some basic operations, such as removing irrelevant and redundant features (studied as feature selection [10, 32, 33, 26, 11, 25, 18]), combining features (a.k.a., feature cross [30, 20]) and bucketing and compressing the vocabulary of the features [2, 7, 28, 1].

[1] Authors are ordered alphabetically.

Finding an *efficient* set of features to combine (*i.e.*, cross) is one of the main primitives in feature engineering. Let us start with a text book example to show the importance of feature cross for the linear model. Consider a model with two features, language, which can be English or Spanish, and country, which can be Mexico or Scotland. Say if English appears with Scotland, or if Spanish appears with Mexico, the label is 1. Otherwise the label is 0. It is easy to see that in this case there is no linear model using these two features with a nontrivial accuracy (*i.e.*, the best model matches the label with probability 1/2). By crossing these two features, we get a new feature with four possible values (English, Mexico), (English, Scotland), (Spanish, Mexico), (Spanish, Scotland). Now, a linear model based on this new feature can perfectly match the label. This is a well-known concept in feature engineering.

Unlike feature selection and vocabulary compression, and despite the importance of feature cross search in practice, this problem is not well studied from a theoretical perspective. While some heuristics and exponential-time algorithms have been developed for this problem (*e.g.*, [30, 20]), the complexity of designing approximation algorithms for this problem is not studied. This might be due to the complex behavior of crossing features on the accuracy of the learning models. In this work, we provide a simple formulation of this problem, and initiate a theoretical study.

Let us briefly define the problem as follows and defer the formal definition of the problem to a later section: Given a set of n features, and a number k, compute a set of at most k features out of n features and combine these k features such that the accuracy of the optimum linear model on the combined feature is maximized. To measure the accuracy we use normalized *Area Under the Curve (AUC)*. The bound k is to avoid over fitting.[2] This is a very basic definition for the feature cross search problem and can be considered as a building block in feature engineering. In fact, as we discuss later in the paper, it is still hard to design algorithms for this basic problem.

First, we show that there is no $n^{1/\log\log n}$-approximation algorithm for feature cross search unless the exponential time hypothesis fails. Our hardness result also implies that there does not exist a polynomial-time algorithm for feature cross search unless $\mathsf{P} = \mathsf{NP}$. It is easy to extend these hardness results to other notions of accuracy such as probability of matching the label. Obviously, this hardness result holds for any extension of the problem as well.

In fact, often, the real world inputs are not adversarially constructed. Usually, the inputs follow some structural properties that allow simple algorithms to work efficiently. With this intuition in mind, to complement our hardness result, for features under the naïve Bayes assumption [22, 29, 9], we provide a $(1 - 1/e)$-approximation algorithm that only needs polynomially many function evaluations. We further discuss and justify this assumption in Section 1.1.

In Section 1.1, we define the problem formally and present our results as well as an overview of our techniques. In Section 2, we provide the preliminary definitions and observations that will be used later in the proofs. In Section 3, we present our hardness results. We relate the maximum AUC to the log-likelihood ratio and the total variation of the commutator of two probability distributions in Section 4. Section 5 establishes the monotonicty and submodularity of the maximum AUC as a set function. This section forms the most technical part of the paper. In Section 6 we present other related works. Finally, Section 7 concludes the paper.

[2] In practice this number is chosen by tracking the accuracy of the model on the validation data. However, this is out of the scope of this paper.

1.1 Problem Statement and Our Contributions

We start with some definitions necessary to present our results, and then, we present our contributions. Assume that the dataset comprises $n = |U|$ categorical feature columns and a binary label column, where U is the set of all feature columns. Let the random variable X_i denote the value of the i-th feature column ($i \in U$) and $C \in \{0, 1\}$ be the value of the binary label. The random variables $X_1, \ldots, X_{|U|}, C$ follow a joint distribution $\mathcal{D}$. Additionally, we assume that the support of the random variable X_i is a finite set $V_i \subseteq \mathbb{N}$. The set V_i is also known as the *vocabulary* of the i-th feature column. If $A \subseteq U$ is a set of feature columns, we write V_A for $\prod_{a \in A} V_a$ and write X_A for $(X_a | a \in A)$, where $(X_a | a \in A)$ is a vector indexed by $a \in A$ (for example, if $A = \{1, 2, 4\}$, the vector X_A is a 3-dimensional vector (X_1, X_2, X_4).

Suppose that we focus on a set of feature columns and temporarily ignore the remaining feature columns. In other words, we consider the dataset modeled by the distribution of (X_A, C). Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ denote the set of extended real numbers. Given a function $\sigma : V_A \to \overline{\mathbb{R}}$ that assigns a score to each possible value of X_A, and given a threshold τ, we predict a positive label for X_A if $\sigma(X_A) > \tau$ and predict a negative label if $\sigma(X_A) < \tau$. If $\sigma(X_A) = \tau$, we allow for predicting a positive or negative label at random. Let TPR and FPR denote the true and false positive rate of this model given a certain decision rule, respectively. Note that both true and false positive rates lie in $[0, 1]$. If one varies τ from $-\infty$ to ∞ while fixing the score function σ, a curve that consists of the collection of achievable points (FPR, TPR) is produced and the curve resides in the square $[0, 1] \times [0, 1]$. The area under the curve (AUC) [4] is then defined as the area of the region enclosed by this curve, and the two lines $\text{FPR} = 1$ and $\text{TPR} = 0$.

An equivalent definition is that AUC is roughly the probability that a random positive instance has a higher score (in terms of σ) than a negative instance (we say *roughly* because in Definition 1, we have to be careful about tiebreaking, i.e., the second term).

▶ **Definition 1** (Area under the curve (AUC) [4]). *Given a set of feature columns A and a function $\sigma : V_A \to \mathbb{R}$, the area under the curve (AUC) of A and σ is*

$$\text{AUC}_\sigma(A) = \Pr[\sigma(X_A^+) > \sigma(X_A^-) | C^+ = 1, C^- = 0] + \frac{1}{2} \Pr[\sigma(X_A^+) = \sigma(X_A^-) | C^+ = 1, C^- = 0],$$

where $(X_A^+, C^+), (X_A^-, C^-) \sim \mathcal{D}$ are i.i.d. and $X_A^\gamma = (X_a^\gamma | a \in A)$ obeys a marginal distribution of $\mathcal{D}$ (γ is either $+$ or $-$).

The maximum AUC is the AUC of the best scoring function. It is a function of the set of feature columns and independent of the scoring function.

