A Unified Perspective on Regularization and Perturbation in Differentiable Subset Selection

Xiangqian Sun[1] **Cheuk Hang Leung**[2] **Yijun Li**[2] **Qi Wu**[2,†]
Department of Management Sciences[1]
School of Data Science[2]
City University of Hong Kong

Abstract

Subset selection, i.e., finding a bunch of items from a collection to achieve specific goals, has wide applications in information retrieval, statistics, and machine learning. To implement an end-to-end learning framework, different relaxed differentiable operators of subset selection are proposed. Most existing work relies on either *regularization method* or *perturbation method*. In this work, we provide a probabilistic interpretation for regularization relaxation and unify two schemes. Besides, we build some concrete examples to show the generic connection between these two relaxations. Finally, we evaluate the perturbed selector as well as the regularized selector on two tasks: the maximum entropy sampling problem and the feature selection problem. The experimental results show that these two methods can achieve competitive performance against other benchmarks.

1 INTRODUCTION

High-dimensional data has been pervasive nowadays across science and industry. Despite their appealing fine-grained details and high-quality properties, they also pose an unprecendent challenge in analyzing these datasets. For example, it could be computationally expensive to retrieve information from a high-resolution image dataset (Cordonnier et al., 2021). Therefore, subset selection methods that can reduce the dimensionality are of great importance to dealing with these high-dimensional data. For example, in feature extraction (Balın et al., 2019), one aims to select the most informative pixels to represent the original data; in neural machine translation, the beam search algorithm needs to find the k sequences of largest likelihood.

Unfortunately, subset selection operators cannot be integrated into an end-to-end learning framework due to their non-smoothness and thus non-differentiability. To this end, different differentiable relaxation methods for subset selection are proposed. There are mainly two types of relaxation methods: regularization relaxation and pertubation relaxation. Similar ideas can be found in other tasks, refer to Blondel et al. (2020); Berthet et al. (2020) and references therein.

Although there is a voluminous literature for relaxation methods, these differentiable operators of subset selection are investigated separately, either in a regularization scheme or a perturbation scheme. In this paper, we present a unified viewpoint for these two schemes. Regularization and perturbation play the same role in shifting the output from vertices to the interior of a feasible region and thus introduce smoothness and differentiation. In addition, it connects the regularization relaxation with a probabilistic model, which provides a different interpretation. To have a comprehensive understanding, we build two examples which link the perturbed distribution and the regularized entropy function. The first example is the Gumbel distribution of perturbation relaxation and the corresponding negative Shannon entropy. The second example is the connection between logistic distribution and the binary entropy function.

We propose two learning algorithms based on the two relaxation methods for maximum entropy sampling problem. As the only two end-to-end learning algorithms, they achieve competitive results compared to other methods. Finally, we evaluate two relaxations on the feature selection problem and achieve better performance.

Contribution. Our contribution is threefold:

1. We unify two differentiable relaxations, i.e., regularization relaxation and perturbation relaxation, for subset selection. This generic connection provides a

†Corresponding author. Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

probabilistic interpretation for regularized relaxation.

2. We build some concrete examples for both regularized relaxation and perturbed relaxation and show their equivalence.

3. To the best of our knowledge, we are the first to introduce perturbed top-k selector as well as regularized top-k selector to the maximum entropy sampling problem and feature selection problem.

This paper is organized as follows. In Section 2, we review related work from regularization and perturbation schemes. Section 3 introduces preliminary knowledge of differentiable subset selection. Section 4 introduces the main results, which reveal the generic connection between two schemes. Two learning algorithms for maximum entropy sampling problem (MESP) as well as applications of two relaxations to feature selection are introduced in Section 5.

Notations. Denote $[N] \coloneqq \{1, \dots, n\}$ and $S \subseteq [N]$. $|S|$ denotes the cardinality of S . Let $\mathcal{M}_p \coloneqq \{x \in \{0,1\}^n\}$ be the power set of $[N]$ and $\mathcal{M}_k \coloneqq \{x \in \{0,1\}^n | \mathbf{1}^\top x = k\}$ be the subset with fixed cardinality k. $\theta \in \Theta$ is the score vector; $\mathcal{J}$ denotes the Jacobian. We denote $H(\cdot)$ as entropy function; Z as perturbed error and $\tau > 0$ as the temperature parameter. $\mathrm{Diag}(x)$ is a diagonal matrix which has the vector x on its diagonal and I is the identity matrix. $\|\cdot\|_F$ represents Forbenius norm.

2 RELATED WORK

We review the related work from three broad sections: subset selection with regularized relaxation; subset selection with perturbed relaxation; and other subset selection methods.

Subset Selection with Regularized Relaxation. Regularization technique has been a pervasive approach across different subject areas, including structured prediction (Niculae and Martins, 2020; Blondel et al., 2020), dynamic programming (Mensch and Blondel, 2018), reinforcement learning (Geist et al., 2019) and sorting (Cuturi et al., 2019). Here, we restrict the review of this versatile method to subset selection only.

Amos et al. (2019) proposes the *Limited Multi-Label*(LML) projection layer based on constrained linear programming with binary entropy regularization. However, it proposes the binary entropy function as the regularizer only and does not introduce the temeprature parameter. The adaptive Euclidean projection of linear objective on (n, k)-simplex, which is equivalent to subset selection with ℓ_2 norm regularizer, is investigated in Kong et al. (2020). A similar idea can be found in sparsemax (Martins and Astudillo, 2016). Xie et al. (2020) proposes a differentiable top-k operator with negative Shannon entropy regularized. It is based on optimal transport with a designed cost matrix and marginal distribution. Petersen et al. (2022) proposes a family of differentiable top-k selector which considers multiple k.

Subset Selection with Perturbed Relaxation. Perturbation technique has been an important trick in a wide range of fields. A well-known example is the Gumbel-Max trick, which is applied in random choice models. Besides, it is also exploited in online learning and bandit problems, dubbed FTPL (*Follow the Perturbed Leader*) (Abernethy et al., 2014, 2016). Berthet et al. (2020) proposes general differentiable optimizer with perturbed relaxtion. Cordonnier et al. (2021) extends the idea of perturbed optimizer to top-k operator and applies it to image recognition.

Other Subset Selection Methods. Plötz and Roth (2018) proposes k-Nearest-Neighbors (kNN) which leverages self-similarity to sample k elements from n choices. To enable differentiation, kNN updates the logits by using the expected weight vector instead of the discrete samples. Xie and Ermon (2019) revisits the kNN and applies it to sentiment classfication problem. Kool et al. (2019) extends the idea of the Gumbel-Max trick to *Gumbel-Top-k* and uses *REINFORCE* to construct the gradient, which means it does not admit reparametrization. Struminsky et al. (2021) leverages recursive Gumbel-max trick to define distributions over structure domains and uses the score function estimator to estimate gradient. Hazimeh et al. (2021) develops a continuous relaxation for top-k selection by introducing a smooth-step function.

3 DIFFERENTIABLE SUBSET SELECTION

In this section, we introduce some preliminaries to subset selection and review the existing relaxation methods based on regularization and perturbation techniques.

