

Large Database Compression Based on Perceived Information

Thomas Maugey *Member, IEEE*, Laura Toni *Senior Member, IEEE*

Abstract—Lossy compression algorithms trade bits for quality, aiming at reducing as much as possible the bitrate needed to represent the original source (or set of sources), while preserving the source quality. In this letter, we propose a novel paradigm of compression algorithms, aimed at minimizing the *information loss perceived by the final user* instead of the actual source quality loss, under compression rate constraints. As main contributions, we first introduce the concept of perceived information (PI), which reflects the information perceived by a given user experiencing a data collection, and which is evaluated as the volume spanned by the sources features in a *personalized* latent space. We then formalize the rate-PI optimization problem and propose an algorithm to solve this compression problem. Finally, we validate our algorithm against benchmark solutions with simulation results, showing the gain in taking into account users' preferences while also maximizing the perceived information in the feature domain.

Index Terms—Data compression, Large database, Sampling, Repurposing.

I. INTRODUCTION

The era of data explosion we live in has led to cutting edge findings in big data analysis and deep learning algorithms but at an expensive cost in terms of data storage. Storage growth is exceeding even the highest estimates with no sign of it slowing down anytime soon: 2.5 quintillion bytes of data are created each day at our current pace [1], and it will only accelerate with the advent of IoTs, volumetric videos, and new sensors. The storage burden has been partially alleviated by state-of-the-art compression algorithms, which can substantially reduce the amount of bits needed to store one or multiple sources, *e.g.*, end-to-end learning-based image compression algorithms to minimize the compression rate [2], MPEG standards to ensure exploitation of spatial and temporal correlation [3], joint source compression [4], [5], [6], [7]. All these coding strategies have led to impressive compression ratio, which however will be scaling always with the number of sources. However, to contain the upcoming avalanche of data, there is the need for a much drastic compression rate, which cannot be reached till the ultimate goal of the compression algorithm is to represent *each* original source with high fidelity.

In this letter, we aim at addressing this challenge by proposing a new paradigm-shift for compression algorithm aimed instead at preserving a global information perceived by the final user. We define this information as *perceived information* (PI). Sources should be compressed in such a

way that the *information of interest for the final user* – rather than per source information – is preserved. With this aim in mind, we proposed a first solution in the case of the encoder being a sampling algorithm. To achieve this goal, we first introduce the PI metric as the volume spanned by the sources features in a personalized latent space, *i.e.*, feature domain distorted by the user preferences. Then, we formalize our PI-based compression problem as a selection of the subset of sources that maximizes PI under sample size constraints and we propose an adaptive sampling algorithm to solve it. The latter selects for each user a subset of sources, which is the most representative of the original database, in terms of features most preferable by the user. Finally, we evaluate the performance of the proposed algorithm via simulation results, proving its gain against baseline algorithms taking into account user's preference or source redundancy disjointly.

To the best of our knowledge, this is the first work that proposes a compression algorithm in such a way that *i)* not all sources are preserved at the decoder side, *ii)* a global information about the dataset perceived by the user (PI) is however preserved during the compression, *iii)* this information is tailored based on users preferences. With respect to classical sampling strategies [8], [9], [10], which usually aims at retrieving the input data or at accomplishing a given task, our goal is different as we do not minimize the per-source reconstruction error, but rather the *perceived information loss* knowing that *the entire information will not be reconstructed*. The partial preservation of the original sources is typical of summarization algorithms [11], [12], [13], [14], which select sources to provide an overview of the original database. This means balancing source-quality (relevance of each source to database overview) and the subset-diversity (source redundancy, *i.e.*, maximization of the feature domain covered by the subset). Our algorithm has a fundamental difference: the subset is not used to give an overview of the database content but to preserve the information spanned by the sampled database in the latent space. This translates in an algorithm that balances *features-perceived quality* (how relevant each feature is to the user) and *features-diversity* (how well features are represented within the selected subset).

II. PI - BASED COMPRESSION

We now formulate the PI-based compression problem in its general shape, then we provide the specific formulation with the encoder being a sampling algorithm of the input space. Given the set of sources (dataset) $\mathcal{X} = \{X_1, \dots, X_N\}$ stored on a given server and consumed by a user u , each user

Thomas Maugey is with Inria Rennes Bretagne-Atlantique, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France, e-mail: thomas.maugey@inria.fr. Laura Toni is with Electronic and Electrical Engineering Department at University College London, Gower Street, London, UK, e-mail: l.toni@ucl.ac.uk.