▶ **Definition 2** (Maximum AUC). *Given a set of feature columns A, the maximum AUC is*

$$\text{AUC}^*(A) = \sup_{\sigma : V_A \to \mathbb{R}} \text{AUC}_\sigma(A).$$

Now we are ready to present our results. We start with Observation 3 which provides a characterization of AUC via the total variation distance.

▶ **Observation 3** (AUC as total variation distance). *Let P_i^A be the conditional distribution $\Pr[X_A | C = i]$ on V_A and let $d_{TV}(P, Q)$ denote the total variation distance between two probability measures P and Q. We have*

$$\text{AUC}^*(A) = \frac{1}{2} + \frac{1}{2} d_{TV}(P_1^A \times P_0^A, P_0^A \times P_1^A) = \frac{1}{2} + \frac{1}{2} \sum_{x, y \in V_A} \left| P_1^A(x) P_0^A(y) - P_0^A(x) P_1^A(y) \right|,$$

where $P_1^A \times P_0^A$ and $P_0^A \times P_1^A$ denote the product measures.

Recall that if P and Q are two probability measures on a common σ-algebra $\mathcal{F}$, the *total variation distance* between them is

$$d_{TV}(P,Q) \triangleq \sup_{A\in\mathcal{F}} |P(A)-Q(A)| \in [0,1]\,.$$

If the sample space Ω (the set of all outcomes) is finite, Scheffé's lemma [27] gives

$$d_{TV}(P,Q) = \frac{1}{2}\|P-Q\|_1 \triangleq \frac{1}{2}\sum_{\omega\in\Omega} |P(\omega)-Q(\omega)|\,. \tag{1}$$

We present the proof of Observation 3 in Section 4. Observation 3 shows that the maximum AUC is an affine function of the total variation distance between $P_1^A \times P_0^A$ and $P_0^A \times P_1^A$, where P_i^A is the probability measure conditioned on the label. In light of (1), we have $d_{TV}(P_1^A \times P_0^A, P_0^A \times P_1^A) = \frac{1}{2}\|P_1^A \times P_0^A - P_0^A \times P_1^A\|_1$. We call the signed measure $P_1^A \times P_0^A - P_0^A \times P_1^A$ the *commutator* of the two probability measures P_0^A and P_1^A. Our second remark is that since the total variation distance always resides on $[0,1]$, the range of the maximum AUC is $[1/2,1]$.

The next theorem is our main hardness result, stating that it is not possible to approximate feature cross search, unless the exponential time hypothesis [12] fails. We consider maximization of $2\,\mathrm{AUC}^*(A)-1$ rather than $\mathrm{AUC}^*(A)$ in (2) because the range of the maximum AUC is $[1/2,1]$ (as we remark before) and assigning the same score to all feature values in V_A attains an AUC of $1/2$, thereby achieving at least a $1/2$-approximation. In light of its range, we consider its normalized version $2\,\mathrm{AUC}^*(A)-1$ whose range is $[0,1]$. We prove this theorem in Section 3. In fact, our hardness result also implies that the feature cross search problem is NP-hard (see Corollary 14).

▶ **Theorem 4.** *There is no $n^{1/\,\mathsf{poly}(\log\log n)}$-approximation algorithm for the following maximization problem unless the exponential time hypothesis [12] fails.*

$$\max_{A\subseteq U,|A|=k} (2\,\mathrm{AUC}^*(A)-1)\,. \tag{2}$$

Although the above hardness result rules out the existence of an algorithm with a good approximation factor in the general case, it is very rare to face such hard examples in practice. We consider the naïve Bayes assumption that all feature columns are conditionally independent given the label. We borrowed this assumption from the widely-used naïve Bayes classifier [22]. For example, under the same assumption, [15] established the submodularity of mutual information and [6] proved that in the sequential information maximization problem, the most informative selection policy behaves near optimally. [29] conducted an empirical study on the public software defect data from NASA with PCA pre-processing. They concluded that this assumption was not harmful. Although relaxing the assumption could produce numerically more favorable results, they were not statistically significantly better than assuming this assumption. In another example [9], based on their analysis on three real-world datasets for natural language processing tasks (MDR, Newsgroup and the ModApte version of the Reuters-21578), they drew a similar conclusion that relaxing the assumption did not improve the performance.

▶ **Assumption 1** (Naïve Bayes)**.** Given the label, all feature columns are independent. In other words, it holds for $A \subseteq U$ and $i = 0, 1$ that

$$\Pr[X_A = x_A | C = i] = \prod_{a\in A} \Pr[X_a = x_a | C = i]\,. \tag{3}$$

Our major algorithmic contribution is to show that under the naïve Bayes assumption the set function AUC* is monotone submodular, which in turn, implies that a greedy algorithm provides a constant-factor approximation algorithm for this problem.[3]

▶ **Theorem 5.** *Under the naïve Bayes assumption, the set function* $\mathrm{AUC}^* : 2^U \to \mathbb{R}$ *is monotone submodular.*

This theorem implies the following result in light of the result of [24].

▶ **Corollary 6.** *Under the naïve Bayes assumption, there exists a* $(1 - 1/e)$*-approximation algorithm that only needs polynomially many evaluations of* AUC* *for feature cross search.*

To show Theorem 5, we prove Proposition 7. Proving this proposition requires an involved analysis and it may be of independent interest in statistics.

▶ **Proposition 7** (Proof in Section 5). *Let* U *be a finite index set. Assume that for every* $a \in U$*, there are a pair of probability measures* P_0^a *and* P_1^a *on a common sample space* V_a*. For any* $A \subseteq U$*, define the set function* $F : 2^U \to \mathbb{R}_{\geq 0}$ *by*

$$F(A) = d_{TV}\left(\bigtimes_{a\in A} P_1^a \times \bigtimes_{a\in A} P_0^a, \bigtimes_{a\in A} P_0^a \times \bigtimes_{a\in A} P_1^a\right). \tag{4}$$

The set function F *is monotone and submodular.*

Its proof is presented in Section 5. In fact, the most technical part of this paper is to prove Proposition 7 which claims that the total variation of the commutator of probability measures is monotone submodular. The monotonicity is a consequence of the subadditivity of the absolute value function. Submodularity is the technically harder part and is shown in the following four steps.