3.1 Subset Selection

Subset selection aims to select a bunch of components to achieve targets. For example, maximum entorpy sampling problem aims to select the subset with the largest determinant. The cardinality of the subset can be fixed or not. In the following discussion, we fix the cardinality of the subset. Consider the subset $\mathcal{M} = \mathcal{M}_k$, and $\theta \in \mathbb{R}^n$ is a score vector, the *top-k selector* is defined as follows:

$$y(\theta) \coloneqq \underset{y \in \mathcal{M}}{\arg\max} \ \langle \theta, y \rangle \tag{1}$$

The solution of (1) is simple: it returns a k-hot vector which is the indices of θ's top-k elements. For simplicity, we

assume this vector is unique. We call it as the *hard selector* in contrast to the following relaxed selector. Obviously, the solution of (1) is discrete and non-differentiable. This problem hinders the end-to-end learning framework of most deep learning architectures.

3.2 Regularized Relaxation

As we discussed above, the top-k selector is a discrete and non-differentiable operator which cannot be embedded into downstream layers. To implement an end-to-end differentiable fashion, the *regularized top-k selector* is proposed by introducing a regularization term compared with (1). Let $\mathrm{conv}(\mathcal{M})$ be convex hull of $\mathcal{M}$ and $H(y)$ be an convex function, the *regularized top-k selector* is defined as follows

$$y_H(\theta;\tau) \coloneqq \underset{y\in\mathrm{conv}(\mathcal{M})}{\arg\max}\ \langle\theta, y\rangle - \tau H(y), \tag{2}$$

where $\tau > 0$ is the temperature parameter. With a slight abuse of notation, we omit τ and write $y_H(\theta;\tau)$ as $y_H(\theta)$.

We refer $H(y)$ in (2) as "entropy function" which stems from information theory. Here, it plays an important role in introducing smoothness and differentiability to (1). The objective of (2) involves two terms: a linear term $\langle\theta, y\rangle$ and a regularization term $-\tau H(y)$. The maximizer of the linear term $\langle\theta, y\rangle$ returns the vertices in $\mathcal{M}$. However, the maximizer of the regularization term $-\tau H(y)$, e.g. Shannon entropy, returns the uniform distribution over all vertices of $\mathcal{M}$. Consequently, introducing the regularization term $-\tau H(y)$ tends to shift the maximizer from vertices to the interior of $\mathcal{M}$. Figure 1a illustrates the main idea of the regularized top-k selector.

Here, we list different choices of entropy function.

- *Negative Shannon entropy*: $H(y) = \sum_i y_i \log y_i$. The solution to *regularized top-k selector* with *negative Shannon entropy* is

  $$y_H(\theta)_i = \Pi_{[0,1]}\big(e^{(\theta_i - u^*)/\tau - 1}\big),$$

  where $\Pi_{[0,1]}(x) \coloneqq \min(\max(0, x), 1)$ denotes projection of x on interval $[0, 1]$ and u^* satisfies

  $$\sum_{i=1}^{n} \Pi_{[0,1]}\big(e^{(\theta_i - u^*)/\tau - 1}\big) = k.$$

- *Binary entropy*: $H(y) = \sum_{i=1}^{n} y_i \log y_i + (1 - y_i)\log(1 - y_i)$. The solution to *regularized top-k selector* with *binary entropy* is

  $$y_H(\theta)_i = \frac{1}{1 + \exp(-\theta_i/\tau - \nu^\star)}, \tag{3}$$

  where $\nu^\star$ satisfies:

  $$\sum_{i=1}^{n} 1/(1 + \exp(-\theta_i/\tau - \nu^\star)) = k.$$

Differentiation. The proposed regularized relaxations provide a smooth operator and can serve as a forward pass in network architecture. However, these relaxations cannot be directly integrated into network architecture since the backward pass requires $\frac{\partial y_H(\theta)}{\partial \theta}$ which cannot be computed explicitly. It is difficult because the differentiation of $y_H(\theta)$ depends on ν^* which itself implicitly depends on θ. Therefore, the *implicit function theorem* is required to derive the gradient. Combining implicit differentiation with deep learning has been widely explored in Amos et al. (2017); Amos and Kolter (2017); Amos et al. (2019); Blondel et al. (2022).

The regularized top-k selector can be formualted as the maximizer of a constrained optimization problem. Define the constrained optimization problem as

$$\begin{aligned} \max_{\mathbf{0}\le y\le \mathbf{1}}\ & \langle\theta, y\rangle - \tau H(y) \\ \text{s. t.}\quad & \mathbf{1}^\top y = k. \end{aligned}$$

The Karush–Kuhn–Tucker (KKT) condition is

$$\begin{aligned} \theta - \tau\nabla H(y) + \mathbf{1}\nu = 0, \\ \mathbf{1}^\top y - k = 0. \end{aligned} \tag{4}$$

Denote (y^*, ν^*) as the optimal value, and take implicit differentiation on (4), we have

$$\begin{bmatrix} \tau\nabla^2 H(y^*) & -\mathbf{1} \\ -\mathbf{1}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}\nu \end{bmatrix} = \begin{bmatrix} \mathrm{d}\theta \\ \mathbf{0} \end{bmatrix}. \tag{5}$$

We can form the Jacobian $\frac{\partial y_H(\theta)}{\partial \theta}$ now. One can simply set $\mathrm{d}\theta = I$, and solve the equation (5), then $\mathrm{d}y$ would be the desired Jacobian. Refer to Amos and Kolter (2017) and Blondel et al. (2022) for more details.

3.3 Perturbed Relaxation

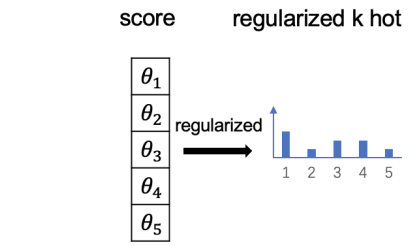
We introduce a perturbed relaxation to top-k selector, dubbed *perturbed top-k selector*. The perturbed top-k selector operator is defined as:

$$y_Z(\theta;\tau) \coloneqq \mathbb{E}[\ \underset{y\in\mathcal{M}}{\arg\max}\langle\theta + \tau Z, y\rangle\ ], \tag{6}$$

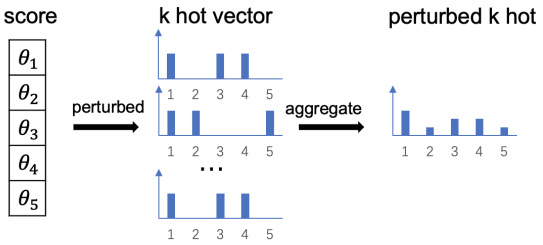
where Z is an $n-$dimensional random variable. We write $y_Z(\theta;\tau)$ as $y_Z(\theta)$ with a slight abuse of notation.

Notably, the feasible set of *perturbed top-k* inside the expectation is $\mathcal{M}$ while it is the convex hull of $\mathcal{M}$ in *regularized top-k*. Although for each sample, the perturbed top-k selector returns a discrete output, the expectation that aggregates all discrete outputs returns a smooth optimizer which takes value in the convex hull of $\mathcal{M}$. The idea of the perturbed top-k selector is demonstrated in Figure 1b.

