Universal Compression of a Mixture of Parametric Sources with Side Information

Ahmad Beirami, *Member, IEEE,* Liling Huang, Mohsen Sardari, and Faramarz Fekri, *Senior Member, IEEE*

***Abstract*—This paper investigates the benefits of the side information on the universal compression of sequences from a mixture of K parametric sources. The output sequence of the mixture source is chosen from the source $i \in \{1, \ldots, K\}$ with a d_i-dimensional parameter vector at random according to probability vector $\mathbf{w} = (w_1, \ldots, w_K)$. The average minimax redundancy of the universal compression of a new random sequence of length n is derived when the encoder and the decoder have a common side information of T sequences generated independently by the mixture source. Necessary and sufficient conditions on the distribution $\mathbf{w}$ and the mixture parameter dimensions $\mathbf{d} = (d_1, \ldots, d_K)$ are determined such that the side information provided by the previous sequences results in a reduction in the first-order term of the average codeword length compared with the universal compression without side information. Further, it is proved that the optimal compression with side information corresponds to the clustering of the side information sequences from the mixture source. Then, a clustering technique is presented to better utilize the side information by classifying the data sequences from a mixture source. Finally, the performance of the clustering on the universal compression with side information is validated using computer simulations on real network data traces.**

***Index Terms*—Universal Lossless Compression; Side Information; Mixture Source; Clustering.**

I. Introduction

UNIVERSAL compression aims at reducing the average number of bits required to describe a sequence from an unknown source from a family of sources, while good performance is desired for most of the sources in the family [4]–[12]. However, it often needs to observe a very long sequence so that it can effectively learn the existing patterns in the sequence for efficient compression. Therefore, universal compression performs poorly on relatively small sequences [13], [14] where sufficient data is not available for learning of the statistics and training of the encoder. On the other hand, the presence of side information at the decoder has proven to be useful in several source coding applications (cf. [15]–[17] and the references therein). In particular, the impact of side information on *universal* compression has also been shown to be useful (cf. [12], [18]–[20]). However, to the best of the authors' knowledge, the problem of the universal compression of a mixture of parametric sources with side information has not been explored in the literature.

The recent rapid growth in the network traffic has motivated new research directions to leverage the existing correlations in the sequences (network packets) in order to reduce the traffic. These solutions must be transparent to the user and the application and hence must reside on the network layer, where the correlated sequences in the network flow are present [21]–[24]. As network packets are relatively small, universal compression solutions (if employed naively) do not result in much traffic reduction [1], [13], [23], [24]. Further, the existing universal compression schemes do not exploit the cross correlation among the packets destined to different users. As such, recently, we proposed universal compression of network packets using network memory in [23], [24], where the common memory between the encoder server (or router) and the decoder router was used as the side information to improve the performance of universal compression on network packets. As each packet may be generated by a different source, a realistic modeling of the network traffic requires to consider the content server to be a mixture source [1]. This motivates us to study the universal compression of sequences from a mixture source using common side information between the encoder and the decoder. With a different motivation, Krishnan and Baron recently proposed a MDL-based parallel universal compression algorithm to exploit the cross correlation among the packets [25], [26].

In [20], [27], we derived the optimal universal compression performance with side information for a single source, i.e., $K = 1$; we proved that significant improvement is obtained from the side information in the universal compression of small sequences when sufficiently large side information is available. It was shown that a few megabytes of side information can drive the sequence length very close to the entropy. On the other hand, it is natural to expect that network packets that can be observed on a router are generated by a mixture of parametric sources. Motivated by this fact, in this paper, we extend the setup of the memory-assisted universal compression to a mixture of K parametric sources. Although the problem formulation is inspired from the network traffic compression, universal compression of a mixture source with side information finds applications in a wide variety of problems, such as data storage systems, and migration of virtual machines, where the compression of data before transmission is desirable.

A. Beirami was with the School of Electrical and Computer Engineering, Georgia Institute of Technology. He is currently with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA. e-mail: (ahmad.beirami@duke.edu).

L. Huang is with the School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China e-mail: sunny_hll@sjtu.edu.cn.

M. Sardari and F. Fekri are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA. e-mail: ({mohsen.sardai, fekri}@ece.gatech.edu).

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1017234.

This paper was presented in part at the 2013 IEEE International Conference on Computer Communications (INFOCOM 2013) [1], and the 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton 2013) [2], and the 15th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2014) [3].

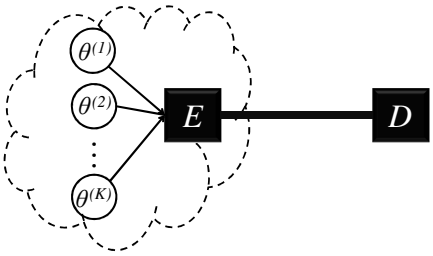

Fig. 1. The basic scenario of universal compression with side information for a mixture source.

As shown in Fig. 1, we assume that each sequence (e.g., network packet) is a sample of length n from a mixture of K parametric sources. We consider the scenario where T sequences from the mixture source are shared as side information between the encoder E and the decoder D and the first objective is to derive the average minimax redundancy incurred in the *optimal* universal compression with side information as a function of n, K, and T. We further develop a clustering algorithm for the universal compression with side information based on the Hellinger distance of the sequences and show its effectiveness on real network traffic traces. We prove that the adopted clustering algorithm is consistent and asymptotically *optimal* using the side information in the sense that, given the side information, the minimum codeword length in the universal compression of a new sequence from the mixture source using side information is attained.

Our contributions in this paper can be summarized as follows:

- We formally characterize the average minimax redundancy incurred in universal compression of a random sequence of length n from a mixture source given that the encoder and the decoder have access to a shared side information of T sequences (each of length n from the mixture of K parametric sources).
- We demonstrate that the performance of the optimal universal compression with side information (in the minimax sense) is almost surely that of the universal compression with perfect clustering of the memory based on the originator source in the mixture when source labels are available. Hence, clustering is optimal in the first-order redundancy term for universal compression with side information.
- We propose a clustering strategy for the side information that aims at grouping the side information sequences that share similar statistical properties. A newly generated packet by the mixture source is classified into one of the clusters for compression. We demonstrate the effectiveness of the proposed algorithm through experiments performed on real network traffic traces.

The rest of this paper is organized as follows. In Section II, we review the necessary background on universal compression. In Section III, we present the formal definition of the problem. In Section IV, we derive the entropy of the mixture source, which serves as a lower limit on the average codeword length. In Section V, we provide the main results on the universal compression of mixture sources with and without side information and discuss their implications. In Section VII, we present the clustering algorithms used for the compression of the mixture sources. In Section VIII, we provide simulation results that support our theoretical results on the compression of the mixture sources. Finally, Section IX concludes this paper.

II. Background on Universal Source Coding

In this section, we briefly review the necessary background on the universal compression of parametric sources. We defer the generalization to a mixture source to Section III. Let a parametric source be defined using a d-dimensional parameter vector $\theta = (\theta_1, ..., \theta_d) \in \Lambda_d$ that is a priori unknown, where d denotes the number of the source parameters and $\Lambda_d \subset \mathbb{R}^d$ is the space of d-dimensional parameter vectors of interest. Denote μ_θ as the parametric source (i.e., the probability measure defined by the parameter vector θ on sequences of length n).

Let $\mathcal{X}$ denote a finite alphabet. Let X^n denote a sample (random vector of length n) from the probability measure μ_θ. We further denote $x^n = (x_1, ..., x_n) \in \mathcal{X}^n$ as a realization of the random vector X^n. Then, define $H_n(\theta) \triangleq H(X^n|\theta)$ as the source entropy given the parameter vector θ, i.e.,

$$H_n(\theta) = \mathbf{E}\log\left(\frac{1}{\mu_\theta(X^n)}\right) = \sum_{x^n \in \mathcal{X}^n} \mu_\theta(x^n) \log\left(\frac{1}{\mu_\theta(x^n)}\right). \tag{1}$$

Throughout this paper $\log(\cdot)$ always denotes the logarithm in base 2 and expectations are taken over the random sequence X^n with respect to the probability measure μ_θ.