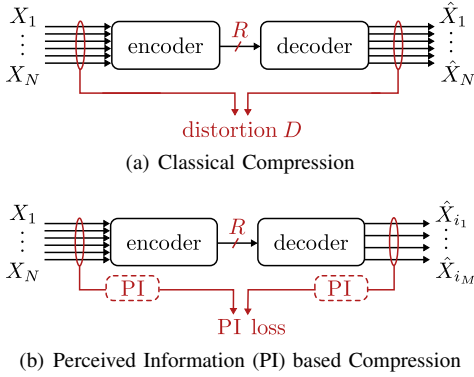


Fig. 1: Comparison between a classical source compressor and a PI-based compressor.

perceives the same data set differently, based on his/her own preferences or interests. Let us denote by $\pi_u(\mathcal{X})$ the perceived information (PI), which measures the information carried by \mathcal{X} and perceived by user u . Let us then define the encoder and decoder functions ϕ and ξ , respectively, as $\phi : \mathcal{X} \mapsto z$, and $\xi : z \mapsto \mathcal{Y}$, with z being the compressed vector, and $\mathcal{Y} = (\phi \circ \xi)(\mathcal{X})$ being the set of decoded sources. For user u , the compression algorithm is defined as follows

$$(\phi, \xi)_u^* : \arg \min_{\phi, \xi} \{[\pi_u(\mathcal{X}) - \pi_u((\phi \circ \xi)(\mathcal{X}))] + \lambda R(\phi(\mathcal{X}))\} \quad (1)$$

with $R(z)$ being the rate of the encoded vector z , λ the regularizing term, and $[\pi_u(\mathcal{X}) - \pi_u(\mathcal{Y})]$ is the PI-loss. In the degenerative case in which \mathcal{Y} is a distorted version of \mathcal{X} and all user preferences are uniform across features, converging to a classical compression algorithm or autoencoder that minimizes the per source distortion loss (Fig. 1(a)). In this letter, we are interested in the case in which a different source \mathcal{Y} is reconstructed and the user-perceived information is maximized, Fig. 1(b). As starting point, we propose a sampling algorithm in which only a subset $\mathcal{Y} \subseteq \mathcal{X}$ of the source database is preserved, while minimizing the degradation of the user perception of \mathcal{X} ¹. Namely, ϕ represents the sampling algorithm, z is the subset of selected sources, and $\mathcal{Y} = \xi(z) = z$ is an identity function. For a user u , we seek \mathcal{Y}_u^*

$$\mathcal{Y}_u^* : \arg \min_{\mathcal{Y} \subseteq \mathcal{X}} \{[\pi_u(\mathcal{X}) - \pi_u(\mathcal{Y})] + \lambda |\mathcal{Y}|\}. \quad (2)$$

The key intuition of the above problem formulation is to create a different goal for compression algorithms: the encoding rate can be translated into the subset cardinality $|\mathcal{Y}|$ and the tradeoff between features-quality and features-diversity is now reflected by the PI loss. While mathematically simple, the optimization problem in (2) hides two key and unsolved challenges: (i) how to define the PI explicitly, (ii) how to sample the \mathcal{X} space efficiently with respect to this metric.

III. PERCEIVED INFORMATION

Properties and goals: First, we define the key properties that the PI should respect:

¹Without loss of generality, we focus mainly on the effect of the source selection on the PI. Therefore coding artifacts are not taken into account in the following as they would affect equally the database or the subset.

1) $\pi_u(\mathcal{Y})$ is non-decreasing with the size of \mathcal{Y} :

$$\text{For any item } X, \quad \pi_u(\mathcal{Y}) \leq \pi_u(\mathcal{Y} \cup \{X\})$$

2) PI metric is user dependant:

$$\exists u \neq u' \quad \text{such that} \quad \pi_u(\mathcal{Y}) \neq \pi_{u'}(\mathcal{Y}).$$

Property 1) states that adding items to a dataset cannot decrease the PI. Property 2) states that two different users do not necessarily appreciate equally the same dataset.