First, we introduce the notion of *involution equivalence*. An involution is a map from a set to itself that is equal to its inverse map. Two probability measures P and P' are said to be involution equivalent if there exists an involution f on the sample space Ω such that for every $x \in \Omega$, it holds that $P(x) = P'(f(x))$. Note that if $P(x) = P'(f(x))$ holds for every $x \in \Omega$, we have $P'(x) = P(f(x))$ also holds for every $x \in \Omega$. In fact, it defines a symmetric relation on probability measures on Ω. If P and Q are two probability measures on a common sample space, the product measures $P \times Q$ and $Q \times P$ are involution equivalent and connected by the natural transpose involution f that sends $(x, y) \in \Omega^2$ to $(y, x) \in \Omega^2$.

The second step is in light of a key observation that summing a bivariate function of two involution equivalent probability measures over the common sample space remains invariant under the swapping of the two measures. Based on this key observation, if P and P' are involution equivalent, for every x in their common sample space, we construct the probability measures of two Bernoulli random variables U_x and U'_x such that $U_x(1) = U'_x(0) = \frac{P(x)}{P(x)+P'(x)}$ and $U_x(0) = U'_x(1) = \frac{P'(x)}{P(x)+P'(x)}$. The two Bernoulli probability measures U_x and U'_x are again involution equivalent and connected by the swapping of 0 and 1. To establish submodularity, one has to check an inequality that characterizes the diminishing returns property (see equation (5) in Section 2.2). Another key observation is that after defining the Bernoulli probability measures, the desired inequality can be shown to be a conic combination of the same inequality with *some* (not all) probability measures in the inequality replaced

[3] We will review the definition of submodular and monotone set functions in Section 2.2.

by the Bernoulli probability measures U_x and U'_x. To make the above observation work, we have to require that the remaining probability measures unreplaced in the inequality must be either of the form $P \times Q$ or its transpose $Q \times P$. Using this approach, we reduce the problem to the Bernoulli case.

Third, after reducing the problem to the Bernoulli case, performing a series of more involved algebraic manipulations, we re-parametrize the desired inequality that formulates the diminishing returns property and obtain that the inequality is equivalent to the positive semi-definiteness of a quadratic form with respect to a kernel matrix. However, this re-parametrization is valid only for elements of a positive measure with respect to some probability measures in the inequality. As a consequence, prior to the algebraic manipulations and re-parametrization, we have to eliminate those elements of measure zero by showing that their total contribution to the sum is zero. We would like to remark here that the individual terms may not be zero but they are canceled out under the summation.

Finally, to show that the aforementioned kernel matrix is positive semi-definite, we prove that it is induced by a positive definite function. We establish the positive definiteness of the function by showing that its inverse Fourier transform is non-negative everywhere (this is an implication of the Bochner's theorem, see Section 2).

Theorem 5 is a straightforward corollary of Proposition 7.

Proof. Let $P_i^A[\cdot]$ denote $\Pr[X_A|C=i]$, the conditional probability measure on V_A given the labeling being i, where $i = 0, 1$. When $A = \{a\}$ is a singleton, we write P_i^a for $P_i^{\{a\}}$ as a shorthand notation. Under Assumption 1, (3) can be re-written as $P_i^A[x_A] = \prod_{a\in A} P_i^a[x_a]$, or in a more compact way,

$$P_i^A = \bigtimes_{a\in A} P_i^a \,.$$

By Observation 3, we have

$$\begin{aligned}
\mathrm{AUC}^*(A) &= \frac{1}{2} + \frac{1}{2} d_{TV}(P_1^A \times P_0^A, P_0^A \times P_1^A) \\
&= \frac{1}{2} + \frac{1}{2} d_{TV}(\bigtimes_{a\in A} P_1^a \times \bigtimes_{a\in A} P_0^a, \bigtimes_{a\in A} P_0^a \times \bigtimes_{a\in A} P_1^a) \\
&= \frac{1}{2} + \frac{1}{2} F(A) \,.
\end{aligned}$$

Since $F(A)$ is monotone submodular by Proposition 7, so is AUC^*. ◀

2 Preliminaries

Throughout this paper, let Δ_Ω denote the set of all probability measures on a finite set Ω and we always assume that the sample space Ω is finite. The set of extended real numbers is denoted by $\overline{\mathbb{R}}$ and defined as $\mathbb{R} \cup \{-\infty, +\infty\}$.

2.1 Involution Equivalence

We first review the definition of an involution.

▶ **Definition 8** (Involution). *A map $f : \Omega \to \Omega$ is said to be an involution if for all $x \in \Omega$, it holds that $f(f(x)) = x$.*

In this paper, we introduce a new notion termed *involution equivalence*, which forms an equivalence relation on Δ_Ω. Intuitively, two probability measures on a common sample space Ω are involution equivalent if they are the same after renaming the elements in Ω via an involution.

▶ **Definition 9** (Involution equivalence). *Let P, P' be two probability measures on a common sample space Ω. We say that P and P' are involution equivalent if there exists an involution f such that for all $x \in \Omega$, $P(x) = P'(f(x))$. If P and P' are involution equivalent, we denote it by $P \overset{f}{\sim} P'$ or $P \sim P'$ with the involution f omitted when it is not of our interest.*

▶ Remark 10. If $P \overset{f}{\sim} P'$, we have $P'(x) = P'(f(f(x)) = P(f(x))$.

▶ Remark 11 (Transpose involution). If P and P' are two probability measures on a common sample space Ω, the product measure $P \times P'$ is involution equivalent to $P' \times P$ via the natural transpose map $\top$ that sends $(x, y) \in \Omega^2$ to $\top(x, y) = (y, x) \in \Omega^2$. Thus we write $P \times P' \overset{\top}{\sim} P' \times P$ and term $\top$ a *transpose involution*.

2.2 Submodular and Monotone Set Functions

Let us recall the definition of submodular and monotone set functions. Submodular set functions are those satisfying that the marginal gain of adding a new element to a set is no smaller than that of adding the same element to its superset. This property is called the *diminishing returns property*, which naturally arises in data summarization [23], influence maximization [34], and natural language processing [19], among others.

▶ **Definition 12** (Submodular set function, [24, 14]). *A set function $f : 2^U \to \mathbb{R}_{\geq 0}$ is* submodular *if for any $A \subseteq U$ and $a, b \in \Omega \setminus A$ such that $a \neq b$, it satisfies*

$$f(A \cup \{a\}) - f(A) \geq f(A \cup \{a, b\}) - f(A \cup \{b\}) . \tag{5}$$

The above Equation (5) formulates the diminishing returns property. Its left-hand side is the marginal gain of adding a to a set A while the right-hand side is the marginal gain of adding the same element a to the superset $A \cup \{b\}$.