Differentiation. The perturbed relaxation is widely investigated in online learning (Abernethy et al., 2014, 2016;



(a) regularized relaxation



(b) perturbed relaxation

Figure 1: Schemes of two relaxations.

Berthet et al., 2020). It is differentiable with a non-zero Jacobian and can be computed by integration by parts. The following theorem gives the Jacobian of $y_Z(\theta)$ at θ.

Theorem 1 (Abernethy et al. (2016)). *For noise Z with distribution $d\mu(z) \propto \exp(-\nu(z))dz$ and twice differentiable ν, the Jacobian matrix of $y_Z(\theta)$ at θ is*

$$\mathcal{J} = \mathbb{E}\left[\, y_Z(\theta)\nabla_z \nu(Z)^\top/\tau \right]. \tag{7}$$

Different distributions lead to different Jacobians. Here, we list some common choices of noise Z:

- *Gumbel Distribution:* The probability density function of $\mathrm{Gumbel}(0,1)$ is $f(z) = \exp(-z + \exp(-z))$. In this case, $\nu(z) = z - \exp(-z)$ and $\nu'(z) = 1 - \exp(-z)$.
- *Normal Distribution:* The probability density function of $N(0,1)$ is $f(z) = \frac{1}{\sqrt{2\pi}}\exp(-z^2/2)$. In this case, $\nu(z) = z^2/2$ and $v'(z) = z$.

In practice, it is infeasible to derive the explicit form of $y_Z(\theta)$ in (6), let alone the Jacobian in (7). Therefore, an approximation of the Monte Carlo estimator is proposed. Generally, it takes Z as $\mathrm{Gumbel}(0,1)$ distribution. To construct a Monte Carlo estimator, one first draws samples $\{z_i\}_{i=1}^M \overset{\text{i.i.d}}{\sim} \mathrm{Gumbel}(0,1)$. The Monte Carlo estimator of perturbed top-k selector is

$$\widehat{y}_Z(\theta) = \frac{1}{M}\sum_{i=1}^M \widetilde{y}_i(\theta), \tag{8}$$

where $\widetilde{y}_i(\theta) := \arg\max_{y\in\mathcal{M}}\langle \theta + \tau z_i, y\rangle$. Consequently, the Monte Carlo estimator of Jacobian is

$$\widehat{\mathcal{J}} = \frac{1}{M}\sum_{i=1}^M \widetilde{y}_i(\theta)(1-\exp(-z_i))^\top/\tau. \tag{9}$$

4 CONNECTION BETWEEN TWO RELAXATIONS

We have introduced regularized relaxation and perturbed selection separately. However, it seems there is some intrinsic connection between two relaxations when one takes a particular perturbed error and entropy function. We provide two examples to show this intrinsic connection.

Gumbel—Negative Shannon Entropy. Consider $\mathcal{M} = \mathcal{M}_1$ and $Z \overset{\text{i.i.d}}{\sim} \mathrm{Gumbel}(0,1)$, then

$$\begin{aligned} y_Z(\theta)_i &= \mathbb{E}[\ \arg\max_{y\in\mathcal{M}_1}\langle \theta + \tau Z, y\rangle\ ]_i \\ &= \mathbb{P}\left(\theta_i + \tau Z_i \geq \theta_j + \tau Z_j, \forall j \neq i\right), \\ &= \frac{\exp(\theta_i/\tau)}{\sum_{j=1}^n \exp(\theta_j/\tau)}. \end{aligned}$$

It is also known as *Gumbel-Max* trick (Maddison et al., 2014). Besides, if $\mathcal{M} = \mathcal{M}_1$ and $H(y) = \sum_i y_i \log y_i$, then we have

$$y_H(\theta) = \arg\max_{y\in\mathrm{conv}(\mathcal{M}_1)} \langle \theta, y\rangle - \tau\sum_i y_i \log y_i. \tag{10}$$

By the first order condition of (10), we have

$$y_H(\theta)_i = \frac{\exp(\theta_i/\tau)}{\sum_{j=1}^n \exp(\theta_j/\tau)}. \tag{11}$$

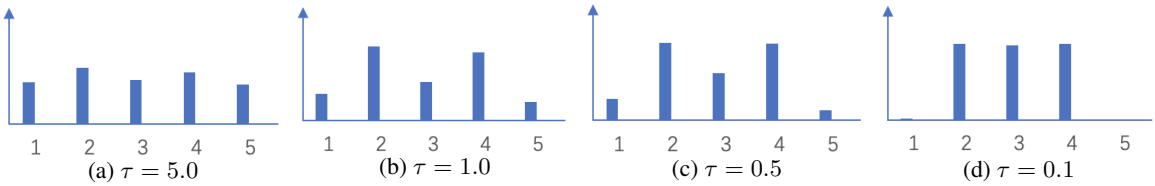
The above discussion illustrates that the regularized and perturbed selectors are the same. This connection has also been investigated in Berthet et al. (2020).

Logistic—Binary Entropy. Consider $\mathcal{M} = \mathcal{M}_p$, $Z \overset{\text{i.i.d}}{\sim}$ $\mathrm{Logistic}(0,1)$ and note that cumulative distribution function of $\mathrm{Logistic}(0,1)$ is $F(x) = 1/\big(1+\exp(-x)\big)$ then

$$\begin{aligned} y_Z(\theta)_i &= \mathbb{E}[\ \arg\max_{y\in\mathcal{M}_p}\langle \theta + \tau Z, y\rangle\ ]_i \\ &= \mathbb{P}\left(\theta_i + \tau Z_i \geq 0\right) \\ &= \frac{1}{1+\exp(-\theta_i/\tau)}. \end{aligned}$$

Besides, if $\mathcal{M} = \mathcal{M}_p$ and $H(y) = \sum_{i=1}^n y_i \log y_i + (1-y_i)\log(1-y_i)$, then

$$\begin{aligned} y_H(\theta) = &\arg\max_{y\in\mathrm{conv}(\mathcal{M}_p)} \langle \theta, y\rangle \\ &- \tau\sum_{i=1}^n \big(y_i \log y_i + (1-y_i)\log(1-y_i)\big). \end{aligned}$$



(a) $\tau = 5.0$ (b) $\tau = 1.0$ (c) $\tau = 0.5$ (d) $\tau = 0.1$

Figure 2: example of differentiable subset selection with decreasing temperature $\tau = [\ 5.0,\ 1.0,\ 0.5,\ 0.1\ ]$. Here, $\theta = [\ -0.6,\ 1.9,\ -0.2,\ 1.1,\ -1.0]$ and $k = 3$.

By the first order condition, we have

$$y_H(\theta)_i = \frac{1}{1+\exp(-\theta_i/\tau)}. \tag{12}$$

To the best of our knowledge, our work first establishes the connection between *logistic distribution* and *binary entropy function*.