In this paper, we focus on the class of strictly lossless uniquely decodable fixed-to-variable codes defined as the following. The code $c_n : \mathcal{X}^n \to \{0,1\}^*$ is called strictly lossless (also called zero-error) on sequences of length n if there exists a reverse mapping $d_n : \{0,1\}^* \to \mathcal{X}^n$ such that $\forall x^n \in \mathcal{X}^n$, we have $d_n(c_n(x^n)) = x^n$. Further, let $l : \mathcal{X}^n \to \mathbb{R}$ denote the universal strictly lossless length function for the codeword $c_n(x^n)$ associated with the sequence x^n such that $l(\cdot)$ satisfies Kraft's inequality to ensure unique decodability. That is $\sum_{x^n \in \mathcal{X}^n} 2^{-l(x^n)} \leq 1$. In this paper, we ignore the integer constraint on the length function, which results in a negligible redundancy upper bounded by 1 bit analyzed exactly in [28], [29].

Denote $R_{n,d}(l, \theta)$ as the average (expected) redundancy of the code c_n with length function l on a sequence of length n for the parameter vector θ, defined as

$$R_{n,d}(l, \theta) = \mathbf{E}l(X^n) - H_n(\theta). \tag{2}$$

Note that the average redundancy is non-negative. Further, a (universal) code is called weakly optimal if its average codeword length normalized to the sequence length uniformly converges to the source entropy rate, i.e., $\lim_{n\to\infty} \frac{1}{n} R_{n,d}(l, \theta) = 0$ for all $\theta \in \Lambda_d$.

Define $\underline{R}_{n,d}$ as the average maximin redundancy, i.e.,

$$\underline{R}_{n,d} = \max_{p(\cdot)} \min_{l} \int_{\Lambda_d} R_{n,d}(l, \theta) p(\theta) d\theta. \tag{3}$$

The average maximin redundancy is associated with the best code under the worst prior on the space of parameter vectors

(i.e., the capacity achieving Jeffreys' prior). Let $\bar{R}_{n,d}$ denote the average minimax redundancy, which is defined as

$$\bar{R}_{n,d} = \min_{l} \max_{\theta} R_{n,d}(l, \theta). \tag{4}$$

Gallager showed that the average minimax redundancy and the average maximin redundancy (as defined above) are equal [18]. Let $\mathcal{I}(\theta)$ be the Fisher information matrix associated with the parameter vector θ, i.e.,

$$\mathcal{I}(\theta) \triangleq \lim_{n\to\infty} \frac{1}{n \log e} \mathbf{E}\left\{ \frac{\partial^2}{\partial\theta_i \partial\theta_j} \log\left(\frac{1}{\mu_\theta(X^n)} \right) \right\}. \tag{5}$$

Fisher information matrix quantifies the amount of information, on average, that each symbol in a sample sequence x^n from the source conveys about the source parameter vector. Let Jeffreys' prior on the parameter vector θ be denoted by

$$p_J(\theta) \triangleq \frac{|\mathcal{I}(\theta)|^{\frac{1}{2}}}{\int_{\Lambda_d} |\mathcal{I}(\lambda)|^{\frac{1}{2}} d\lambda}. \tag{6}$$

Jeffreys' prior is optimal in the sense that the average minimax redundancy is asymptotically achieved (up to a constant) when the parameter vector θ is assumed to follow Jeffreys' prior [18], [30], [31].[1] Jeffreys' prior is particularly interesting because it is also maximin optimal, which corresponds to the worst-case prior for the best compression scheme (called the capacity achieving prior) [18].

We need some regularity conditions to hold for the parametric model so that our results can be derived.

1) The parametric model is smooth, i.e., twice differentiable with respect to θ in the interior of Λ so that the Fisher information matrix can be defined. Further, the limit in (5) exists.
2) The determinant of fisher information matrix is finite for all θ in the interior of Λ and the normalization constant in the denominator of (6) is finite.
3) The parametric model has a minimal d-dimensional representation, i.e., $\mathcal{I}(\theta)$ is full-rank. Hence, $\mathcal{I}^{-1}(\theta)$ exists.
4) We require that the central limit theorem holds for the maximum likelihood estimator $\hat{\theta}(x^n)$ of each θ in the interior of Λ so that $(\hat{\theta}(X^n) - \theta)\sqrt{n}$ converges to a normal distribution with zero mean and covariance matrix $\mathcal{I}^{-1}(\theta)$.

The average minimax (maximin) redundancy is well studied for a single parametric source given by the following theorem.

Theorem 1 (**[30], [31]).** *The average minimax (maximin) redundancy is given by*

$$\bar{R}_{n,d} = \frac{d}{2} \log\left(\frac{n}{2\pi e} \right) + \log \int_{\Lambda_d} |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta + o(1).^2 \tag{7}$$

[1]The boundary risk is asymptotically strictly larger than the interior risk by a constant using Jeffreys' prior and when the space of the parameter vectors includes the boundary, a modification of Jeffreys' prior towards the boundary to compensate for this is minimax optimal (cf. [32]).

[2]Throughout this work, we have used the following asymptotic notations:

- $f(n) = o(g(n))$ iff $|f(n)| \leq |g(n)|\epsilon,\ \forall\epsilon$,
- $f(n) = O(g(n))$ iff $|f(n)| \leq |g(n)|k,\ \exists k$,
- $f(n) = \omega(g(n))$ iff $g(n) = o(f(n))$,
- $f(n) = \Omega(g(n))$ iff $g(n) = O(f(n))$,
- $f(n) \sim g(n)$ iff $\lim_{n\to\infty} f(n)/g(n) = 1$,
- $f(n) \lesssim g(n)$ iff $f(n) = o(g(n))$, and
- $f(n) \gtrsim g(n)$ iff $f(n) = \omega(g(n))$.

According to Theorem 1, the average maximin redundancy scales as $\frac{d}{2}\log n + O(1)$. This redundancy may indeed be a significant overhead on top of the entropy for small sequences, as the second term in (7) could be relatively large for small n as characterized in [13].

III. Problem Setup

In this section, we present the setup of the universal compression with common side information at the encoder and the decoder. Let $\Delta \triangleq \left\{\theta^{(i)}\right\}_{i=1}^{K}$ denote the set of $K \triangleq |\Delta|$ parameter vectors of interest where $\theta^{(i)} \in \Lambda_{d_i}$ is a d_i-dimensional parameter vector. Note that we let K deterministically scale with n. Let $d_{\max} \triangleq \max\{d_1, \ldots, d_K\}$ denote the maximum dimension of the parameter vectors, where we assume that $d_{\max} = O(1)$, i.e., $d_{\max}$ is finite. We further assume that for any $d < d'$, we have $\Lambda_d \subset \Lambda_{d'}$, and hence, Δ consists of K points on the space $\Lambda_{d_{\max}}$. In this setup, as in Fig. 1, the source is a mixture of K parametric sources $\mu_{\theta^{(1)}}, \ldots, \mu_{\theta^{(K)}}$, where for all $i \in [K] \triangleq \{1, \ldots, K\}$, $\theta^{(i)}$ is a d_i-dimensional unknown parameter vector. For the generation of each sequence of length n, the generator source is selected according to the probability vector $\mathbf{w} = (w_1, \ldots, w_K)$ from the mixture, i.e., Δ. In other words, $p(\theta|\Delta) = \sum_{i=1}^{K} w_i \delta(\theta - \theta^{(i)})$, where w_i is the probability that the sequence is generated by source $\theta^{(i)}$ in the mixture. The random set Δ (which is unknown a priori) is generated once and is used thereafter for the generation of all sequences from the mixture source. Let S be a random variable that determines the source index from which of the sequence is generated. As such, S follows the distribution $\mathbf{w}$ over $[K]$, i.e., $\mathbf{P}[S = i] = w_i$. Then, by definition, we have $\theta = \theta^{(S)}$ given Δ. Unlike Δ that is generated once, S is chosen via $\mathbf{w}$ every time a new sequence is generated. Let the mixture entropy $H(\mathbf{w})$ be defined as $H(\mathbf{w}) = -\sum_{i\in[K]} w_i \log w_i$.[3]

We assume that, in Fig. 1, both the encoder E and the decoder D have access to a common side information of T previous sequences (indexed by $[T]$) from the mixture of K parametric sources, where each of these sequences is independently generated according to the above procedure. Let $m \triangleq nT$ denote the aggregate length of the previous T sequences from the mixture source.[4] Further, denote $\mathbf{y}^{n,T} = \{y^n(t)\}_{t=1}^{T}$ as the set of the previous T sequences shared between E and D, where $y^n(t)$ is a sequence of length n generated from the source $\theta^{S(t)}$ at time epoch t, where $S(t)$ follows $\mathbf{w}$ on $[K]$. In other words, $y^n(t) \sim \mu_{\theta^{(S(t))}}$. Further, denote $\mathbf{S}$ as the vector $\mathbf{S} = (S(1), ..., S(T))$, which contains the indices of the sources that generated the T previous side information sequences.