Considering the items of \mathcal{Y} as continuous random variables and identifying the PI with the differential entropy of \mathcal{Y} is a natural first choice. The main limitation is however that the differential entropy would fail in respecting Property 1). Specifically, the entropy maximization is equivalent to the maximization of the volume spanned by the covariance matrix of the items in the dataset. This volume however can be decreasing with the cardinal of \mathcal{Y} (when correlated items are added for example), as shown in determinantal point processes (DPP) algorithms [15]. To overcome this limitation, we propose a metric that reflects the volume spanned by the covariance matrix of the features (instead of the sources). The feature matrix has a key role in describing the meta-information, *i.e.*, the information in the latent space, instead of the information in the source domain. Finally, the spanned volume needs to also take into account the users preferences to be compliant to Property 2). These considerations lead to one of the key contribution of this paper, which is the definition of the PI metric provided in the following subsection.

Proposed PI metric: The items in \mathcal{Y} are generally very complex objects, whose statistical behavior is difficult to model. We assume that a kernel function, $\kappa : X \mapsto \mathbf{b}$, maps each of the item X into a D dimensional vector \mathbf{b} , living in the so-called *latent space*. The vector \mathbf{b} thus describes the item X with a set of D features. In this work, we assume that such a kernel exists. Depending on applications, several ways might be considered to define such a function, *e.g.*, source modeling, distribution parametrization, or even more recently CNN architectures. We refer readers to [16], [17], [18] for more details. We consider now that the feature vector \mathbf{b} is the realization of a multivariate random variable B of dimension D . We model the PI as the differential entropy of the random variable B , whose distribution $p_{\mathcal{Y},u}$ depends on the sample set \mathcal{Y} and the user u :

$$\pi_u(\mathcal{Y}) = h(B) = - \int_{\mathbf{b} \in \mathbb{R}^D} p_{\mathcal{Y},u}(\mathbf{b}) \log_2 p_{\mathcal{Y},u}(\mathbf{b}) d\mathbf{b}. \quad (3)$$

We specify the general definition in the equation above assuming that B follows a multivariate Gaussian distribution $p_{\mathcal{Y},u} = \mathcal{N}(\mu, \Sigma_{\mathcal{Y}}^u)$, where $\Sigma_{\mathcal{Y}}^u$ is the covariance matrix corresponding to the subset of source \mathcal{Y} and the user u . We define this *personalized* covariance matrix as:

$$\Sigma_{\mathcal{Y}}^u = \mathbf{U} \mathbf{B}_{\mathcal{Y}} \mathbf{B}_{\mathcal{Y}}^{\top} \mathbf{U} \quad (4)$$

where $\mathbf{B}_{\mathcal{Y}}$ is a $D \times M$ matrix containing the M feature vectors of the items in the subset $\mathcal{Y} \subset \mathcal{X}$, with $M = |\mathcal{Y}|$. Furthermore, \mathbf{U} is a diagonal matrix, with each $u_j \in]0, 1]$ reflecting the user's preference for feature j , with 0 meaning that the feature is not interesting for the user and 1 meaning that the feature

must be preserved in the sampled collection. In this work, we consider that these scores are a-priori known. This is without loss of generality as users' preference can also be learned, *e.g.*, in recommendation systems [19], [20], [21]. In the proposed $D \times D$ personalized covariance matrix, each entry depicts the covariance between the features weighted by the user's score, given a sampled set \mathcal{Y} .

From (3) and (4), we define the PI metric as:

$$\pi_u(\mathcal{Y}) = \frac{D}{2} \log_2 (2\pi e |\Sigma_{\mathcal{Y}}^u|) = \frac{D}{2} \log_2 (2\pi e |\mathbf{U}\mathbf{B}_{\mathcal{Y}}\mathbf{B}_{\mathcal{Y}}^{\top}\mathbf{U}|). \quad (5)$$

This formulation is based on the determinant of a covariance matrix, which can be interpreted as the volume spanned by the sampled set in the latent space [8] and this volume is non-decreasing with the sample size given the definition in (4), thus matching with Property 1). Furthermore, we consider a personalized covariance matrix, in which the user's preferences described by \mathbf{U} express the relative importance of each feature in this volume evaluation. These weights ensures that the PI metric favors sources with popular features (*i.e.*, having high features for the ones that are highly scored by the users). Taking the determinant of this weighted (or personalized) covariance ensures a penalization when features are highly represented by multiple sources. Hence, the PI captures the features-perceived quality and features-diversity tradeoff, given user's preference. Therefore, different users will perceive different PI values, respecting Property 2).