Proposition 1. *Let* $\Omega(\theta) = \mathbb{E}[\ \max_{y\in\mathcal{M}}\langle\theta + Z, y\rangle\ ]$ *for some distribution* Z*, then* $y_H(\theta) = y_Z(\theta)\ \forall\theta \in \Theta$ *if and only if*

$$H = \Omega^\star + C, \tag{13}$$

where $\Omega^\star$ *denotes Fenchel conjugate of* Ω *and* $C \in \mathbb{R}$ *is a constant.*

While a sufficient condition of two relaxations has been studied before (Abernethy et al., 2014, 2016; Berthet et al., 2020), this proposition reveals not only a sufficient but also a necessary condition between two relaxations. Moreover, it builds a bridge between perturbed relaxation and regularized relaxation. It links the regularizer under regularization scheme with the random perturbed error under perturbation scheme. Besides, this provides a probabilistic interpretation for regularized relaxation. Equipped with this connection, the regularization relaxation can be interpreted as probability under a corresponding random noise induced by H.

We derive relaxation results for the Gumbel distribution and the Logistic distribution and show their equivalence above. Here, we revisit these results and prove them as corollaries of Proposition 1 .

Corollary 1. *If* $\mathcal{M} = \mathcal{M}_1$, $Z \overset{i.i.d}{\sim} \text{Gumbel}(0,1)$ *and* $H(y) = \sum_i y_i \log y_i$ *, then*

$$y_H(\theta) = y_Z(\theta).$$

Proof. Consider $\mathcal{M} = \mathcal{M}_1$, $Z \overset{\text{i.i.d}}{\sim} \text{Gumbel}(0,1)$, then

$$\begin{aligned}\Omega(\theta) &= \mathbb{E}[\ \max_{y\in\mathcal{M}_1}\langle\theta + Z, y\rangle\ ] \\ &= \log\sum_{i=1}^n \exp(\theta_i) + \gamma,\end{aligned}$$

where $\gamma \approx 0.577$ is Euler's constant. The Fenchel conjugate of Ω is

$$\begin{aligned}\Omega^\star(y) &= \sup_\theta\ \langle\theta, y\rangle - \Omega(\theta) \\ &= \sum_i y_i \log y_i,\end{aligned}$$

which is *negative Shannon entropy*, see Boyd et al. (2004) Example 3.25 for conjugate derivation. By applying Proposition 1, the proof is complete. □

Corollary 2. *If* $\mathcal{M} = \mathcal{M}_p$, $Z \overset{i.i.d}{\sim} \text{Logistic}(0,1)$ *and* $H(y) = \sum_{i=1}^n y_i \log y_i + (1-y_i)\log(1-y_i)$*, then*

$$y_H(\theta) = y_Z(\theta).$$

Proof. Consider $\mathcal{M} = \mathcal{M}_p$, $Z \overset{\text{i.i.d}}{\sim} \text{Logistic}(0,1)$,

$$\begin{aligned}\Omega(\theta) &= \mathbb{E}[\ \max_{y\in\mathcal{M}_p}\langle\theta + Z, y\rangle\ ] \\ &= \sum_{i=1}^n \mathbb{E}[\ \max\big(\theta_i + Z_i, 0\big)\ ] \\ &= \sum_{i=1}^n \log\big(1+\exp(\theta_i)\big).\end{aligned}$$

The Fenchel conjugate of Ω is

$$\begin{aligned}\Omega^\star(\theta) &= \sup_\theta\langle\theta, y\rangle - \Omega(\theta) \\ &= \sum_{i=1}^n y_i \log y_i + (1-y_i)\log(1-y_i),\end{aligned}$$

which is the *binary entropy*, see Appendix for conjugate derivation. The proof is completed as a consequence of Proposition 1 . □

These two corollaries provide another perspective on the equivalence of distribution and the entropy function. Next, we establish some common properties of two relaxations.

Proposition 2 (Properties of Two Relaxations)**.** *Suppose* Z *is an i.i.d random vector and Equation* (13) *is satisfied, then* $y_R(\theta) \in \{y_H(\theta), y_Z(\theta)\}$ *share some common properties:*

1. *(temperature limit) If $\tau \to 0$, then $y_R(\theta) \to y(\theta)$; if $\tau \to \infty$, then $y_R(\theta) \to k/n \cdot \mathbf{1}$.*
2. *(order preserving) If $\theta_i > \theta_j$, then $y_R(\theta)_i \geq y_R(\theta)_j$.*
3. *(permutation invariance) If P is a permutation matrix, then $y_R(P\theta) = Py_R(\theta)$.*
4. *(temperature scaling) $y_R(\theta; \tau) = y_R(\theta/\tau; 1)$.*

The *temperature limit* property describes the limiting case of two relaxations as the temperature parameter approaches zero or infty. Figure 2 illustrates relaxation results of the top-3 selector among five indices with decreasing τ. When τ is large ($\tau = 5.0$), the relaxed selector approaches an "uniform" selector: each index has equal value. While τ is small ($\tau = 0.1$), the relaxed selector approaches *hard selector* as in (1). The *order perserving* property is straightforward: the larger the score is, the larger the output is. Figure 2 demonstrates this property for any temperature. The *permutation invariannce* illustrates the symmetry of two relaxations. The *temperature scaling* property shows the effect of the temperature parameter is equivalent to scaling the score vector.

Finally, we highlight some differences. Although two relaxations can produce smooth and differentiable operators for the original hard operator, they manifest themselves in different forms. The regularized relaxation can produce an exact solution by solving a constrained optimization problem with a specifically designed entropy function. However, the perturbed relaxation, which produces a Monte Carlo estimator as given in (8), differs from the regularized ones. Moreover, the perturbed relaxation has more flexibility than the regularized relaxation since the choice of perturbation distribution could be any *exponential family distribution*. Although there is a connection between distribution and entropy function in (13), it is hard to derive the explicit form of Ω for a general distribution Z. Another difference between the two relaxations lies in the gradient. The gradient of regularized relaxation, which is exact, can be derived by implicit differentiation of the KKT condition. By contrast, the gradient of *perturbed relaxation* can only be approximated by random samples. One first draws *i.i.d* samples from the distribution Z and computes the Monte Carlo estimator as (9). The perturbed relaxation gradient is simple and easy to implement. However, it is also time-consuming to compute the Monte Carlo estimator whenever the dimension is high.

5 APPLICATION

In this section, we carry out experiments on two relaxations on two tasks: the maximum entropy sampling problem and the feature selection problem. We have made the code for our algorithm and experiments available on a public repository[1]

5.1 Maximum Entropy Sampling Problem

Maximum entropy sampling problem (MESP), which aims to select the most informative submatrix, has wide applications in meteorology, environmental statistics, and statistical geology. A typical example of MESP comes from spatial statistics: it aims to select s locations from which to collect the consequent data with data obtained from time-series observations of n environmental locations.

MESP can be defined as follows. Let C be a $n \times n$ positive semidefinite matrix. The goal is to select a $k \times k$ minor to maximize its logarithm of determinant. Formally,

$$\ell \coloneqq \max \big\{ \operatorname{ldet} C[S, S] : S \subset [N], |S| = k \big\}, \qquad (14)$$

where ldet denotes natural logarithm of the determinant and $C[S, S]$ denotes a principal submatrix of C having row and column index S. It is notable that the "entropy" from MESP represents the logarithm of the determinant of the submatrix, which is different from the "entropy" in *regularized Top-k* which refers to $H(y)$.