[3]We define entropy $H(\mathbf{r})$ for any vector $\mathbf{r}$ such that $\sum_i r_i = 1$ in the same manner throughout the paper.

[4]For simplicity of the discussion, we consider the lengths of all sequences to be equal to n. However, most of the results are readily extendible to the case where the sequences are not necessarily equal in length.

Let $l_{\mathrm{M}}(x^n, \mathbf{y}^{n,T})$ denote a length function that utilizes the side information $\mathbf{y}^{n,T}$ in the compression of a new sequence x^n. The objective is to analyze the average redundancy in the compression of a new sequence x^n that is independently generated by the same mixture source with source index Z (which also follows $\mathbf{w}$). We investigate the fundamental limits of the universal compression with side information ($\mathbf{y}^{n,T}$) that is shared between the encoder and the decoder and compare with that of the universal compression without side information of the previous sequences. In this respect, it is straightforward to show that the minimax and maximin average redundancy are equivalent and are given by the capacity of the channel between the sequence X^n and the parameter vectors Δ given side information sequence $Y^{n,T}$. Hence, $I(X^n; \Delta|\mathbf{Y}^{n,T})$ and $I(X^n; \Delta)$, for different values of the sequence length n, memory (side information) size $m = nT$, the weight of the mixture $\mathbf{w}$, and the dimensions of the parameter vectors $\mathbf{d}$, serve as two of the main fundamental limits of the universal compression in this setup.

IV. Entropy of the Mixture Source: Compression with Known Source Parameter Vectors

In this section, we derive the limits of compression when the source parameter vectors are known. It is well known that for the mixture source, optimal compression is achieved by mixing the models. In other words, let $p(x^n)$ denote the mixture probability distribution on sequences of length n, which is defined as

$$p(x^n) = \sum_{i=1}^{K} w_i \mu_{\theta^{(i)}}(x^n). \tag{8}$$

Hence, the length function

$$l(x^n) = \log\left(\frac{1}{p(x^n)}\right) \tag{9}$$

is the optimal length function in this case, and it will achieve the entropy of the mixture source.

To derive the limits of compression for known source parameter vectors, we need to derive the entropy of the mixture source. Let $H_n(\Delta, Z) \triangleq H(X^n|\Delta, Z)$ be defined as the entropy of a random sequence X^n from the mixture source given that the source parameters are known to be the set Δ and the index of the source that has generated the sequence (i.e., Z) is also known.[5] Then, in this case, by definition

$$H_n(\Delta, Z) = \sum_{i=1}^{K} w_i H_n(\theta^{(i)}), \tag{10}$$

where $H_n(\theta^{(i)})$ is the entropy of source $\mu_{\theta^{(i)}}$ given $\theta^{(i)}$ defined in (1). Note that $H_n(\Delta, Z)$ is *not* the achievable performance of the compression. It is merely introduced here so as to make the presentation of the results more convenient.

Let the set Δ be partitioned into subsets in the following fashion.

$$\Delta = \cup_{d=1}^{d_{\max}} \Delta_d, \tag{11}$$

[5]We assume that the random set of parameter vectors is generated once and used for the generation of all sequences of length n thereafter. Therefore, throughout the paper, whenever we assume that Δ is given, we mean that the set of the parameter vectors is known to be the set Δ.

where Δ_d is the set of the d-dimensional parameter vectors in Δ. Further, let $K_d \triangleq |\Delta_d|$ be the number of parameter vectors in set Δ_d. In other words, K_d is the number of sources of dimension d in the mixture source. Hence, $\sum_{d=1}^{d_{\max}} K_d = K$. Now, we can relabel the elements in Δ according to their parameter vectors. Let $\Delta_d = \{\theta^{(d,1)}, \ldots, \theta^{(d,K_d)}\}$. Denote $\mathbf{w}_d = (w_{d,1}, \ldots, w_{d,K_d})$ as the weight of the d-dimensional parameter vectors. Further, let $v_d \triangleq \sum_{i=1}^{K_d} w_{d,i}$ be the aggregate weight of all d-dimensional parameter vectors and denote $\mathbf{v} \triangleq (v_1, \ldots, v_{d_{\max}})$. Let $\hat{\mathbf{w}}_d \triangleq \mathbf{w}_d / v_d$, i.e., we have $\hat{w}_{d,i} \triangleq w_{d,i}/v_d$, for $1 \leq i \leq K_d$.

Hence, $H_n(\Delta, Z)$ can be rewritten as

$$H_n(\Delta, Z) = \sum_{d=1}^{d_{\max}} \sum_{i=1}^{K_d} w_{d,i} H_n(\theta^{(d,i)}) \tag{12}$$

$$= \sum_{d=1}^{d_{\max}} v_d \sum_{i=1}^{K_d} \hat{w}_{d,i} H_n(\theta^{(d,i)}). \tag{13}$$

Next, we derive the entropy of the mixture source (which sets the asymptotic fundamental lower limit on the codeword length for the known source parameters case), i.e., when Δ is known. Define $H_n(\Delta) \triangleq H(X^n|\Delta)$.

Theorem 2. *The entropy of the mixture source for all Δ except for a set $A(n)$ whose volume asymptotically vanishes as $n \to \infty$, is given by*

$$H_n(\Delta) = H_n(\Delta, Z) + H(\mathbf{v}) + \sum_{d=1}^{d_{max}} v_d H_d + o(1), \tag{14}$$

where H_d is given by

$$H_d = \begin{cases} H(\hat{\mathbf{w}}_d) & \text{if } H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n \\ \bar{R}_{n,d} & \text{if } H(\hat{\mathbf{w}}_d) \gtrsim \frac{d}{2}\log n \end{cases}, \tag{15}$$

and $\bar{R}_{n,d}$ is given by (7)*.*

Proof: The proof is explained in the appendix. ∎

Theorem 2 determines the entropy of the mixture source, which corresponds to the minimum codeword length when the parameter vectors in the set Δ are known to the encoder and the decoder (i.e., non-universal compression). Note that $H_n(\Delta)$ also serves as a trivial lower bound on the codeword length for the case of universal compression (i.e., unknown parameter vectors). For sufficiently low-entropy $\hat{\mathbf{w}}_d$ (or roughly sufficiently small K_d), the price of describing the d-dimensional parameter vectors is, on average, equal to $H(\hat{\mathbf{w}}_d)$, which corresponds to describing the respective source parameter vector in the encoder.

Remark. Theorem 2 does not hold for an asymptotically vanishing volume of the parameter vectors. This is because one can choose the parameter vectors in a way that they do not conform to asymptotic scaling. For example, if all the parameter vectors are chosen to be equal, then the extra redundancy term over $H_n(\Delta, Z) = H_n(\theta^{(1)})$ would be zero. On the other hand, the result states that the volume of the space covered by such choices would become vanishingly small as $n \to \infty$. This is equivalent to saying if the parameter vectors are chosen independently according to a uniform prior on the

state of parameter vectors, then the probability of the event that they do not conform to the scaling predicted by Theorem 2 is vanishingly small.

The following corollary describes the entropy when the number of source parameter vectors are sufficiently small.