While PI definition in (5) might appear similar to other sampling formulations such as DPP [22] or max-entropy [8], there is a fundamental difference. The former formulations defines the covariance matrix as $\mathbf{B}_{\mathcal{Y}}^{\top}\mathbf{B}_{\mathcal{Y}}$ (or $\mathbf{B}_{\mathcal{Y}}^{\top}\mathbf{U}\mathbf{U}\mathbf{B}_{\mathcal{Y}}$) with dimension $|\mathcal{Y}| \times |\mathcal{Y}|$ (*i.e.*, subset dimension). As a consequence, the matrix's dimension grows with the size of the sampled set, and leads to a null volume when the size is larger than the rank of the matrix (*i.e.*, $M \geq D$), leading to covariance matrices that would not be compliant to our Property 1). These properties are a consequence of the covariance quantifying the quality and diversity tradeoff among the sources of the subset. Therefore, while DPP-like formulation is fully compatible with summarization tasks, this is not the case for compression tasks (given our interpretation of compression). On the contrary, our formulation considers a covariance with dimension $|\mathcal{D}| \times |\mathcal{D}|$, independent on the subset size. Moreover, our matrix reflects the covariance of each feature across the sampled set instead of the covariance of each source across the latent space.

IV. PI-BASED COMPRESSION ALGORITHM

In this section, we describe our proposed algorithm aimed at optimizing the rate-PI compression curve. Specifically, replacing the PI definition (5) in (2) and imposing the total subset rate as hard constraint, we get the following problem formulation²:

$$\mathcal{Y}_u^* : \arg \max_{\mathcal{Y} \subseteq \mathcal{X}} \frac{D}{2} \log_2 (2\pi e |\mathbf{U}\mathbf{B}_{\mathcal{Y}}\mathbf{B}_{\mathcal{Y}}^{\top}\mathbf{U}|) \quad \text{s.t.} \quad |\mathcal{Y}| \leq M. \quad (6)$$

²Under the hard constraint on $|\mathcal{Y}|$, minimizing the PI loss is equivalent to maximizing the PI.

Algorithm 1 Proposed PI-compression algorithm

Input: $\mathcal{X}, \mathbf{B}, \mathbf{U}, K, \lambda$
Output: \mathcal{Y}

```

 $\mathcal{Y} \leftarrow \emptyset$ 
/*Compute the popularity  $\mathbf{p}$  (of size  $1 \times N$ ) of item of the database*/
 $\mathbf{p} = \text{diag}(\mathbf{B}^{\top}\mathbf{U}\mathbf{B})$ 
/*Initialize the similarity vector  $\mathbf{s}$  (of size  $1 \times N$ )*/
 $\mathbf{s} \leftarrow \mathbf{0}^{\top}$ 
for  $k = 1$  to  $K$  do
  /*Set the probability  $\pi$  of the items that is at a tradeoff between the popularity and and the dissimilarity*/
   $\pi \leftarrow \sqrt{\mathbf{p}} - \lambda\mathbf{s}$ 
   $\pi \leftarrow \pi / \sum_{j=1}^N \pi_j$ 
  /*Randomly choose one item with the probability  $\pi$  */
   $l \leftarrow \text{RandomSelect}(\mathcal{X} \setminus \mathcal{Y} \mid \pi_{\mathcal{X} \setminus \mathcal{Y}})$ 
   $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{l\}$ 
  /*Adjust the popularity and the similarity */
   $\mathbf{s} \leftarrow \frac{1}{k} \left( \sum_{i=1}^k \mathbf{B}_{\mathcal{Y}(i)} \right)^{\top} \mathbf{B}$ 
end for