Maximum entropy sampling problem is proved to be NP-hard (Ko et al., 1995). It is impractical to explicitly list all elements of $[N]$ even if n and k are of moderate size. For example, for $n = 90$ and $k = 40$, it has around 6×10^{25} possibilities in total. The optimal solution to MESP requires the branch-and-bound method. Therefore, it is time-consuming to solve an MESP problem to optimality, even of moderate size. For instance, it takes 52.04 hours to optimality for $n = 90$ and $k = 40$ (Anstreicher, 2020). Therefore, deriving an efficient bound for MESP is of great importance.

It is not straightforward to construct a relaxation for MESP from (14). Fortunately, Anstreicher (2020) proposes an identity which can be utilized to directly construct a relaxation for MESP.

Lemma 2 (Anstreicher (2020))**.** *For a subset $S \subset [N]$, let C be a positive semidefinite matrix and $y = [y_1, \ldots, y_n]^\top$, where $y_i = 1$, if $i \in S$, and $y_i = 0$, if $i \in [N] \backslash S$. Then*

$$2 \operatorname{ldet} C[S, S] = \operatorname{ldet} \Big(C \operatorname{Diag}(y) C + I - \operatorname{Diag}(y) \Big). \qquad (15)$$

We can construct a relaxation for MESP by leveraging identity (15). Combining two relaxations of subset selection, our objective becomes to maximize:

$$\ell(\theta) = \frac{1}{2} \operatorname{ldet} \big(C \operatorname{Diag}(y_R(\theta)) C + I - \operatorname{Diag}(y_R(\theta)) \big),$$

[1]Code available at: https://github.com/xqsun4/subset-selction

where $y_R(\theta) \in \{y_H(\theta), y_Z(\theta)\}$. We present the pseudocode for regularized relaxation MESP as Algorithm 1 and perturbed relaxation MESP as Algorithm 2 .

Algorithm 1 Maximum Entropy Sampling with Regularized Relaxation

Input: C
Ouput: S
$\theta^0 \leftarrow \mathbf{0}$; ▷ initialize θ
$\widehat{y}(\theta^0) \leftarrow$ (3);
while $1 \leq t \leq T$ **do**
  $\ell(\theta^t) \leftarrow \text{ldet}\left(C\,\text{Diag}(\widehat{y})C + I - \text{Diag}(\widehat{y})\right)$;
  $\theta^{t+1} \leftarrow \theta^t + \alpha\nabla\ell(\theta^t)$;
  $\widehat{y}(\theta^{t+1}) \leftarrow$ (3);
end while
Return: $S = \text{top-k}(\theta)$. ▷ return index

Algorithm 2 Maximum Entropy Sampling with Perturbed Relaxation

Input: C
Ouput: S
$\theta^0 \leftarrow \mathbf{0}$; ▷ initialize θ
draws samples $\{z_i\}_{i=1}^M \overset{\text{i.i.d}}{\sim} \text{Gumbel}(0,1)$;
$\widehat{y}(\theta^0) \leftarrow$ (8);
while $1 \leq t \leq T$ **do**
  $\ell(\theta^t) \leftarrow \text{ldet}\left(C\,\text{Diag}(\widehat{y})C + I - \text{Diag}(\widehat{y})\right)$;
  $\theta^{t+1} \leftarrow \theta^t + \alpha\nabla\ell(\theta^t)$;
  draws samples $\{z_i\}_{i=1}^M \overset{\text{i.i.d}}{\sim} \text{Gumbel}(0,1)$;
  $\widehat{y}(\theta^{t+1}) \leftarrow$ (8);
end while
Return: $S = \text{top-k}(\theta)$. ▷ return index

We evaluate two algorithms on canonical datasets from the MESP literature with dimensions of $n = 90$ and $n = 124$. Two other sampling methods (Li and Xie, 2020) [2] are compared and the results are listed on Table 1 and Table 2. The optimal solution is obtained by the branch-and-bound method (Anstreicher, 2020). Our proposed methods achieve competitive results where they are slightly better than *sampling* method and slightly inferior to the *local search* method. However, these sampling methods depend on either spectral decomposition or Cholesky factorization, rather than an end-to-end manner. More details can be found in Appendix.

5.2 Feature Selection

Feature selection aims to identify a subset of data with the most informative information to achieve dimensionality reduction. Balın et al. (2019) proposes *Concrete Autoencoder* (CAE), which ia an end-to-end differentiable selection selection framework. It utilizes the reparametrization of *Concrete distribution* (Jang et al., 2017; Maddison et al., 2017). We implement two relaxations on the same feature selection structure and compare the performance of the two relaxations with that of the concrete layer.

[2] https://github.com/yongchunli-13/Approximation-Algorithms-for-MESP

Table 1: Maximum entropy sampling problem with $n = 90$.

k	Optimality	Sampling	Local Search	Regularized	Perturbed
10	58.532	58.521	58.532	57.882	57.882
20	111.482	111.207	111.482	110.885	110.885
30	161.539	160.884	161.539	161.285	161.11
40	209.969	208.757	209.958	209.403	208.943
50	257.16	255.736	257.154	256.715	256.821
60	303.019	301.474	303.008	302.498	301.782
70	347.471	345.861	347.453	347.071	345.435
80	389.997	389.002	389.997	389.926	389.411

Table 2: Maximum entropy sampling problem with $n = 124$.

k	Optimality	Sampling	Local Search	Regularized	Perturbed
20	77.827	77.726	77.826	77.106	77.106
30	106.7	105.843	106.7	105.95	105.95
40	131.055	128.988	131.055	129.832	129.832
50	149.498	145.831	149.498	148.272	148.428
60	164.012	157.955	163.916	160.545	159.773
70	172.528	165.816	172.528	167.288	166.36
80	175.091	167.898	175.091	166.201	167.147
90	171.262	160.425	171.262	170.008	160.304
100	162.865	155.592	162.865	153.483	151.976

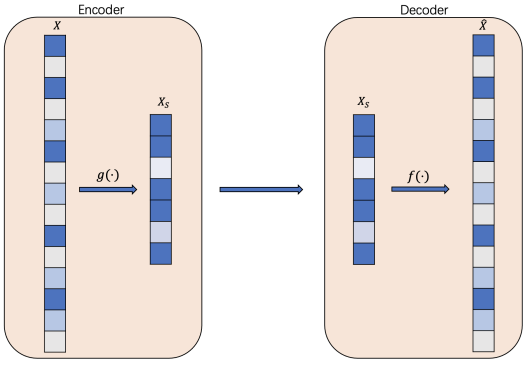


Figure 3: The model architecture of {Concrete, Perturbed, Regularized} Autoencoder.