Corollary 3. *If $K = O\left(n^{\frac{1}{2}-\epsilon}\right)$ for some $\epsilon > 0$, then for all Δ except for a set $A(n)$ whose volume asymptotically vanishes as $n \to \infty$, we have*

$$H_n(\Delta) = H_n(\Delta, Z) + H(\mathbf{w}) + o(1). \tag{16}$$

Proof: Since $K = O\left(n^{\frac{1}{2}-\epsilon}\right)$ for some $\epsilon > 0$, we have $K_d = O\left(n^{\frac{d}{2}-\epsilon}\right)$ for some $\epsilon > 0$ and for all $1 \leq d \leq d_{\max}$. Thus, we have $H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n$. Thus, $H_d = H(\hat{\mathbf{w}}_d)$ for all $1 \leq d \leq d_{\max}$. The proof is completed by noting that

$$H(\mathbf{w}) = H(\mathbf{v}) + \sum_{d=1}^{d_{\max}} v_d H(\hat{\mathbf{w}}_d). \tag{17}$$

■

According to the corollary, when $K = O\left(n^{\frac{1}{2}-\epsilon}\right)$ for some $\epsilon > 0$, the optimal coding strategy (when the source parameters are known) for asymptotically almost all the parameter vectors would be to encode the source index Z and then use the optimal code (e.g., Huffman code) associated with parameter $\theta^{(Z)}$ for sequences of length n to encode the sequence x^n. In fact, if $H(\mathbf{w}) \lesssim \frac{d}{2}\log n$, then the cost of encoding the parameter is asymptotically smaller than the cost of universally encoding the parameter and hence it is beneficial to encode the parameter vector using an average of $H(\mathbf{w})$ bits. Further, if $K = 1$, then $\Delta = \theta^{(1)}$ and $Z = 1$ would be deterministic. Hence, $H_n(\Delta) = H_n(\Delta, Z) = H_n(\theta^{(1)})$, which was introduced in (1) as the average compression limit for the case of a single known source parameter vector.

Corollary 4. *If $H(\hat{\mathbf{w}}_d) \gtrsim \frac{d}{2}\log n$ for all $1 \leq d \leq d_{max}$ such that $v_d > 0$, then for all Δ except for a set $A(n)$ whose volume asymptotically vanishes as $n \to \infty$, we have*

$$H_n(\Delta) = H_n(\Delta, Z) + H(\mathbf{v}) + \sum_{d=1}^{d_{max}} v_d \bar{R}_{n,d} + o(1). \tag{18}$$

Proof: The proof is very similar to the previous corollary and is omitted for brevity. ■

According to the corollary, in the case where the number of sources in the mixture is very large, the mixture entropy converges to $H_n(\Delta, Z)$ plus $H(\mathbf{v})$ plus the weighted average of the $\bar{R}_{n,d}$ terms (which are exactly the average maximin redundancy in the *universal* compression of parametric sources with d *unknown* parameters given in Theorem 1). At the first glance, it may seem odd that the codeword length in the case of *known* source parameter vectors incurs a term that is associated with the universal compression of a source with an *unknown* parameter vector. A closer look, however, reveals that in this case the cost of encoding the source index of a d-dimensional parameter vector surpasses the cost of universally encoding the source parameter vector. Hence, intuitively, it no longer makes sense to encode the d-dimensional parameter vector for the compression of the sequence x^n using an average of $H(\hat{\mathbf{w}}_d)$ bits. More rigorously speaking, as was shown in the proof of Theorem 2, the probability distribution of x^n given $\theta \in \Delta_d$ would converge to the probability distribution of x^n when the source has one *unknown* d-dimensional parameter vector that follows Jeffreys' prior. This in turn results in the $\bar{R}_{n,d}$ term in the compression performance.

V. Fundamental Limits of Universal Compression for Mixture Sources

In the previous section, we derived the limits of the compression of mixture sources when the source parameter vectors are known. In this section, we will turn to the universal compression problem and will quantify the benefits of side information. To see the impact of the universality and side information on the compression performance, i.e., to investigate the impact of Δ being unknown, we will need to analyze and compare the average minimax redundancy (the excess codeword length on top of the entropy) for the following important *fundamental* schemes.

- Ucomp: Universal compression, which is the conventional compression based solution. This is the usual universal compression in the literature with length function $l(x^n)$.
- UcompM: Universal compression with side information (common memory between the encoder and the decoder), which takes in the side information sequence into the encoding and decoding with length function $l_{\text{M}}(x^n, \mathbf{y}^{n,T})$.
- UcompMS: Universal compression with side information and source indices, which uses the side information sequences and also the indices of the sources that generated them (at the encoder/decoder). The respective length function will be denoted by $l_{\text{MS}}(x^n, \mathbf{y}^{n,T}, \mathbf{S}, Z)$.[6]

We quantify the performance of these fundamental schemes using their respective average redundancies. Let $R(l, \Delta)$ denote the average redundancy of the Ucomp compression algorithm for the universal compression of a mixture source, which is defined in the usual way, as in (2), given by

$$R(l, \Delta) = \mathbf{E}l(X^n) - H_n(\Delta). \tag{19}$$

Further, let $\underline{R}(n, \mathbf{w}, \mathbf{d})$ and $\bar{R}(n, \mathbf{w}, \mathbf{d})$ denote the average maximin and minimax redundancy, respectively, which are defined in the same manner as in (3) and (4) in Section II. Our goal is to characterize the performance of universal compression as a function of the mixture weights $\mathbf{w}$ and source parameter vector dimensions $\mathbf{d}$. Note that the average maximin redundancies $\underline{R}_{\text{M}}(n, m, \mathbf{w}, \mathbf{d})$ and $\underline{R}_{\text{MS}}(n, m, \mathbf{w}, \mathbf{d})$, and the average minimax redundancies $\bar{R}_{\text{M}}(n, m, \mathbf{w}, \mathbf{d})$ and $\bar{R}_{\text{MS}}(n, m, \mathbf{w}, \mathbf{d})$ can also be defined similarly.

It is straightforward to extend Gallager's Theorem to the following.

Theorem 5. *Consider Ucomp, UcompM, and UcompMS for the compression of mixture sources with the set of parameter vectors $\Delta \in \Lambda'(n)$, where $\Lambda'(n)$ is defined in* (40)*. Then,*

[6]UcompMS scheme may be uninteresting from practical point of view as the source indices may be unknown in a lot of applications.

the average minimax redundancy and the average maximin redundancy are equivalent, i.e.,

$$\begin{aligned}\bar{R}(n,\mathbf{w},\mathbf{d}) &= \underline{R}(n,\mathbf{w},\mathbf{d}) \\ &= \max_{p} I(X^n;\Delta). \qquad (20)\\ \bar{R}_{\text{M}}(n,m,\mathbf{w},\mathbf{d}) &= \underline{R}_{\text{M}}(n,m,\mathbf{w},\mathbf{d}) \\ &= \max_{p} I(X^n;\Delta|\mathbf{Y}^{n,T}). \qquad (21)\\ \bar{R}_{\text{MS}}(n,m,\mathbf{w},\mathbf{d}) &= \underline{R}_{\text{MS}}(n,m,\mathbf{w},\mathbf{d}) \\ &= \max_{p} I(X^n;\Delta|\mathbf{Y}^{n,T},\mathbf{S},Z) \qquad (22)\end{aligned}$$

Further, if Δ is chosen such that for $i \neq j$, we have $\theta^{(i)}$ and $\theta^{(j)}$ are independent and the marginal distribution of each $\theta^{(i)}$ is Jeffreys' prior on the d_i-dimensional space Λ_{d_i}, such prior is asymptotically capacity achieving as $n \to \infty$.

Proof: The proof is explained in the appendix. ∎

Remark. Note that our results hold for a set $\Lambda'(n) = \Lambda \setminus A(n)$ whose volume asymptotically equals that of Λ. In other words, if you pick the parameter vectors according to any distribution whose support is the entire set Λ (i.e., it puts non-zero mass over any point in Λ), then our results would hold asymptotically almost surely (a.s.).[7]

Next, we state a trivial ordering on the average minimax redundancy of these *fundamental* schemes.

Proposition 6. *The following ordering holds for the average minimax redundancies of Ucomp, UcompM, and UcompMS.*

$$\bar{R}_{\text{MS}}(n,m,\mathbf{w},\mathbf{d}) \leq \bar{R}_{\text{M}}(n,m,\mathbf{w},\mathbf{d}) \leq \bar{R}(n,\mathbf{w},\mathbf{d}). \qquad (23)$$

Proof: This holds as the UcompMS length function can choose to ignore $\mathbf{S}$ and Z, and also the UcompM length function can choose to ignore $\mathbf{y}^{n,T}$. In other words, more information cannot hurt. ∎

In the rest of this section, our goal is to characterize the average minimax redundancies of the aforementioned fundamental schemes, and in particular the gaps between them, to understand the *fundamental* benefits provided by side information in the universal compression of a mixture of parametric sources.