A monotone set function is a function that assigns a higher function value to a set than all its subsets.

▶ **Definition 13** (Monotone set function). *A set function $f : 2^U \to \mathbb{R}$ is* monotone *if for any $A \subseteq B \subseteq U$, we have $f(A) \leq f(B)$.*

3 Hardness Result

In this section we show the hardness of approximation of the feature cross search problem. We say an algorithm is an α-approximation algorithm for the feature cross search problem if its accuracy (*i.e.*, $2\,\mathrm{AUC} - 1$) is at least α times that of the optimum algorithm.

As a byproduct, we show a hardness result for a feature selection problem based on mutual information defined as follows. In the label-based mutual information maximization problem we have a universe of features U and a vector of labels C, and we want to select a subset S of size k from U that maximizes the mutual information $I(S; C)$. In other words we want to solve $\operatorname{argmax}_{S \subseteq U, |S| = k} I(S; C)$. We say an algorithm is an α-approximation algorithm for the label-based mutual information maximization problem if it reports a set S such that $\alpha \leq \frac{I(S;C)}{\max_{S' \subseteq U, |S'| = k} I(S';C)}$.

For both problems, we show that an α-approximation algorithm for the problem implies an α-approximation algorithm for the k-densest subgraph problem. In the k-densest subgraph problem we are given a graph $G(V, E)$ and a number k and we want to pick a subset S of size k from V such that the number of edges induced by S is maximized. Recently, [21] shows that there is no [almost polynomial] $n^{-1/\,\mathsf{poly}(\log\log n)}$- approximation algorithm for k-densest subgraph that runs in polynomial time unless the exponential time hypothesis fails. The best known algorithm for this problem has approximation factor $n^{-1/4}$ [3].

▶ **Theorem 4.** *There is no $n^{1/\,\mathsf{poly}(\log\log n)}$-approximation algorithm for the following maximization problem unless the exponential time hypothesis [12] fails.*

$$\max_{A\subseteq U, |A|=k} (2\,\mathrm{AUC}^*(A) - 1)\,. \tag{2}$$

Proof. We prove this theorem via an approximation preserving reduction from k-densest subgraph. Let $G(V, E)$ be an instance of k-densest subgraph problem. We construct a set of features as follows. There are $n = |V|$ features each corresponding to one vertex of G. For a vertex $v \in V$ we indicate the value of the feature corresponding to v by x_v. There are three possible feature values, 0, 1 and #. The values of the features are determined by the following random process. Select an edge (u, v) uniformly at random from E. The value of the features x_v and x_u are chosen independently and uniformly at random from $\{0, 1\}$. The value of all other features are #. The value of the label is $x_v \oplus x_u$. To show the hardness of approximation of the feature cross search, we show that any solution of accuracy $\phi = 2\,\mathrm{AUC} - 1$ corresponds to a subgraph of G with k vertices and ϕm edges and vise versa.

Let H be a subgraph of G with k vertices and ϕm edges. Let S be the set of features corresponding to the vertices in H. We analyze this in two cases.

Case 1. The value of the crossed feature contains zero or one numbers (*i.e.*, all are #, or all but one are #). Note that this case corresponds to a scenario that the pair of features with binary value are not both in S and hence it happens with probability $\frac{m-\phi m}{m} = 1 - \phi$. Moreover, note that in this case the value of the crossed feature is independent of the value of the label (i.e., given the value of the feature the label is 0 or 1 with probability 1/2).

Case 2. The value of the crossed feature contains two numbers. In this case one can easily predict the correct label with probability 1 (i.e., if the numbers are both 0 or both 1 output 0, otherwise output 1). Moreover, note that this case corresponds to a scenario that the pair of features with binary values are both in S and hence it happen with probability $\frac{\phi m}{m} = \phi$.

Case 1 happens with probability $1 - \phi$ and in this case the label is independent of the value of the crossed feature, and Case 2 happens with probability ϕ, where the label can be predicted with probability 1. Therefore, we have $\mathrm{AUC} = \int_0^1 \phi + (1-\phi)p dp = \phi + \frac{1-\phi}{2} = \frac{1+\phi}{2}$ which gives us $2\,\mathrm{AUC} - 1 = \phi$ as claimed. ◀

In fact, the densest subgraph problem in NP-hard as well, and hence the reduction in the proof of Theorem 4 directly implies the NP-hardness of feature cross search as well.

▶ **Corollary 14.** *The feature cross search problem is NP-hard.*

Similar proof to that of Theorem 4 implies the hardness of feature selection via label based mutual information maximization.

▶ **Theorem 15.** *There is no $n^{-1/\,\mathsf{poly}(\log\log n)}$-approximation algorithm for feature selection via label based mutual information maximization unless the exponential time hypothesis fails.*

Proof. Similar to Theorem 4 we prove this theorem via an approximation preserving reduction from k-densest subgraph. Consider the hard example provided in the proof of Theorem 4. Here we show that for any arbitrary set of features S if the induced subgraph of the corresponding vertices has ϕm edges, we have $I(S;C) = \phi$. We define a random variable X as follows. X is 0 if none or one of the features in S has a binary value, and X is 1 if two of the features in S have binary values. Note that the value of C is independent of X, and thus we have $I(S;C) = I(S;C|X)$. Hence, we have

$$\begin{aligned} I(S;C) = I(S;C|X) &= \mathbb{E}_X[I(S,C)|X] \\ &= \Pr[X=0](I(S;C)|X=0) + \Pr[X=1](I(S;C)|X=1)\,. \end{aligned}$$

Note that given $X = 0$, S and C are independent and hence we have $(I(S;C)|X=0) = 0$. On the other hand if $X = 1$, S uniquely defines C, and hence we have $(I(S;C)|X=1) = 1$. Therefore we have $I(S;C) = \mathbb{E}_X[I(S,C)|X] = \Pr[X=1] = \frac{\phi m}{m} = \phi$, as claimed. ◀

4 Reformulating Maximum AUC

Here, we prove Observation 3 and thereby reformulate the maximum AUC as an affine function of the total variation of the commutator of two probability measures. Furthermore, we show that the maximum AUC is achieved by a specific scoring function, *i.e.*, the log-likelihood ratio.