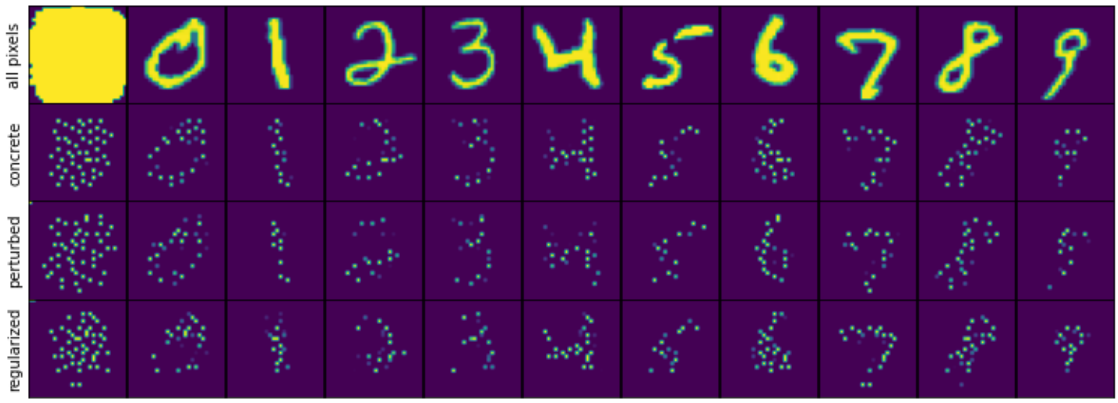


Figure 4: A visualization of MNIST (n = 784) feature selection with $k = 50$. The plots in the first row is original images of handwritten digits from zero to nine; the second (third, fourth) row plots the selected pixels and overlaped pixels for Concrete (Perturbed, Regularized) Autoencoder.

Table 3: Mean value of MNIST (n = 784) feature selection with $k = 50$.

criteria	Lap	AEFS	UDFS	MCFS	PFA	CAE	RAE	PAE
NMI (↑)	0.412	0.38	0.288	0.414	0.429	**0.492**	0.445	0.457
ACC (↑)	0.412	0.479	0.349	0.46	0.484	0.523	0.529	**0.535**
MSELR (↓)	0.072	0.068	0.077	0.064	0.049	0.024	**0.022**	**0.022**
MSE(↓)	0.176	0.122	0.185	0.138	0.086	0.057	**0.021**	**0.021**
ClASS (↑)	0.412	0.476	0.382	0.462	0.498	0.5	0.527	**0.531**
CLASSDT (↑)	0.62	0.785	0.648	0.801	0.848	0.894	**0.911**	0.91

The CAE employs an encoder-decoder architecture where the encoder is *concrete selector layer* and the decoder is used to reconstruct input via selected feature. Analoguesly, we propose *Perturbed Autoencoder* (PAE) and *Regularized Autoencoder* (RAE) which replace the concrete layer with perturbed relaxation and regularized relaxation. All subset selection layers are differentiable and can be incorporated into the end-to-end learning framework. We would like to optimize

$$\arg\min_{\theta,\phi} \mathbb{E}\big[\, \big\| f_\phi(X_S) - X \big\|_F \,\big], \quad \text{where } X_S = g_\theta(X).$$

The reconstruction error is designed as the Forbenius norm between the inputs and the outputs. We adopt the *Adam* optimizer with a learning rate of 10^{-3}. Besides, the same annealing schedule as Balın et al. (2019) for temperature parameter is adopted. The temeperature at epoch b is $\tau(b) = \tau_0(\tau_B/\tau_0)^{b/B}$, where τ_B is high temperature at beginning of training and τ_0 is the lower bound of temperature. Here, the initial temperature τ_B is 10.0 and the final temperature τ_0 is 0.01. The dataset is randomly devided into a 90-10 split to train and test the model. We evaluate the models on differnet criteria: *normalized mutual information* (NMI) (Li et al., 2017); accuracy by *K-means clustering* (ACC) (Li et al., 2017); MSE *by linear regression* (MSELR); MSE by *MLP* (MSE); accuracy of the *k-nearest neighbors classifier* (CLASS); accuracy of *extremely randomized trees classifier* (CLASSDT) Geurts et al. (2006).

From a theoretical pointview, the concrete distribution can be treated as the stochastic relaxation of corresponding *perturbed top-1 selector* (namely, the Gumbel-Max trick) or *regularized top-1 selector* (namely, softmax operator). See Figure 3 for the architecture of Concrete Autoencoder.

To manifest the capability of our relaxation selector, we apply the proposed methods to the MNIST dataset and test their performance. First, we run Concrete [3], Perturbed, Regularized} Autoencoder to select ex ante relaxed features . Then, an ex post hard selector is implemented. A visualization of the result is shown in Figure 4 . We compare our models with Lap (He et al., 2005), PFA (Lu et al., 2007), MCFS (Cai et al., 2010), UDFS (Yang et al., 2011), AEFS (Han et al., 2018)and CAE (Balın et al., 2019). We evaluate the informativeness of subsets extracted by these methods as well as other benchmarks. We repeat the experiments ten times and take the average results. Table 3 shows that our proposed methods outperform benchmarks. More details can be found in Appendix.

[3] https://github.com/mfbalin/Concrete-Autoencoders

6 CONCLUSION

In this paper, we build a generic connection between existing differentiable relaxation methods, which are based on the regularization method and the perturbation method. This connection also provides a probabilistic interpretation for regularized relaxations. We first introduce two relaxations on the maximum entropy sampling problem and test their performance with the Gumbel-Softmax trick in the feature selection problem.

As far as we know, there are two promising directions to explore. The connection we build is based on a subset selection problem. One possible direction could extend the idea to other strutural models. We first try these relaxation methods with MESP, which is an important combinatorial problem. However, these methods can only produce approximate results without any optimality guarantee. Another direction could integrate these relaxations to construct a more efficient branch-and-bound method to optimality.

Acknowledgements

Qi Wu acknowledges the support from The CityU-JD Digits Joint Laboratory in Financial Technology and Engineering; The Hong Kong Research Grants Council [General Research Fund 14206117, 11219420, and 11200219]; The CityU SRG-Fd fund 7005300, and The HK Institute of Data Science. The work described in this paper was partially supported by the InnoHK initiative, The Government of the HKSAR, and the Laboratory for AI-Powered Financial Technologies.

References

Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823. PMLR, 2014.

Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. Perturbation techniques in online learning and optimization. *Perturbations, Optimization, and Statistics*, 233, 2016.

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

Brandon Amos, Vladlen Koltun, and J Zico Kolter. The limited multi-label projection layer. *arXiv preprint arXiv:1906.08707*, 2019.

Kurt M Anstreicher. Efficient solution of maximum-entropy sampling problems. *Operations Research*, 68 (6):1826–1835, 2020.

Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *International Conference on Machine Learning*, volume 97, pages 444–453, 09–15 Jun 2019.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

Mathieu Blondel, André FT Martins, and Vlad Niculae. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 2022.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342, 2010.

Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021.

Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. *Advances in neural information processing systems*, 32, 2019.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. Autoencoder inspired unsupervised feature selection. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2941–2945. IEEE, 2018.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul

Mazumder, Lichan Hong, and Ed Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. *Advances in neural information processing systems*, 18, 2005.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

Weiwei Kong, Walid Krichene, Nicolas Mayoraz, Steffen Rendle, and Li Zhang. Rankmax: An adaptive projection alternative to the softmax function. *Advances in Neural Information Processing Systems*, 33:633–643, 2020.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pages 3499–3508. PMLR, 2019.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

Yongchun Li and Weijun Xie. Best principal submatrix selection for the maximum entropy sampling problem: scalable algorithms and performance guarantees. *arXiv preprint arXiv:2001.08537*, 2020.

Yijuan Lu, Ira Cohen, Xiang Sean Zhou, and Qi Tian. Feature selection using principal feature analysis. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 301–304, 2007.

C Maddison, A Mnih, and Y Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR 2017)*. International Conference on Learning Representations, 2017.

Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. *Advances in neural information processing systems*, 27, 2014.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pages 3462–3471. PMLR, 2018.

Vlad Niculae and Andre Martins. Lp-sparsemap: Differentiable relaxed optimization for sparse structured prediction. In *International Conference on Machine Learning*, pages 7348–7359. PMLR, 2020.

Felix Petersen, Hilde Kuehne, Christian Borgelt, and Oliver Deussen. Differentiable top-k classification learning. In *International Conference on Machine Learning*, pages 17656–17668. PMLR, 2022.

Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018.

Kirill Struminsky, Artyom Gadetsky, Denis Rakitin, Danil Karpushkin, and Dmitry P Vetrov. Leveraging recursive gumbel-max trick for approximate inference in combinatorial spaces. *Advances in Neural Information Processing Systems*, 34:10999–11011, 2021.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *IJCAI*, 2019.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. *Advances in Neural Information Processing Systems*, 33:20520–20531, 2020.

Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. L 2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI international joint conference on artificial intelligence*, 2011.

Supplementary Material: A Unified Perspective on Regularization and Perturbation in Differentiable Subset Selection

A Technical Results

A.1 Proof of Proposition 1

Proof. Without loss of generality, we set $\tau = 1$ (otherwise, let $\theta' = \theta/\tau$). We note

$$F(\theta) \coloneqq \max_{y \in \mathrm{conv}(\mathcal{M})} \langle \theta, y \rangle - H(y) \tag{1}$$

It is maximized at $\nabla_\theta F(\theta) = y_H(\theta)$. By the definition of Fenchel-Rockafellar duality (see Wainwright et al. (2008), Appendix A) and $\Omega(\theta) \coloneqq \mathbb{E}[\ \max_{y \in \mathcal{M}} \langle \theta + Z, y \rangle\ ]$,

$$\Omega(\theta) = \max_{y \in \mathrm{conv}(\mathcal{M})} \langle \theta, y \rangle - \Omega^\star(y) = \max_{y \in \mathrm{conv}(\mathcal{M})} \langle \theta, y \rangle - H(y) + C = F(\theta) + C \tag{2}$$

Because the expectation has a unique maximizer with probability one, we can swap the expectation and gradient (Bertsekas, 1973)

$$y_Z(\theta) = \mathbb{E}[\ \arg\max_{y \in \mathcal{M}} \langle \theta + Z, y \rangle\ ] = \nabla \Omega(\theta) = \nabla_\theta F(\theta) = y_H(\theta) \tag{3}$$

The inverse direction is just straightforward. Since $y_Z(\theta) = \nabla \Omega(\theta)$, $y_H(\theta) = \nabla_\theta F(\theta)$, and

$$y_Z(\theta) = y_H(\theta), \ \ \forall \theta \in \Theta, \tag{4}$$

then

$$H = \Omega^\star + C. \tag{5}$$

□

A.2 Conjugate Derivation

Lemma 1. *Let Z be the standard logistic distribution, then*

$$\mathbb{E}[\ \max(\theta + Z, 0)\ ] = \log(1 + \exp(\theta)). \tag{6}$$

Proof. $X = \theta + Z$ is also logistic distribution with location $\mu = \theta$ and its CDF is

$$F(x) = \frac{1}{1 + \exp(-(x - \theta))}, \tag{7}$$

then

$$\int x dF(x) = \frac{x \exp(-(x - \theta))}{1 + \exp(-(x - \theta))} - \log(1 + \exp(-(x - \theta))) + C.$$

$$\mathbb{E}[\ \max(\theta + Z, 0)\ ] = \int_0^\infty x dF(x) = \log(1 + \exp(\theta)). \tag{8}$$

□

Lemma 2. *The Fenchel conjugate of* $\Omega(\theta) = \sum_{i=1}^{n} \log(1 + \exp(\theta_i))$ *is*

$$\Omega^\star(x) = \sum_{i=1}^{n} x_i \log x_i + (1 - x_i) \log(1 - x_i). \tag{9}$$

Proof. First, we consider the Fenchel conjugate of $f(\theta) = \log(1 + \exp(\theta))$. The Fenchel conjugate of $f(\theta)$ is defined as

$$f^\star(x) = \sup_{\theta} \langle x, \theta \rangle - f(\theta). \tag{10}$$

By the first order condition,

$$x - \frac{\exp(\theta)}{1 + \exp(\theta)} = 0 \implies \theta = \log(x) - \log(1 - x) \tag{11}$$

which indicates $0 < x < 1$. Substituting into Eq. (10),

$$f^\star(x) = x \log x + (1 - x) \log(1 - x) \tag{12}$$

By applying the property of Fenchel conjugate of seperate functions, we obtain

$$\Omega^\star(x) = \sum_{i=1}^{n} x_i \log x_i + (1 - x_i) \log(1 - x_i). \tag{13}$$

This completes the proof. □

A.3 Regularized Relaxation

This part introduces the derivation of maximizer of different regularized optimization problem.

Binary Entropy: $H(y) = \sum_{i=1}^{n} y_i \log y_i + (1 - y_i) \log(1 - y_i)$. The *binary entropy regularized optimization* (BERO) problem becomes

$$\begin{aligned} \max \quad & \langle \theta, y \rangle - \left(\sum_{i=1}^{n} y_i \log y_i + (1 - y_i) \log(1 - y_i) \right) \\ \text{s.t.} \quad & \mathbf{1}^\top y = k, \\ & 0 \le y_i \le 1, \forall i = 1, \dots, n. \end{aligned} \tag{BERO}$$

It is obvious that BERO is a convex and contrained optimization problem. The solution to BERO is:

$$y_H(\theta)_i = \frac{1}{1 + \exp(-\theta_i - \nu^\star)} \tag{14}$$

where $\nu^\star$ satisfies:

$$\sum_{i=1}^{n} \frac{1}{1 + \exp(-\theta_i - \nu^\star)} = k \tag{15}$$

Negative Shannon Entropy: $H(y) = \sum_{i=1}^{n} y_i \log y_i$. The *relative entropy regularized optimization* (RERO) problem is as follows

$$\begin{aligned} \max \quad & \langle \theta, y \rangle - \sum_{i=1}^{n} y_i \log y_i \\ \text{s.t.} \quad & \mathbf{1}^\top y = k, \\ & 0 \le y_i \le 1, \forall i = 1, \dots, n. \end{aligned} \tag{RERO}$$

The solution to the RERO is given by:

$$y_H(\theta)_i = \Pi_{[0,1]}\big(\exp(\theta_i - \lambda^*)\big), \quad i = 1, \dots, n. \tag{16}$$

where $\Pi_{[0,1]}(x) \coloneqq \min(\max(0, x), 1)$ and λ^* satisfies:

$$\sum_{i=1}^{n} \Pi_{[0,1]}\big(\exp(\theta_i - \lambda^*)\big) = k \tag{17}$$

B Experimental Details

B.1 Maximum Entropy Sampling Problem

Maximum entropy sampling problem (MESP) has wide applications in meteorology, environmental statistics, and statistical geology. For example, Figure 1 from Anstreicher (2020) illustrates the 90 temperature monitoring stations in the Pacific Northwest of the United States. And the data from these 90 monitoring stations constitutes a 90×90 non-singular matrix. Another example is covariance matrices ($n = 124$) coming from an application to re-designing an environmental monitoring network.