```

To maximize the determinant $|\mathbf{U}\mathbf{B}_{\mathcal{Y}}\mathbf{B}_{\mathcal{Y}}^{\top}\mathbf{U}|$, items need to be selected in such a way that $\mathbf{U}\mathbf{B}_{\mathcal{Y}}\mathbf{B}_{\mathcal{Y}}^{\top}\mathbf{U}$ has the largest diagonal elements and the minimal off diagonal terms. The diagonal terms are maximal when most popular items belongs to the subset \mathcal{Y} . The popularity of an item with a feature \mathbf{b} is given by $\|\mathbf{U}\mathbf{b}\|_2^2$. The off-diagonal terms depict the correlations between the features in the subset \mathcal{Y} . Said differently, it measures the correlations between the rows of $\mathbf{B}_{\mathcal{Y}}$. The proposed algorithm randomly selects the items of the database. The probability of each item in \mathcal{X} is set as the trade-off between the popularity of the item and its similarity with the chosen ones. The proposed algorithm is described in Algorithm 1.

V. EXPERIMENTS AND CONCLUSION

Dataset: We simulate a database of $N = 2000$ items. Without loss of generality, we assume each item is characterized by features defined into a latent space with dimensionality $D = 20$. Items features are generated with a normal distribution centered around a dominant feature, selected uniformly at random between 6 possible values. This results in the feature matrix \mathbf{B} depicted in Fig. 2(a). Items can be clustered based on their mean value, Fig. 2(b). Finally, we assume a final user with higher a preference toward red and yellow colors, as shown in Fig. 2(c).

Baseline methods: for further validation, we compare our algorithm with 3 sampling methods. *Max-Proba* selects items based on user's preferences only (features-perceived quality) without taking into account the redundancy between the selected items (features-diversity trade-off). This method picks the M most popular items, *i.e.*, having the maximum user's score measured by the diagonal elements of $\mathbf{B}^{\top}\mathbf{U}\mathbf{B}$. *Random* picks M items randomly, each of them having a probability given by the corresponding diagonal elements of $\mathbf{B}^{\top}\mathbf{U}\mathbf{B}$. This method also relies on the popularity of each item. However, the randomness of the selection enables some dissimilarities between the items. *DPP* [15][23] is a very popular sampling algorithm used for summarization. This algorithm enables to sample M elements that maximize the $\log_2(|\mathbf{B}^{\top}\mathbf{U}\mathbf{B} + \delta\mathbf{I}_N|)$. The role of δ is to guarantee that the rank of the kernel is greater than M , which is mandatory for DPP, as already discussed in Sec. III.

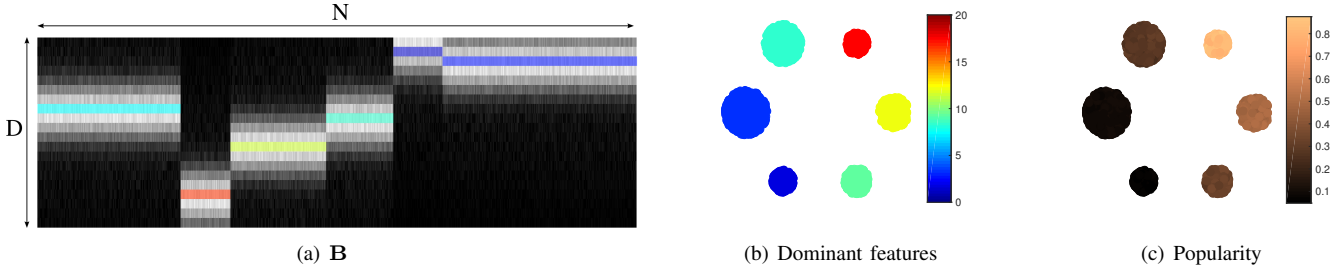


Fig. 2: Synthetic dataset for experiments. (a) the $D \times N$ matrix \mathbf{B} containing the features of each item of the dataset (black = 0, white = 1, color = the dominant feature in the colormap of (b)), (b) the dataset’s embedding and the representation of the dominant value for each item, (c) the popularity of each item, when features with larger indices (the reddish) are preferred.

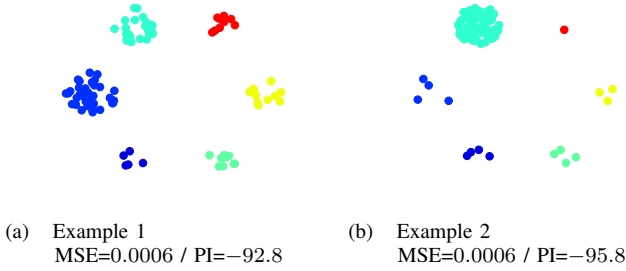


Fig. 3: Two examples of sampled sets \mathcal{Y} .