We start with some definitions. We define the log-likelihood ratio of an event E by $\mathcal{L}(E) = \log \frac{P_1[E]}{P_0[E]}$ provided that $P_0[E]P_1[E] \neq 0$, where $P_i[\cdot] = \Pr[\cdot|C=i]$. If $P_0[E] = 0$, the log-likelihood ratio $\mathcal{L}(E)$ is defined to be $+\infty$. If $P_1[E] = 0$, it is defined to be $-\infty$. As a result, the range of the log-likelihood ratio is the set of extended real numbers, denoted by $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. Proposition 16 shows that the maximum AUC is achieved by a specific scoring function, *i.e.*, the log-likelihood ratio $\mathcal{L}$. Here we abuse the notation and define the score $\mathcal{L}(x_A)$ assigned to each $x_A \in V_A$ to be $\mathcal{L}(X_A = x_A)$, where X_A and C are jointly sampled from $\mathcal{D}$. In other words, if we assign to each value in V_A a score accordingly, then the AUC is maximized.

▶ **Proposition 16.** *The log-likelihood ratio achieves the maximum AUC among all functions* $\sigma : V_A \to \overline{\mathbb{R}}$

$$\mathrm{AUC}_{\mathcal{L}}(A) = \max_{\sigma: V_A \to \overline{\mathbb{R}}} \mathrm{AUC}_{\sigma}(A)\,.$$

The above proposition is a folklore result. However, we provide a proof for completeness, and the proof steps are also used to prove Observation 3.

Proof of Proposition 16 and Observation 3. To prove the above proposition, it suffices to show that for any scoring function σ, its achieved AUC is no greater than that achieved by using the log-likelihood ratio as the scoring function. In other words, we aim to prove that for any $\sigma : V_A \to \overline{\mathbb{R}}$,

$$\mathrm{AUC}_{\mathcal{L}}(A) \geq \mathrm{AUC}_{\sigma}(A).$$

Recall Definition 1. The AUC given a scoring function σ can be described using i.i.d. random variables $(X_U^+, C^+), (X_U^-, C^-) \sim \mathcal{D}$. We can express this quantity using indicator functions as follows

$$\mathrm{AUC}_{\sigma}(A) = \mathbb{E}[\mathbf{1}\{\sigma(X_A^+) > \sigma(X_A^-)\} + \frac{1}{2}\mathbf{1}\{\sigma(X_A^+) = \sigma(X_A^-)\}|C^+ = 1, C^- = 0]\,.$$

Note that $\mathbf{1}\{\sigma(X_A^+) > \sigma(X_A^-)\} + \mathbf{1}\{\sigma(X_A^+) = \sigma(X_A^-)\} + \mathbf{1}\{\sigma(X_A^+) < \sigma(X_A^-)\} = 1$, we have

$$\begin{aligned}
\mathrm{AUC}_\sigma(A) &= \frac{1}{2} + \mathbb{E}\left[\frac{1}{2}\mathbf{1}\{\sigma(X_A^+) > \sigma(X_A^-)\} - \frac{1}{2}\mathbf{1}\{\sigma(X_A^+) < \sigma(X_A^-)\} | C^+ = 1, C^- = 0\right] \\
&= \frac{1}{2} + \frac{1}{2}\mathbb{E}\left[\mathbf{1}\{\sigma(X_A^+) > \sigma(X_A^-)\} - \mathbf{1}\{\sigma(X_A^+) < \sigma(X_A^-)\} | C^+ = 1, C^- = 0\right] \\
&= \frac{1}{2} + \frac{1}{2}\mathbb{E}\left[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0\right] . \qquad (6)
\end{aligned}$$

To maximize the AUC, we focus on the term $\mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0]$. By symmetrizing this quantity, we obtain the following equations.

$$\begin{aligned}
&\mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0] \\
&= \frac{1}{2}\left(\mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0] + \mathbb{E}[\mathrm{sign}\left(\sigma(X_A^-) - \sigma(X_A^+)\right) | C^+ = 0, C^- = 1]\right) \\
&= \frac{1}{2}\left(\mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0] - \mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 0, C^- = 1]\right) \\
&= \frac{1}{2}\sum_{x_A^+, x_A^- \in V_A} \left(P_1[x_A^+]P_0[x_A^-]\,\mathrm{sign}\left(\sigma(x_A^+) - \sigma(x_A^-)\right) - P_1[x_A^-]P_0[x_A^+]\,\mathrm{sign}\left(\sigma(x_A^+) - \sigma(x_A^-)\right)\right) . \\
&= \frac{1}{2}\sum_{x_A^+, x_A^- \in V_A} \left(P_1[x_A^+]P_0[x_A^-] - P_1[x_A^-]P_0[x_A^+]\right)\mathrm{sign}\left(\sigma(x_A^+) - \sigma(x_A^-)\right) . \qquad (7)
\end{aligned}$$

The above expression can be upper bounded by the total variation distance between $P_1 \times P_0$ and $P_0 \times P_1$.

$$\begin{aligned}
\mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0] &\le \frac{1}{2}\sum_{x_A^+, x_A^- \in V_A} \left|P_1[x_A^+]P_0[x_A^-] - P_1[x_A^-]P_0[x_A^+]\right| \\
&= d_{TV}(P_1^A \times P_0^A, P_0^A \times P_1^A) . \qquad (8)
\end{aligned}$$

Note that whenever $P_1[x_A^+]P_0[x_A^-]$ or $P_1[x_A^-]P_0[x_A^+]$ is non-zero, the log-likelihood ratios $\mathcal{L}(x_A^+)$ and $\mathcal{L}(x_A^-)$, as well as $\mathcal{L}(x_A^+)$-$\mathcal{L}(x_A^-)$, are well defined on $\overline{\mathbb{R}}$. Hence, if $P_1[x_A^+]P_0[x_A^-] - P_1[x_A^-]P_0[x_A^+] \neq 0$, one can show that

$$\mathrm{sign}(P_1[x_A^+]P_0[x_A^-] - P_1[x_A^-]P_0[x_A^+]) = \mathrm{sign}\left(\mathcal{L}(x_A^+) - \mathcal{L}(x_A^-)\right).$$

Consequently, all equality conditions in (8) can be achieved by using log-likelihood ratio as the scoring function, and we have

$$\begin{aligned}
\mathbb{E}[\mathrm{sign}\left(\sigma(X_A^+) - \sigma(X_A^-)\right) | C^+ = 1, C^- = 0] &\le \mathbb{E}[\mathrm{sign}\left(\mathcal{L}(X_A^+) - \mathcal{L}(X_A^-)\right) | C^+ = 1, C^- = 0] \\
&= d_{TV}(P_1^A \times P_0^A, P_0^A \times P_1^A) . \qquad (9)
\end{aligned}$$

Combining (6) and (9), we have the following bound that holds true for any σ,

$$\mathrm{AUC}_\sigma(A) \le \mathrm{AUC}_\mathcal{L}(A) = \frac{1}{2} + \frac{1}{2}d_{TV}(P_1^A \times P_0^A, P_0^A \times P_1^A) , \qquad (10)$$

which completes the proof. ◀

According to Proposition 16 and equation (10), Observation 3 directly follows.