A. Ucomp: Universal Compression without Side Information

We refer to Ucomp as the universal compression without side information, in which a universal length function $l(x^n)$ is used to compress the sequence x^n without regard to the side information sequence $\mathbf{y}^{n,T}$.

Next, we state the main result in characterizing the average minimax redundancy.

Theorem 7. *In the case of Ucomp, we have*

$$\bar{R}(n,\mathbf{w},\mathbf{d}) = \sum_{d=1}^{d_{max}} v_d(\bar{R}_{n,d} - H_d) + o(1)\ a.s., \qquad (24)$$

where H_d is defined in (15)*.*

Proof: The proof is explained in the appendix. ∎

[7]An event A_n happens asymptotically almost surely (a.s.) if and only if $\lim_{n\to\infty} \mathbb{P}[A_n] = 1$.

According to Theorem 7, in the universal compression of a sequence of length n from the mixture source, the main term of the redundancy scales as the weighted average of $(\bar{R}_{n,d} - H_d)$ terms. This can be significantly large if $H(\mathbf{w}_d)$ is much smaller than $\frac{d}{2}\log n$. Again, if $K = 1$, we have $\bar{R}(n,1,d) = \bar{R}_{n,d}$; this is exactly the average minimax (maximin) redundancy in the case of one unknown d-dimensional source parameter vector described in Theorem 1.

Theorem 7 also suggests that independently from K and $H(\mathbf{w})$, the price to be paid for universality is given by $\bar{R}_{n,d}$ over and above $H_n(\Delta, Z)$, i.e., the entropy when Δ and Z are known. In other words, $H(X^n) - H_n(\Delta, Z)$ scales like $\sum_d v_d \bar{R}_{n,d}$ (which is the price of universal compression of a sequence of length n from a single source with an unknown d-dimensional parameter vector that follows the worst-case Jeffreys' prior averaged over d).

Corollary 8. *If $H(\hat{\mathbf{w}}_d) \gtrsim \frac{d}{2}\log n$ for all $1 \leq d \leq d_{max}$, then*

$$\bar{R}(n,\mathbf{w},\mathbf{d}) = o(1)\ a.s. \qquad (25)$$

Proof: If $v_d > 0$, then $H(\hat{\mathbf{w}}_d) \gtrsim \frac{d}{2}\log n$, and hence, we have $H_d = \bar{R}_{n,d}$, which means $\bar{R}_{n,d} - H_d$ vanishes. Hence, the main redundancy term $v_d(\bar{R}_{n,d} - H_d)$ in Theorem 7 vanishes for all $1 \leq d \leq d_{\max}$, which completes the proof. ∎

According to the corollary, for large K, we asymptotically almost surely (a.s.) expect no extra redundancy associated with universality on top of the mixture entropy. This is not surprising as even in the case of *known* source parameter vectors, as given by Theorem 2, the redundancy converges to the weighted average of the redundancies for a d-dimensional *unknown* source parameter vector that follow Jeffreys' prior. Therefore, there is no extra penalty when the source parameter vectors are indeed unknown.

B. UcompM: Universal Compression with Side Information

We refer to UcompM as the universal compression with side information. In this section, our goal is to characterize the average minimax redundancy of the UcompM scheme given the side information, i.e., $\bar{R}_{\text{M}}(n,m,\mathbf{w},\mathbf{d})$, where $T = \frac{m}{n}$ sequences from the mixture source are shared between the encoder and the decoder as side information.

Proposition 9. *In the case of UcompM, if $m = O(1)$, then*

$$\bar{R}_{\text{M}}(n,m,\mathbf{w},\mathbf{d}) = \bar{R}(n,\mathbf{w},\mathbf{d}) - O(1). \qquad (26)$$

According to Proposition 9, when m does not grow to infinity, the improvement offered by side information is at most constant, which is negligible compared with the leading term of redundancy which is $O(\log n)$.

Theorem 10. *In the case of UcompM, for $m = \omega(1)$ we have*

$$\bar{R}_{\text{M}}(n,m,\mathbf{w},\mathbf{d}) = \sum_{d=1}^{d_{max}} v_d \sum_{i=1}^{K} \hat{w}_{d,i}\hat{R}_{d,i} + o(1)\ a.s., \qquad (27)$$

where $\hat{R}_{d,i}$ is given by

$$\hat{R}_{d,i} = \begin{cases} \frac{d}{2}\log\left(1 + \frac{n}{\hat{w}_{d,i}m}\right) + \delta & \text{if } H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n \\ 0 & \text{if } H(\hat{\mathbf{w}}_d) \gtrsim \frac{d}{2}\log n \end{cases}, \qquad (28)$$

where δ is an absolute constant with respect to n and can be made arbitrarily small for sufficiently large T.

Proof: The proof is explained in the appendix. ∎

Theorem 10 characterizes the redundancy of the optimal universal compression scheme with side information, which uses a memory of size $m = nT$ (T sequences of size n) in the compression of a new sequence of length n. It is natural to expect that the side information will make the redundancy decrease. The redundancy of the UcompM decreases when $H(\mathbf{w})$ or roughly K is sufficiently small. Again, $K = 1$, results in $\bar{R}_{\text{M}}(n, m, 1, d) = \frac{d}{2}\log\left(1 + \frac{n}{m}\right) + o(1)$, which is consistent with what we derived for a single parametric source in [20]. Further, it is deduced from Theorem 10 that $\lim_{T\to\infty} \bar{R}_{\text{M}}(n, m, \mathbf{w}, \mathbf{d}) = o(1)$ (regardless of $\mathbf{w}$), i.e., the cost of universality would be negligible given that sufficiently large memory (side information) is available. Thus, the benefits of optimal universal compression with side information would be substantial when $H(\mathbf{w})$ is sufficiently small. On the other hand, when $H(\mathbf{w})$ grows very large, no benefit is obtained from the side information in the universal compression and the performance improvement becomes negligible. This is due to the fact that, in light of Theorem 7, the compression performance for the known source parameters case is already equal to that of the universal compression.

C. UcompMS

Next, we analyze the fundamental performance of a class of schemes that have access to the unknown source labels. In particular, we would like to analyze how much performance improvement the knowledge of the unknown source indices would offer over the fundamental limits of UcompM. We refer to UcompMS as the universal compression with perfectly clustered side information sequence $\mathbf{y}^{n,T}$, which is shared between the encoder E and the decoder D. Further, the index vector $\mathbf{S}$ of the memorized sequences and the index Z of the sequence x^n to be compressed are known to both E and D. Therefore, one can imagine that an oracle exists that can partition the sequences in $\mathbf{y}^{n,T}$ based on their source index. Then, it can be shown that it is optimal that E and D cluster the side information sequences according to $\mathbf{S}$ and use the minimax estimator to estimate the source parameter vector associated with each cluster; the encoder E classifies the sequence x^n to the respective cluster using the oracle and encodes the sequence only using the side information provided by the estimated parameter vector of the respective cluster.

Theorem 11. *In the case of UcompMS, we have*

$$\bar{R}_{\text{MS}}(n, m, \mathbf{w}, \mathbf{d}) = \sum_{d=1}^{d_{max}} v_d \sum_{i=1}^{K} \hat{w}_{d,i}\hat{R}_{d,i} + o(1)\ a.s., \quad (29)$$

where $\hat{R}_{d,i}$ is defined in (28)*.*

Proof: The proof is explained in the appendix. ∎

Theorem 11 characterizes the redundancy of the universal compression with perfectly clustered side information. It is straightforward to observe that for sufficiently large m, the redundancy of UcompMS becomes very small. However, UcompMS is impractical in most situations as the oracle that provides the source index is not available. As an important special case if $K = 1$, then $\bar{R}_{\text{MS}}(n, m, \mathbf{w}, \mathbf{d}) = \frac{d}{2}\log\left(1 + \frac{n}{m}\right) + o(1)$, which reduces to Theorem 2 of [20] regarding the average minimax redundancy for the case of a single source with an unknown parameter vector.

Corollary 12. *Regardless of* $\mathbf{w}$ *and* $\mathbf{d}$*, we have*

$$\lim_{T\to\infty} \bar{R}_{\text{MS}}(n, m, \mathbf{w}, \mathbf{d}) = o(1). \quad (30)$$

Proof: Note that $T \to \infty$ simply means $m \to \infty$, and $\frac{d}{2}\log\left(1 + \frac{n}{\hat{w}_{d,i}m}\right) \to 0$ as $m \to \infty$, completing the proof. ∎

According to the corollary, the redundancy vanishes as $T \to \infty$ (or equivalently $m \to \infty$). Therefore, for sufficiently large m, significant performance improvement is expected in terms of the number of bits required to describe a sequence x^n.