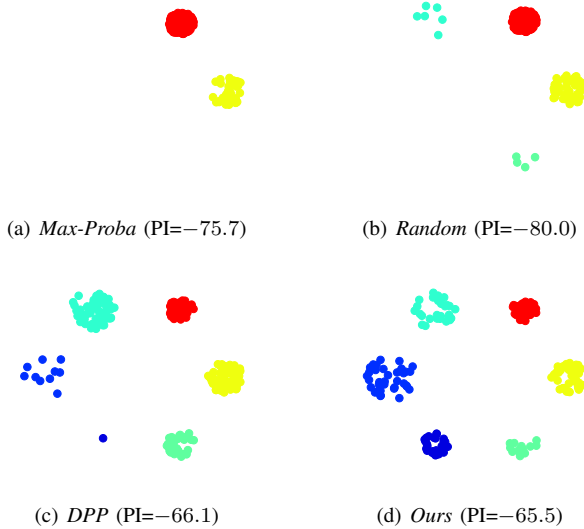


Fig. 4: Sampled set \mathcal{Y} obtained with the different algorithms.

Importance of PI metric: We first show the importance of the PI metric, compared to metrics widely used in sampling algorithms. For comparison, we consider the global distortion (mean squared error, MSE) of the recovered dataset [24], [25], [26], evaluated with the interpolation algorithm in [27], [28]. We compare the PI and the global distortion on two possible sampled sets, shown in Fig. 3, with the left one having sources sampled more uniformly across cluster. If we consider for a moment that all items are equally popular, Example 1 is a more representative sampling of the database. However, from

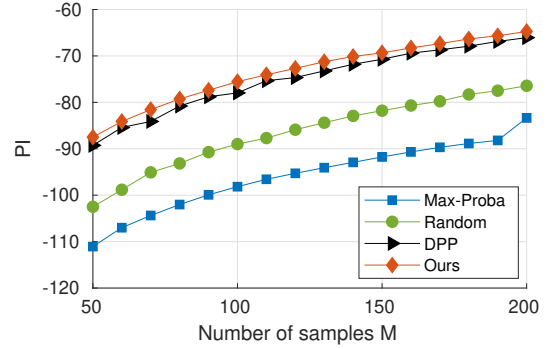


Fig. 5: (M -PI) comparison.

the figure, we observe that this is reflected by the PI metric (the higher the better) but not from the global distortion one, which is the same for both sampled sets.

Results: We consider the dataset described in Fig. 2 with item popularity depicted in Fig. 2(c) sampled with both the proposed method and the baseline ones. We depict the sampled set in Fig. 4 when $M = 200$ items (10% of the database size) are sampled and we provide the corresponding PI values in each subfigure caption. We can visually observe that the items selected by *Max-Prob* all correspond to the most popular features. This favor features-perceived quality but with no preservation of the features-diversity, leading to poor PI values. The visual aspect of the features when the items are selected with the *Random* approach is similar, but with some improved diversity aspect respect to *Max-Prob*, leading to higher a PI value. More interestingly, the DPP enables to achieve a good trade-off between diversity and popularity of the selected items, with the limitation that when $M > D$ the methods converge to a random sampling (because of the $\delta \mathbf{I}$ term that is added to the kernel). On the contrary, our method enables to maintain this good trade-off between features-perceived quality and diversity even when $M \gg D$. This is reflected by a higher PI. In Fig. 5, we provide more quantitative results showing the PI values as a function of the sample size M (when $M > D$). The figure confirms the gain of the proposed algorithm across different values of M . Moreover, the convexity shape of the (M -PI) curve shows that the PI is greatly compacted within few samples, potentially enabling huge compression rate able to cope with data explosion.