5 Total Variation of Commutator of Probability Measures

In this section, we prove Proposition 7, which states that the total variation of commutator of probability measures is a monotone submodular set function.

▶ **Proposition 7** (Proof in Section 5). *Let U be a finite index set. Assume that for every $a \in U$, there are a pair of probability measures P_0^a and P_1^a on a common sample space V_a. For any $A \subseteq U$, define the set function $F : 2^U \to \mathbb{R}_{\geq 0}$ by*

$$F(A) = d_{TV}\left(\bigtimes_{a\in A} P_1^a \times \bigtimes_{a\in A} P_0^a, \bigtimes_{a\in A} P_0^a \times \bigtimes_{a\in A} P_1^a\right). \tag{4}$$

The set function F is monotone and submodular.

5.1 Monotonicty Part of Proposition 7

We first show the monotonicity part.

Proof of Proposition 7 (Monotonicity). Let A and B be two subsets of U such that $A \subseteq B$. For any $A \subseteq U$, let $P_i^A = \bigtimes_{a\in A} P_i^a$. Using the above notation, we have $P_i^B = P_i^A \times P_i^{B\setminus A}$. By the definition of F, we have

$$\begin{aligned}
&F(A)\\
&= d_{TV}\left(P_1^A \times P_0^A, P_0^A \times P_1^A\right)\\
&= \frac{1}{2}\sum_{x,y\in V_A}\left|P_1^A(x)P_0^A(y) - P_0^A(x)P_1^A(y)\right|\\
&= \frac{1}{2}\sum_{x,y\in V_A}\Bigg|\sum_{z,w\in V_{B\setminus A}} P_1^A(x)P_0^A(y)P_1^{B\setminus A}(z)P_0^{B\setminus A}(w)\\
&\quad - \sum_{z,w\in V_{B\setminus A}} P_0^A(x)P_1^A(y)P_0^{B\setminus A}(z)P_1^{B\setminus A}(w)\Bigg|\\
&\leq \frac{1}{2}\sum_{x,y\in V_A}\sum_{z,w\in V_{B\setminus A}}\left|P_1^A(x)P_0^A(y)P_1^{B\setminus A}(z)P_0^{B\setminus A}(w) - P_0^A(x)P_1^A(y)P_0^{B\setminus A}(z)P_1^{B\setminus A}(w)\right|\\
&= d_{TV}(P_1^A \times P_1^{B\setminus A} \times P_0^A \times P_0^{B\setminus A}, P_0^A \times P_0^{B\setminus A} \times P_1^A \times P_1^{B\setminus A})\\
&= d_{TV}\left(P_1^B \times P_0^B, P_0^B \times P_1^B\right)\\
&= F(B).
\end{aligned}$$

where the third equality is because

$$\sum_{z,w\in V_{B\setminus A}} P_1^{B\setminus A}(z)P_0^{B\setminus A}(w) = \sum_{z,w\in V_{B\setminus A}} P_0^{B\setminus A}(z)P_1^{B\setminus A}(w) = 1$$

and the inequality is a consequence of the triangle inequality. ◀

5.2 Submodularity Part of Proposition 7

To prove the submodularity part, we need the following lemmas.

▶ **Lemma 17** (General case, proof in the full version [5]). *Let $R, R' \in \Delta_{\Omega_1}$, $S, S' \in \Delta_{\Omega_2}$, and $P, Q \in \Delta_\Omega$, where $\Omega_1, \Omega_2, \Omega$ are finite sets. If $R \overset{f}{\sim} R'$ and $S \overset{g}{\sim} S'$, it holds that*

$$\begin{aligned} &d_{TV}(R \times S \times P \times Q, R' \times S' \times Q \times P) - d_{TV}(R \times P \times Q, R' \times Q \times P) \\ &-d_{TV}(S \times P \times Q, S' \times Q \times P) + d_{TV}(P \times Q, Q \times P) \leq 0\,. \end{aligned}$$

We begin with the Bernoulli case where Ω_1 and Ω_2 in its statement are both $\{0, 1\}$ so that R, R', S, S' are all probability measures of a Bernoulli random variable.

▶ **Lemma 18** (Bernoulli case, proof in the full version [5]). *Let $R, R', S, S' \in \Delta_{\{0,1\}}$ such that $R \overset{f}{\sim} R'$ and $S \overset{f}{\sim} S'$, where f is a function on $\{0, 1\}$ such that $f(0) = 1$ and $f(1) = 0$. Let $P, Q \in \Delta_\Omega$, where Ω is a finite sample space. The following inequality holds*

$$\begin{aligned} &d_{TV}(R \times S \times P \times Q, R' \times S' \times Q \times P) - d_{TV}(R \times P \times Q, R' \times Q \times P) \\ &-d_{TV}(S \times P \times Q, S' \times Q \times P) + d_{TV}(P \times Q, Q \times P) \leq 0\,. \end{aligned} \tag{11}$$

Proof sketch. To prove the Bernoulli case, we first show that under the summation (recall that according to Equation (1), the total variation distance is half of the L^1 distance, and the L^1 distance is the sum of the absolute value of the difference on each singleton), any term that involves an element of measure zero (with respect to P or Q) has no contribution to the expression on the left-hand side. We would like to emphasize that while the term itself may be non-zero, it will be canceled out under the summation. In our second step, we will consider quantities of the form $\sqrt{\frac{P(x)Q(y)}{Q(x)P(y)}}$ in which $Q(x)$ and $P(y)$ must be non-zero for all x and y. As a result, we have to eliminate elements of measure zero in first step by showing that their total contribution is zero.

As the second step, we perform a series of algebraic manipulations and substitutions and finally show that the opposite of left-hand side can be re-written as a quadratic $v^\top M v$, where v is a vector and M is a symmetric square matrix. Recall that the promised inequality claims that the left-hand side is non-positive (thus the opposite of the left-hand side is non-negative). Therefore, we will show it by establishing the positive semi-definiteness of M.