Corollary 13. *We have*

$$\bar{R}_{\text{M}}(n, m, \mathbf{w}, \mathbf{d}) = \bar{R}_{\text{MS}}(n, m, \mathbf{w}, \mathbf{d}) + o(1)\ a.s. \quad (31)$$

Proof: The corollary is proved by combining Theorems 10 and 11. ∎

Remark. The corollary has significant implications. It states that the performance of optimal universal compression with side information (UcompM), which uses a memory of size $m = nT$ (T sequences of size n) in the compression of a new sequence of length n is equal to that of the universal compression with perfectly clustered memory (UcompMS) up to $o(1)$ terms. Hence, when T is sufficiently large, we expect that both have the same performance. This indeed demonstrates that *clustering* is optimal for the universal compression with side information. As such, we pursue the clustering of the side information (i.e., memory) in this paper in Section VII.

VI. Operational Limits of Universal Compression for Mixture Sources

In addition to the fundamental schemes (and respective length functions), in this paper, we will also assess two *operational* schemes listed below. Both schemes fall in the UcompM coding regime that we have access to the memory but not the source indices.

- UcompM1: Simple universal compression with side information (common memory between the encoder and the decoder), which treats the side information as if it were generated from a single parametric source. In other words, it uses the minimax estimator for the unknown parameter vector of the source for a single source. The length function associated with this operational scheme is denoted by $l^1_{\text{M}}(x^n, \mathbf{y}^{n,T})$.
- UcompMc: Universal compression with clustering of the side information, which is the practical clustering-based scheme proposed in this paper and shall be described in Section VII.

Since for these operational schemes the length function is predetermined, we will quantify their performance under the worst-case prior on the space of the source parameter vectors. The worst-case prior is derived as a by-product of Theorem 5.

A. UcompM1: Simple Universal Compression with Side Information

Next, we comment on the performance of the simple universal compression with side information scheme that is regarded as UcompM1. In this compression scheme, it is assumed that the encoder E and the decoder D (in Fig. 1) both have access to the memorized sequence $\mathbf{y}^{n,T}$ from the mixture source. The sequence $\mathbf{y}^{n,T}$ is used to form the optimal minimax estimator of one unknown source parameter vector. Observe that the scheme would be minimax optimal if $\mathbf{y}^{n,T}$ was generated by a single parametric source with an unknown parameter vector. The estimated source parameter using the minimax estimator for one unknown parameter vector is then used for the compression of the sequence x^n.

As discussed in Section IV, when the source parameter vectors are known, then mixing the probability distributions is optimal and achieves the entropy. The subtlety here is that since the source parameter vectors are unknown, there is a penalty to be paid for learning them. When the source is a mixture of more than one source parameter vectors, UcompM1 will naively start to build a larger model for the source with much more parameters for it to be able to closely follow the source statistics. As the length of the sequences become sufficiently large, such an approach will be able to learn the source statistics fairly well. It will indeed converge to the source model as the depth of the built context tree grows but with significantly larger number of parameters. Unfortunately, model reduction methods such as context pruning [10] will not be a remedy to this issue either. We will comment more on the performance of UcompM1 in Section VIII.

B. UcompMc: Universal Compression with Clustering of the Side Information

Thus far, we argued why a naive memory-assisted compression (UcompM1) would suffer from curse of dimensionality in learning the unknown source parameters. On the other hand, in Section V, we theoretically proved that the optimal memory-assisted compression performs similarly to the memory-assisted compression with known source indices. This suggests that an asymptotically optimal strategy would be to cluster the side information sequences into several distinct models (one model for each source in the mixture). This shall significantly reduce the number of parameter models and hence will improve the compression performance. This is the subject of the next section of this paper.

VII. Clustering Algorithms for Compression of Mixture Sources

In this section, we present two clustering solutions for network packets. The k-means algorithm can be used for this purpose provided that a proper feature space and a relevant distance metric are selected. Further, we have also experimented with the non-parametric k-nearest neighbors clustering algorithm and we will comment on the performance of both algorithms. In the sequel, we describe a hierarchical clustering algorithm that proves to be useful for compression. The proposed hierarchy for the content-aware joint memorization and clustering for network packet compression is shown in Fig. 2. As shown, we first identify whether or not an incoming packet is compressible. If the packet is determined incompressible, it is neither compressed nor stored in the memory. On the other hand, the compressible packets are passed to the clustering unit which operates based on the Hellinger distance metric.

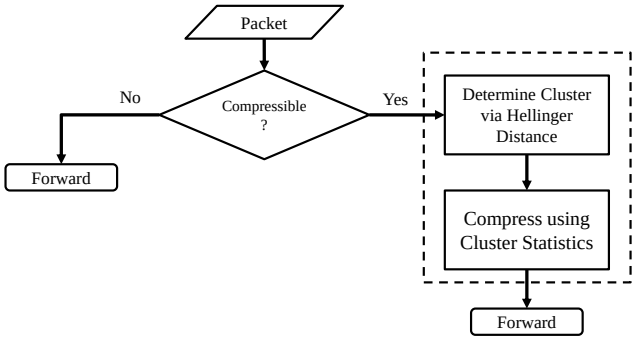

Fig. 2. Network packet compression flowchart. The modules in the dashed box are the components of the k-means clustering using Hellinger distance.

Compressibility determination: The compressibility determination is performed based on the empirical entropy of the data packet. The side information packets in memory may be divided into two categories: one category contains packets with very high entropy rate (close to 8 bits per byte) and hence these packets are incompressible. These include already compressed videos or images. The other category contains packets whose empirical entropy rate is estimated to be much less than 8, and hence, these packets are compressible. Therefore, as the first step, the packets are partitioned into compressible and incompressible. After the partitioning step, the packets in the resulting memory are all compressible. Then, we will perform a clustering of the resulting memory. Note that one may generate man-made models where each sequence has high entropy while the individual sequences are indeed highly correlated. On the other hand, our observations from the real data traces suggest that this issue is not encountered in practice. Hence, we chose to ignore the packets that we determine to be incompressible.

Feature selection: Feature extraction deals with extracting simpler descriptions for a large set of data that can accurately describe characteristics of original data. For memoryless source models, the frequency of each alphabet in the sequence defines an empirical probability density function (pdf) vector which also happens to be the sufficient statistics. Although for more sophisticated source models, the empirical pdf of the packet (i.e., the frequency of each byte in the packet) is not a sufficient statistics anymore as collisions may occur between different parametric sources in the marginal symbol distribution, the empirical probability distribution would still match for packets from the same parametric source while the probability of collision is relatively low. Further, since the lengths of the data packets are relatively small on the order of several kilobytes, any model beyond a memoryless model would overfit the data [11], [33]–[35]. Hence, we assume that each packet is generated using a memoryless model. We

choose the vector of the empirical pdf as our feature vector and since we work at the byte granularity (i.e., $|\mathcal{X}| = 256$), the feature vector is 255-dimensional (255 independent variables). We stress that the chosen feature space is not necessary optimal but simulations confirm that it works close to optimal in practice for packets of size 1,500 bytes or longer.

Distance metric: To perform clustering, we need to use a distance metric that determines the similarity between any two packets. Note that the overall objective is to reduce the compression rate where the compression penalty can be described in terms of KL-divergence between the true model and the estimated model (cf. [11]). On the other hand, KL-divergence is not a metric. Hence, the natural choice for the distance metric would be the Hellinger distance metric, which is widely used to quantify the similarity between two probability distributions (cf. [36]). For two probability distributions $p(\cdot)$ and $q(\cdot)$ defined on symbols from alphabet $\mathcal{X}$, the Hellinger distance is defined as

$$d_H(p,q) = \frac{1}{2}\sqrt{\sum_{x\in\mathcal{X}}\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2}. \qquad (32)$$

In our setup, we calculate the Hellinger distance of two packets using the empirical pdf of the symbols for each packet. Recall that a packet $x^n \in \mathcal{X}^n$ is a vector of n symbols $x_i \in \mathcal{A}$.

k-means clustering: As discussed earlier in Section III, we have a side information sequence of packets $\mathbf{y}^{n,T}$ that consists of T packets that originated from a mixture source model. We stress that the total number of source in the mixture (denoted by K) is unknown. Each packet in the memory needs to be assigned to one clusters from the k choices. We use the binary indicator c_t^j to denote the cluster assignment for the t-th packet $y^n(t)$. The indicator $c_t^j = 1$ if $y^n(t)$ is assigned to cluster $j \in [k]$, otherwise $c_t^j = 0$. Then, the objective function for clustering is given by

$$J = \sum_{t=1}^{T}\sum_{j=1}^{k} c_t^j d_H(q_t, u_j), \qquad (33)$$

where q_t is the distribution on the symbols obtained from $y^n(t)$ and u_j is the probability distribution vector on the symbols associated with the packets in cluster j. The goal of the clustering algorithm is to find the assignment c_t^j for $j \in [k]$ and $t \in [T]$ such that J is minimized.