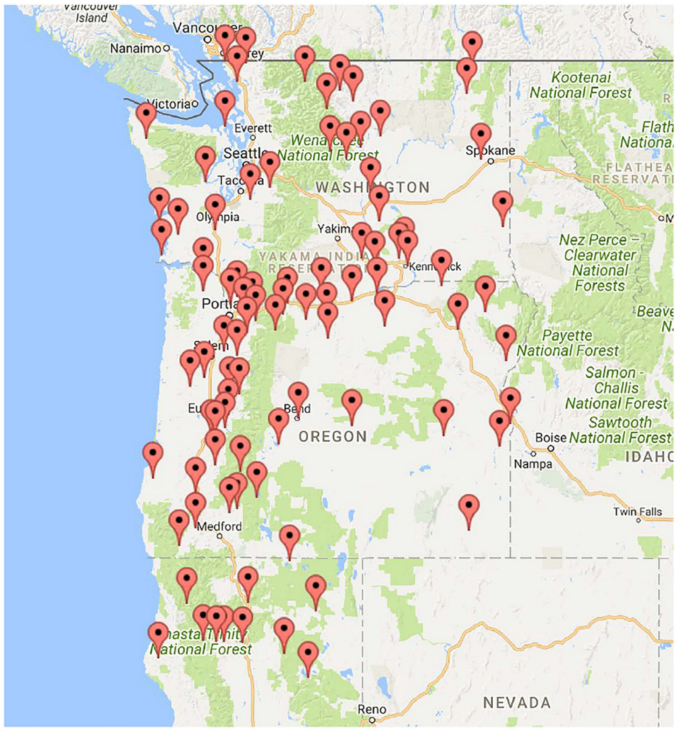


Figure 1: Locations of monitoring stations for matrix with $n = 90$ (Anstreicher, 2020)

Formally, MESP can be defined as follows. Let C be a $n \times n$ positive semidefinite matrix. The goal is to select a $k \times k$ minor to maximize its logarithm of determinant. Formally,

$$\ell(C, k) := \max \left\{ \operatorname{ldet} C[S, S] : S \subset [N], |S| = k \right\}, \tag{18}$$

Anstreicher (2020) proposes an identity which can be used to construct an upper bound, dubbed "Linx bound". For a subset $S \subset [N]$, let C be a positive semidefinite matrix, $y = [y_1, \ldots, y_n]^\top$, $y_i = 1$, if $i \in S$, and $y_i = 0$, if $i \in [N] \backslash S$, then

$$2 \operatorname{ldet} C[S, S] = \operatorname{ldet} \Big(C \operatorname{Diag}(y) C + I - \operatorname{Diag}(y) \Big). \tag{19}$$

Moreover, it is also easy to check that

$$\ell(C, k) = \ell(\gamma C, k) - k \log \gamma,$$

where the scale factor $\gamma > 0$. Therefore, a scaled objective function is

$$\ell(\theta) = \frac{1}{2} \operatorname{ldet} \left(\gamma C \operatorname{Diag}(y) C + I - \operatorname{Diag}(y) \right) - \frac{k}{2} \log \gamma. \tag{20}$$

The optimal scale factor γ for relaxations depends on C and k. We propose a schedule $\gamma = e^{ak+b}$, where $a = 0.057$ and $b = -4.157$.

MESP with Regularized Relaxations. For regularized relaxation, we take the binary entropy $H(y) = \sum_{i=1}^{n} y_i \log y_i + (1 - y_i) \log(1 - y_i)$ as regularizer. The regularized optimizer is $y_H(\theta)_i = \frac{1}{1+\exp(-\theta_i - \nu^\star)}$ given in (14). We make use of available implementation[1] of LML (Amos et al., 2019).

MESP with Perturbed Relaxations. For perturbed relaxation, we take perturbed error as Gumbel distribution and approximate it by the Monte Carlo estimator

$$\widehat{y}_Z(\theta) = \frac{1}{M} \sum_{i=1}^{M} \widetilde{y}_i(\theta), \tag{21}$$

where $\{z_i\}_{i=1}^{M} \overset{\text{i.i.d}}{\sim} \text{Gumbel}(0, 1)$ and $\widetilde{y}_i(\theta) = \arg\max_{y \in \mathcal{M}} \langle \theta + \tau z_i, y \rangle$. We take the available implementation [2] of perturbed optimizer Blondel et al. (2020) and set the linear programming as *top-k selector*.

B.2 MNIST Feature Selection

Balın et al. (2019) proposes *Concrete Autoencoder* (CAE), which ia an end-to-end differentiable selection selection framework. CAE employs an encoder-decoder architecture where the encoder is *concrete selector layer* and the decoder is used to reconstruct input via selected feature.

Analogously, we propose *Perturbed Autoencoder* (PAE) and *Regularized Autoencoder* (RAE) which replace the concrete layer with perturbed relaxation and regularized relaxation. All subset selection layers are differentiable and can be incorporated into the end-to-end learning framework. We would like to optimize

$$\arg\min_{\theta, \phi} \mathbb{E}\left[\left\| f_\phi(X_S) - X \right\|_F \right] \quad \text{where } X_S = g_\theta(X)$$

The reconstruct network is designed to be MLP. The reconstruction error is designed as the Forbenius norm between the inputs and the outputs. We adopt the *Adam* optimizer with a learning rate of 10^{-3}. Besides, the same annealing schedule as Balın et al. (2019) for temperature parameter is adopted. The temeperature at epoch b is $\tau(b) = \tau_0(\tau_B/\tau_0)^{b/B}$, where τ_B is high temperature at beginning of training and τ_0 is the lower bound of temperature. Here, the initial temperature τ_B is 10.0 and the final temperature τ_0 is 0.01.

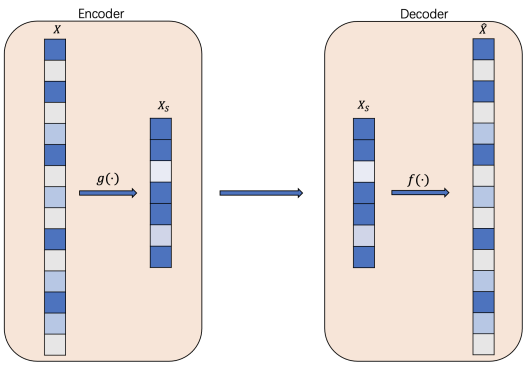


Figure 2: The model architecture of {Concrete, Perturbed, Regularized} Autoencoder.

The dataset is randomly devided into a 90-10 split to train and test the model. We evaluate the models on differnet criterias: *normalized mutual information* (NMI) (Li et al., 2017); accuracy by *K-means clustering* (ACC) (Li et al., 2017); mean squared error by *linear regression* (MSELR); mean squared error by *MLP* (MSE); accuracy of the *k-nearest neighbors classifier* (CLASS); accuracy of *extremely randomized trees classifier* (CLASSDT) Geurts et al. (2006).

[1] https://github.com/locuslab/lml

[2] https://github.com/google-research/google-research/tree/master/perturbations