REFERENCES

- [1] [Online]. Available: <https://www.domo.com/learn/data-never-sleeps-8>
- [2] J. Ascenso, P. Akyazi, F. Pereira, and T. Ebrahimi, "Learning-based image coding: early solutions reviewing and subjective quality evaluation," in *Optics, Photonics and Digital Technologies for Imaging Applications VI*, vol. 11353. International Society for Optics and Photonics, 2020, p. 113530S.
- [3] J. Chen, M. Karczewicz, Y.-W. Huang, K. Choi, J.-R. Ohm, and G. J. Sullivan, "The joint exploration model (JEM) for video compression with capability beyond HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [4] Z. Shi, X. Sun, and F. Wu, "Photo album compression for cloud storage using local features," *IEEE Journal on emerging and selected topics in circuits and systems*, vol. 4, no. 1, pp. 17–28, 2014.
- [5] X. Zhang, Y. Zhang, W. Lin, S. Ma, and W. Gao, "An inter-image redundancy measure for image set compression," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2015, pp. 1274–1277.
- [6] L. Sha, W. Wu, and B. Li, "Novel image set compression algorithm using rate-distortion optimized multiple reference image selection," *IEEE Access*, vol. 6, pp. 66 903–66 913, 2018.
- [7] X. Liu, G. Cheung, C.-W. Lin, D. Zhao, and W. Gao, "Prior-based quantization bin matching for cloud storage of JPEG images," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3222–3235, 2018.
- [8] K. M. Anstreicher, M. Fampa, J. Lee, and J. Williams, "Maximum-entropy remote sampling," *Discrete Applied Mathematics*, vol. 108, no. 3, pp. 211–226, 2001.
- [9] S. P. Chepuri and G. Leus, "Graph sampling for covariance estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 451–466, 2017.
- [10] P. M. Baggerstoss, "Uniform manifold sampling (ums): Sampling the maximum entropy pdf," *IEEE Transactions on Signal Processing*, vol. 65, no. 9, pp. 2455–2470, 2017.
- [11] J.-H. Lim, Q. Tian, and P. Mulhem, "Home photo content modeling for personalized event-based retrieval," *IEEE MultiMedia*, vol. 10, no. 4, pp. 28–37, 2003.
- [12] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder-decoder networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [13] A. Sharghi, A. Borji, C. Li, T. Yang, and B. Gong, "Improving sequential determinantal point processes for supervised video summarization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 517–533.
- [14] A. Belhadji, R. Bardenet, and P. Chainais, "A determinantal point process for column subset selection," *arXiv preprint arXiv:1812.09771*, 2018.
- [15] A. Kulesza and B. Taskar, "k-DPPs: Fixed-size determinantal point processes," *ICML*, 2011.
- [16] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE transactions on neural networks*, vol. 12, no. 2, pp. 181–201, 2001.
- [17] D. Charte, F. Charte, S. Garcia, M. J. del Jesus, and F. Herrera, "A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines," *Information Fusion*, vol. 44, pp. 78–96, 2018.
- [18] L. Kuhnel, T. Fletcher, S. Joshi, and S. Sommer, "Latent space nonlinear statistics," *arXiv preprint arXiv:1805.07632*, 2018.
- [19] M. Naumov, D. Mudigere, H.-J. M. Shi, J. Huang, N. Sundaraman, J. Park, X. Wang, U. Gupta, C.-J. Wu, A. G. Azzolini *et al.*, "Deep learning recommendation model for personalization and recommendation systems," *arXiv preprint arXiv:1906.00091*, 2019.
- [20] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, "Svms classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features," *Knowledge-Based Systems*, vol. 141, pp. 80 – 91, 2018.
- [21] X. Yu, F. Jiang, J. Du, and D. Gong, "A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains," *Pattern Recognition*, vol. 94, pp. 96 – 109, 2019.
- [22] A. Kulesza, B. Taskar *et al.*, "Determinantal point processes for machine learning," *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012.
- [23] N. Tremblay, P.-O. Amblard, and S. Barthelmé, "Graph sampling with determinantal processes," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1674–1678.
- [24] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [25] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.
- [26] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory;? pub _newline=""?" *IEEE transactions on signal processing*, vol. 63, no. 24, pp. 6510–6523, 2015.
- [27] I. Pesenson, "Variational splines and paley–wiener spaces on combinatorial graphs," *Constructive Approximation*, vol. 29, no. 1, pp. 1–21, 2009.
- [28] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.