The problem setup suggests that the k-means clustering algorithm [37] is suitable for our purpose. k-means algorithm is an iterative algorithm which consists of two steps for successive optimization of c_t^j (and hence u_j). Given cluster center u_j, the optimal c_t^j can be easily determined by assigning the packet $y^n(t)$ to the closest cluster with minimum Hellinger distance $d_H(q_t, u_j)$. Then, we fix c_t^j and update u_j. k-means clustering algorithm can successfully cluster data packets in the ideal situation with static number of source model mixture. However, this algorithm can break down when the number of sources cannot be estimated correctly, especially for the infinite mixture source model in real world networks. Note that k-means algorithm also requires the selection of k a priori. In our simulations we observed that choosing a large k would always do the job as most of the clusters will remain empty when the algorithm converges.

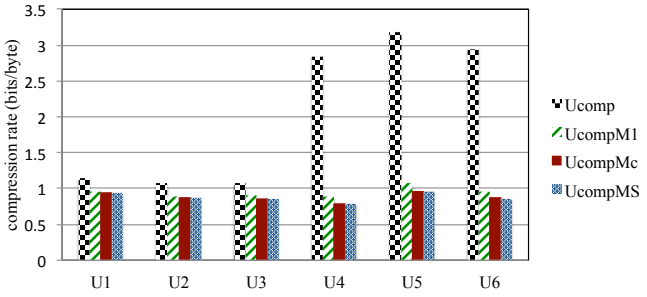

Fig. 3. Average compression-rate for a mixture of 3 memoryless and 3 first-order Markov sources using Lite PAQ compression algorithm.

Non-parametric k-nearest clustering: To cluster packets without assuming any parameters a priori about the data, we also used the dynamic non-parametric clustering method based on the well known k-nearest algorithm. To this end, we partition the memory into m small sub-clusters that are represented by the cluster centers $S = \{s_1, \ldots, s_m\}$. Each sub-cluster consists of about T/m neighboring packets with the minimum variance.

As soon as the fine-grain sub-clusters are produced, then we can process the training packets to form the appropriate memory for compression. After the initialization of the current sub-cluster set $C = S$, the sub-cluster from set C nearest to x^n is merged into the training set Q and is removed from C after merging. In other words, the new dynamic training set Q is updated. The merging ends when the expected number of training packets is reached. The actual number of sub-clusters is fixed according to the minimum number of packets requirement of compressor. Algorithm 1 elaborates the procedures of the non-parametric clustering for selection of training packets.

Algorithm 1 Non-Parametric k-Nearest Clustering Algorithm

```
Compute sub-cluster centers S = {s_1, ..., s_m}
for Incoming packet x^n do
    Compute distance d_H(x^n, s_i)
    Current sub-cluster set C = S
    while training_pkt_num<min_training_num do
        if s_{i*} = min_{s_i ∈ C} d_H(x^n, s_i) then
            Training set Q = Q ∪ {s_{i*}}
            Index set T = T ∪ {i*}
            training_pkt_num update
            Remove s_{i*} from C = {s_1, ..., s_m}
        end if
    end while
    Return Q and T
end for
```

In practice, the feature vectors of data packets are scattered in a high dimensional space and the shapes of clusters are arbitrary. In particular, when the sample data packet does not belong to any of the clusters, the performance of k-means clustering will be adversely impacted. On the other hand, by merging nearby sub-clusters, k-nearest algorithm can collect most useful training data with appropriate consistency for

sample packet compression. Besides, without the knowledge of the number of clusters in advance, the k-nearest clustering algorithm achieves performance improvement compared to k-means clustering. All the detailed simulation in next session will elaborate on the performance of the k-nearest algorithm for data compression.

Compression of a new packet: Once the clustering of memory is performed, we will derive the mixture distribution by mixing all the distributions obtained from the source models, as discussed in Section IV, to achieve the source model. Then, the new sequence can be compressed on the fly without any further processing. This is a perfect fit for the statistical compression methods, such as the CTW and LPAQ.

For dictionary based methods, since mixing is impossible, we perform classification to compress a new packet x^n. We first decide which cluster should be used as the side information to compress x^n. Therefore, we classify the packet x^n by assigning it to a proper cluster. The classification algorithm is as follows. Let c be the cluster label of x^n to be determined. We compute Hellinger distance between the symbol distribution q of x^n and the cluster u_j. Then x^n is assigned to the closest cluster by

$$c = \underset{1 \le j \le K}{\operatorname{argmin}} \, d_H(q, u_j). \tag{34}$$

VIII. Simulation and Evaluation

In this section, we present simulation results to demonstrate the performance of the proposed memory-assisted compression system with non-parametric clustering and the overall improvement obtained from side information in universal compression of a mixture of parametric sources. Furthermore, we discuss the trade-off between compression speed and performance.

A. Simulations on Man-Made Mixture Models

To validate the theoretical results of the paper, we chose to use a mixture of parametric sources as the content-generator for the traffic. In particular, we used a mixture of five memoryless and five first-order Markov sources on 256-ary alphabet ($|\mathcal{X}| = 256$). Consequently for a memoryless source the number of source parameters $d = 255$, while for a first-order Markov source d is 256×255 which is the number of independent transition probabilities. Further, we assume that each packet is selected uniformly at random from the above mentioned mixture. For short-length sequences, we generate 18,000 packets at random from this source model, where each packet is 1,500 bytes long. Then, we used 200 packets from each source as test packets for the purpose of evaluation.

Fig. 3 demonstrates the results of the simulation on man-made data generated from the described mixture source using Lite PAQ compression algorithm. This plot shows the compression rate measured in the number of bits required to describe each source byte. Hence, the uncompressed source would need 8 bits/byte. Sources U1 through U3 are memoryless whilst sources U4 through U6 are first-order Markov sources. As can be seen, when the source model is simpler universal compression (without side information can work relatively much better and get closer to the entropy) whereas when the source model is first-order Markov there is a 300% gap between the performance of the universal compression without side information and with side information. Further, as can be seen, the benefits of UcompMc over UcompM1 become more spelled out when the source model becomes more complex. We will see simulations on real data in the next section.

TABLE I
Simulation Setup Summary

Case	Value
No. of users in mixture source	15
No. of packets from each user in memory	1,800
Total no. of memory data packets	27,000
Average size of each data packet	1kB
Approximate size of total memory	25MB
No. of users for performance testing	15
Total number of test packets	200
Distance metric	Hellinger distance
Clustering algorithm	k-means, k-nearest
Compression algorithm	Gzip, CTW, Lite PAQ

B. Simulations on Real Network Traces

For a realistic evaluation, we perform simulation with data gathered from 20 different mobile users network traces in real world. The data set was gathered by Sanadhya *et al.* in [38]. First, we randomly generate packet sequences from the 27,000-packet mixture of 15 users to construct the commonly accessible memory for clustering. Then, 10 sample packets from each of the 20 users (200 packets in total) are selected as test packets. Note the test packets are distinct from the packets used for training. Besides, there are 50 test packets that are generated from the 5 users which are not used for the generation of the training packets and hence do not have packets in the mixture memory. Average compression rate of each test packet is taken as the compression performance metric. We stress that each test packet is compressed separately and the result is averaged over the sample test packets. This is due to the packets flow in networks is a combination of packets from different sources and can not be simply compressed together. The whole simulation setup is summarized in Table I.