In fact, the matrix M is induced by a positive definite function. The problem of establishing the positive semi-definiteness of M reduces to the problem of proving that the function that induces M is positive definite. In light of the Bochner's theorem (see the full version [5]), we show its positive definiteness by computing its inverse Fourier transform, which turns out to be finite-valued and non-negative everywhere. ◀

The high-level strategy of proving Lemma 17 is to use Observation 19 to reduce the problem to the Bernoulli case (Lemma 18). The proof details can be found in the full version [5].

▶ **Observation 19.** *Let $P, P' \in \Delta_\Omega$ be such that $P \overset{f}{\sim} P'$ and $\phi : \mathbb{R}^2 \to \mathbb{R}$ be a homogeneous bivariate function, i.e., $\phi(\lambda x, \lambda y) = \lambda \phi(x, y)$ holds for any $x, y, \lambda \in \mathbb{R}$. For every element $x \in \Omega$, we define the Bernoulli probability measure U_x on $\{0, 1\}$ such that $U_x(1) = \frac{P(x)}{P(x)+P'(x)}$ and $U'_x(1) = \frac{P'(x)}{P(x)+P'(x)}$. The following equation holds*

$$\sum_{x \in \Omega} \phi(P(x), P'(x)) = \sum_{x \in \Omega} \frac{P(x) + P'(x)}{2} \left(\phi(U_x(1), U'_x(1)) + \phi(U'_x(1), U_x(1))\right)\,.$$

Before presenting the proof of Observation 19, we introduce the involutionary swapping lemma, which is also used in the proof of Lemma 17. Intuitively, the involutionary swapping lemma implies that two involution equivalent probability measures can be swapped inside a summation of a bivariate function.

▶ **Lemma 20** (Involutionary swapping lemma). *Let $P, P' \in \Delta_\Omega$ be such that $P \overset{f}{\sim} P'$ and $\phi : \mathbb{R}^2 \to \mathbb{R}$ be any bivariate function. Then we have*

$$\sum_{x\in\Omega} \phi(P(x), P'(x)) = \sum_{x\in\Omega} \phi(P'(x), P(x)) \,.$$

Proof. Under the assumption of the lemma statement, we have

$$\begin{aligned}
&\sum_{x\in\Omega} \phi(P(x), P'(x)) \\
&= \sum_{x\in\Omega} \phi(P'(f(x)), P(f(x))) \\
&= \sum_{x'\in\Omega} \phi(P'(x'), P(x')) \\
&= \sum_{x\in\Omega} \phi(P'(x), P(x)) \,.
\end{aligned}$$

The first equality is because for any $x \in \Omega$, we have $P(x) = P'(f(x))$ (by the definition of involution equivalence) and $P(f(x)) = P'(x)$ (Remark 10). The second equality is obtained by setting $x' = f(x)$ (this is because any involution map f is a bijection). The final equality is obtained by renaming x' to x. ◀

Proof of Observation 19. Under the assumption of the observation statement, we have

$$\begin{aligned}
\sum_{x\in\Omega} \phi(P(x), P'(x)) &= \frac{1}{2}\sum_{x\in\Omega} \phi(P(x), P'(x)) + \frac{1}{2}\sum_{x\in\Omega} \phi(P(x), P'(x)) \\
&= \frac{1}{2}\sum_{x\in\Omega} \phi(P(x), P'(x)) + \frac{1}{2}\sum_{x\in\Omega} \phi(P'(x), P(x)) \\
&= \sum_{x\in\Omega} \frac{P(x) + P'(x)}{2} \left(\phi(U_x(1), U'_x(1)) + \phi(U'_x(1), U_x(1))\right) \,.
\end{aligned}$$

We use Lemma 20 in the second term on the second line and the third equality is because ϕ is homogeneous. ◀

We are in a position to show the submodularity part, which follows from Lemma 17.

Proof of Proposition 7 (Submodularity). To show that F is submodular, we need to check its definition that for any $A \subseteq U$ and $a, b \in U \setminus A$ such that $a \neq b$, it holds that

$$F(A \cup \{a\}) + F(A \cup \{b\}) \geq F(A \cup \{a, b\}) + F(A) \,,$$

If we define $P_i^A = \bigtimes_{a\in A} P_i^a$, the above definition is equivalent to

$$\begin{aligned}
&d_{TV}\left(P_1^a \times P_0^a \times P_1^A \times P_0^A, P_0^a \times P_1^a \times P_0^A \times P_1^A\right) \\
&\quad + d_{TV}\left(P_1^b \times P_0^b \times P_1^A \times P_0^A, P_0^b \times P_1^b \times P_0^A \times P_1^A\right) \\
&\geq d_{TV}\left(P_1^a \times P_0^a \times P_1^b \times P_0^b \times P_1^A \times P_0^A, P_0^a \times P_1^a \times P_0^b \times P_1^b \times P_0^A \times P_0^A\right) \\
&\quad + d_{TV}\left(P_1^A \times P_0^A, P_0^A \times P_1^A\right) \,.
\end{aligned}$$

Re-arranging the terms yields

$$\begin{aligned} & d_{TV}\left(P_1^a \times P_0^a \times P_1^b \times P_0^b \times P_1^A \times P_0^A, P_0^a \times P_1^a \times P_0^b \times P_1^b \times P_0^A \times P_0^A\right) \\ - \; & d_{TV}\left(P_1^a \times P_0^a \times P_1^A \times P_0^A, P_0^a \times P_1^a \times P_0^A \times P_1^A\right) \\ - \; & d_{TV}\left(P_1^b \times P_0^b \times P_1^A \times P_0^A, P_0^b \times P_1^b \times P_0^A \times P_1^A\right) \\ + \; & d_{TV}\left(P_1^A \times P_0^A, P_0^A \times P_1^A\right) \leq 0\,. \end{aligned}$$

The above inequality follows from Lemma 17 if we set $P = P_1^A$, $Q = P_0^A$, $R = P_1^a \times P_0^a$, $R' = P_0^a \times P_1^a$, $S = P_1^b \times P_0^b$, and $S' = P_0^b \times P_1^b$. Note that $R \sim R'$ and $S \sim S'$ via the transpose involution (see Remark 11). ◀

6 Other Related Works

As discussed before, our problem falls in the category of feature engineering problems. Perhaps, the most studied problem in feature engineering is feature selection [10, 32, 33, 26, 11, 25, 18, 31]. In this problem, the goal is to select a small subset of the features to obtain a learning model with high accuracy and avoid over-fitting. Here we just mention a couple of feature selection algorithm related to submodular maximization and refer to [10] for an introduction to feature selection and many relevant references. [8] used the notion of weak submodularity to design and analyze feature selection algorithms. [16] used the submodularity of mutual information between the sensors to design a $(1-1/e)$-approximation algorithm for sensor placements, which can be directly used for feature selection. However, as we show in Theorem 15, it is not possible to design such algorithms to maximize the mutual information between the features and the label.

Another related well-studied problem in this domain is vocabulary compression [2, 7, 28, 1]. The goal of vocabulary compression is to improve the learning and serving time, and in some cases to avoid overfitting. Vocabulary compression can be done by simple approaches such as filtering and naive bucketing, or more complex approaches such as mutual information maximization. [1] and [28] used clustering algorithms based on the Jenson-Shannon divergence to compress the vocabulary of features. [7] proposed an iterative algorithm that locally maximizes the mutual information between a feature and the label. Recently, [2] considered this problem for binary labels and presented a quasi-linear-time distributed approximation algorithm to maximize the mutual information between the feature and the label. There are polynomial-time local algorithms for binary labels that maximize the mutual information [17, 13], studied in the context of discrete memoryless channels.

[30] designed an integer programming based algorithm for feature cross search and applied it to learn generalized linear models using rule-based features. They show that this approach obtains better accuracy compared to that of the existing rule ensemble algorithms. [20] proposed a greedy algorithm for feature cross search and show that the greedy algorithm works well on a variety of datasets. Neither of these papers provide any theoretical guarantees for the performance of their algorithm.

7 Conclusion

In this paper, we considered the problem of feature cross search. We formulated it as a problem of maximizing the normalized area under the curve (AUC) of the linear model trained on the crossed feature column. We first established a hardness result that no algorithm can provide $n^{1/\log\log n}$ approximation for this problem unless the exponential time hypothesis

fails. Therefore, no polynomial algorithm can solve this problem unless $\mathsf{P} = \mathsf{NP}$. In light of its intractable nature, we motivated and assumed the naïve Bayes assumption. We related AUC to the total variation of the commutator of two probability measures. Under the naïve Bayes assumption, we demonstrated that the aforementioned total variation is monotone and submodular with respect to the set of selected feature columns to be crossed. As a result, a greedy algorithm can achieve a $(1 - 1/e)$-approximation of the problem. Our proof techniques may be of independent interest. Finally, an empirical study showed that the greedy algorithm outperformed the baselines.

References

1 L Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM, 1998.

2 Mohammadhossein Bateni, Lin Chen, Hossein Esfandiari, Thomas Fu, Vahab Mirrokni, and Afshin Rostamizadeh. Categorical feature compression via submodular optimization. In *International Conference on Machine Learning*, pages 515–523, 2019.

3 Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $o(n^{1/4})$ approximation for densest k-subgraph. In *STOC*, pages 201–210. ACM, 2010.

4 Simon Byrne. A note on the use of empirical auc for evaluating probabilistic forecasts. *Electronic Journal of Statistics*, 10(1):380–393, 2016.

5 Lin Chen, Hossein Esfandiari, Gang Fu, Vahab S Mirrokni, and Qian Yu. Feature cross search via submodular optimization. *arXiv preprint arXiv:2107.02139*, 2021.

6 Yuxin Chen, S Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Conference on Learning Theory*, pages 338–363, 2015.

7 Inderjit S Dhillon, Subramanyam Mallela, and Rahul Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of machine learning research*, 3(Mar):1265–1287, 2003.

8 Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568, 2018.

9 Susana Eyheramendy, David D Lewis, and David Madigan. On the naive bayes model for text categorization. In *9th International Workshop on Artificial Intelligence and Statistics*. Citeseer, 2003.

10 Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

11 Nazrul Hoque, Dhruba K Bhattacharyya, and Jugal K Kalita. Mifs-nd: A mutual information-based feature selection method. *Expert Systems with Applications*, 41(14):6371–6385, 2014.

12 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

13 Ken-ichi Iwata and Shin-ya Ozawa. Quantizer design for outputs of binary-input discrete memoryless channels using smawk algorithm. In *2014 IEEE International Symposium on Information Theory*, pages 191–195. IEEE, 2014.

14 Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014.

15 Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 324–331. AUAI Press, 2005.

16 Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of*

the 5th international conference on Information processing in sensor networks, pages 2–10. ACM, 2006.

17 Brian M Kurkoski and Hideki Yagi. Quantization of binary-input discrete memoryless channels. *IEEE Transactions on Information Theory*, 60(8):4544–4552, 2014.

18 Nojun Kwak and Chong-Ho Choi. Input feature selection by mutual information based on parzen window. *IEEE transactions on pattern analysis and machine intelligence*, 24(12):1667–1671, 2002.

19 Hui Lin. *Submodularity in natural language processing: algorithms and applications*. PhD thesis, University of Washington, 2012.

20 Yuanfei Luo, Mengshuo Wang, Hao Zhou, Quanming Yao, Wei-Wei Tu, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. Autocross: Automatic feature crossing for tabular data in real-world applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.*, pages 1936–1945, 2019.

21 Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *STOC*, pages 954–961. ACM, 2017.

22 Tom Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.

23 Marko Mitrovic, Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Data summarization at scale: A two-stage submodular approach. In *ICML*, pages 3593–3602, 2018.

24 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

25 Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

26 Monica Rogati and Yiming Yang. High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM, 2002.

27 Henry Scheffé. A useful convergence theorem for probability distributions. *The Annals of Mathematical Statistics*, 18(3):434–438, 1947.

28 Noam Slonim and Naftali Tishby. The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research*, volume 1, page 200, 2001.

29 Burak Turhan and Ayse Bener. Analysis of naive bayes' assumptions on software fault data: An empirical study. *Data & Knowledge Engineering*, 68(2):278–290, 2009.

30 Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Günlük. Generalized linear rule models. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6687–6696, 2019.

31 Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963. PMLR, 2015.

32 Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for svms. In *Advances in neural information processing systems*, pages 668–674, 2001.

33 Sepehr Abbasi Zadeh, Mehrdad Ghadiri, Vahab Mirrokni, and Morteza Zadimoghaddam. Scalable feature selection via distributed diversity maximization. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

34 Yuanxing Zhang, Yichong Bai, Lin Chen, Kaigui Bian, and Xiaoming Li. Influence maximization in messenger-based social networks. In *GLOBECOM*, pages 1–6. IEEE, 2016.