To demonstrate the impact of the side information on the compression performance, we analyze the average compression rate of the three important schemes (Ucomp, UcompM1, and UcompMc) using gzip, CTW, and Lite PAQ in Figs. 4, 5, and 6, respectively. Please see [24] for a discussion on the pros and cons of using each of these compression algorithms for network data compression. As can be seen, universal compression without help of any memory packets (Ucomp) results in the largest (worst) compression-rate which verifies the penalty of finite-length compression analyzed in [13]. UcompMc, which is the cluster-based memory-assisted compression, consistently outperforms all other schemes. It is worth noting that for the data from users which are not necessarily from mixture source model (users T1,. . . , T5), non-parametric clustering still achieves impressive improvement compared to simple memory assisted compression UcompM1. Compression with memory of user's previous packets UcompMS sometimes performs

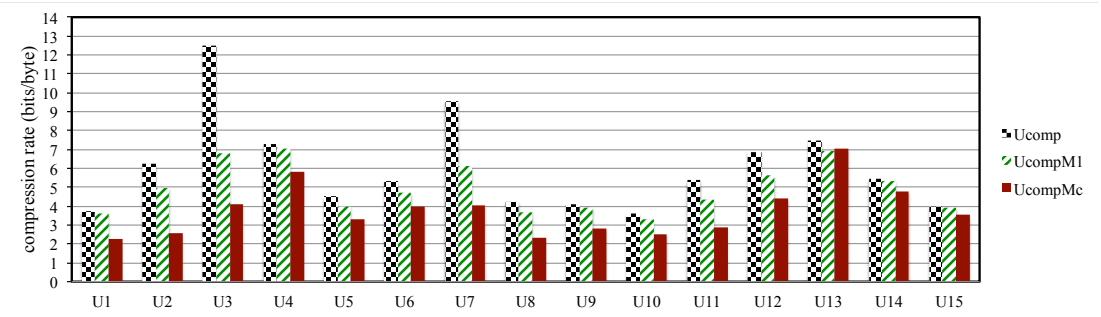

Fig. 4. Average compression-rate of GZIP on real traffic data.

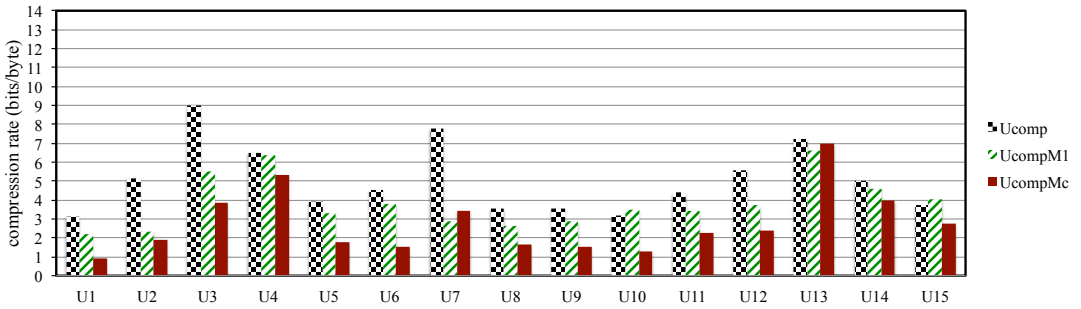

Fig. 5. Average compression-rate of CTW on real traffic data.

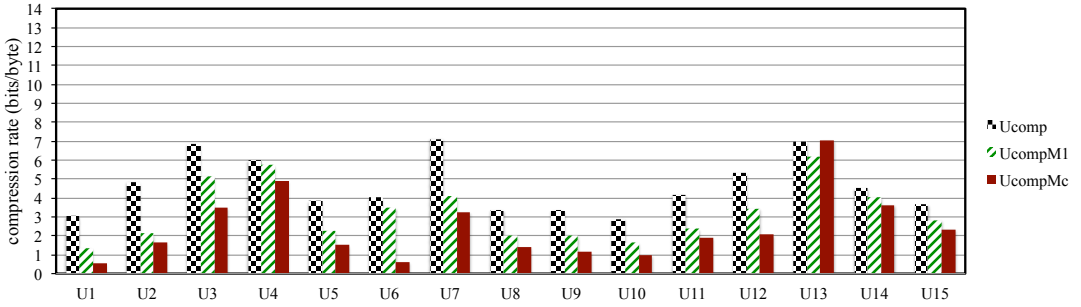

Fig. 6. Average compression-rate of Lite PAQ on real traffic data.

well while it sometimes performs poorly due to the fact that the user data possibly comes from variant source models. In general, clustering algorithm is applicable to both Lite PAQ compression and CTW compression with impressive improvement.

Table II presents the average traffic reduction over all the fifteen users with different compression algorithms. Using the non-parametric clustering scheme, we compare the overall improvement of both dictionary-based compressor (Gzip) [39] and statistical compressor (Lite PAQ and CTW). As can be seen, Lite PAQ (which is close to the state-of-the art in compression) achieves nearly 70% traffic reduction and CTW achieves 65% reduction. With more than 65% traffic reduction, statistical compression outperforms dictionary-based compression, which offers 60% reduction. However, dictionary-based compression tends to have ten times higher compression speed. Wireless applications tolerate more latency compared to the wired networks. Hence, statistical compression is more suitable for wireless data compression while dictionary-based compression is likely to be employed in wired networks.

TABLE II
THE AVERAGE COMPRESSION RATE (BITS/BYTE) OF DIFFERENT COMPRESSION SCHEMES ON THE REAL NETWORK TRAFFIC TRACES.

	Ucomp	UcompM1	UcompMc
Gzip	6.01	4.95	3.14
CTW	5.10	3.85	2.77
Lite PAQ	4.66	3.25	2.43

C. Clustering Algorithm Performance Comparison

We choose packet selection with two algorithms, namely, k-means clustering algorithm and k-nearest clustering algorithm. According to Table III, non-parametric clustering achieve very similar performance, around 8% better than k-means clustering. Besides, non-parametric clustering does not require to know the number of clusters in advance like k-means clustering. By using ball tree data structure [40], the computational cost of nearest sub-clusters search is $O(N\log(N))$, where N is the number of sub-clusters. The average size

TABLE III
AVERAGE COMPRESSION RATE (BITS/BYTE) OF UCOMPMC FOR DIFFERENT CLUSTERING SCHEMES AND COMPRESSION ALGORITHMS.

UcompMc	k-means	k-nearest
Gzip	3.75	3.14
CTW	3.02	2.77
Lite PAQ	2.63	2.43

of training packets selected by k-means clustering is around 1800 packets whereas around 200 packets by non-parametric clustering. With smaller-sized training packet selected by k-nearest clustering algorithm, the compression speed is 9 times quicker than that of k-means clustering. As the average size of clusters generated from k-means is 9 times larger than the non-parametric counterpart. Through compression performance, the k-nearest clustering algorithm is proved to be more effective in network traffic redundancy reduction than referenced k-means clustering algorithm for real world data.

IX. CONCLUSION

In this paper, we derived the fundamental limits of universal compression (with and without side information) for mixture sources. Our results showed that significant improvement can be expected from side information in the universal compression of mixture sources. Our results further demonstrate that the optimal performance using side information corresponds to that of universal compression with known source indices. Motivated by this result, we presented two clustering algorithms for the universal compression of mixture sources with side information and demonstrated their effectiveness on data gathered from real network traces.

APPENDIX

Proof of Theorem 2: Let D be the random dimension of the source parameter vector. It is straightforward to show that

$$H(X^n|\Delta) = H(X^n|\Delta, Z, D) + I(X^n; Z, D|\Delta) \quad (35)$$

Further, if Z is known, D is determined, and hence, $H(X^n|\Delta, Z, D) = H(X^n|\Delta, Z)$ which is derived in (10). On the other hand, we have

$$I(X^n; Z, D|\Delta) = I(X^n; D|\Delta) + I(X^n; Z|\Delta, D). \quad (36)$$

Let us first focus on $I(X^n; D|\Delta)$. We have

$$I(X^n; D|\Delta) = H(D|\Delta) - H(D|\Delta, X^n). \quad (37)$$

Note that $H(D|\Delta)$ is by definition equal to $H(\mathbf{v})$. Further, we can use the maximum likelihood estimator of D using x^n, asymptotically as $n \to \infty$, to consistently estimate D asymptotically almost surely [33].[8] Hence, $H(D|\Delta, X^n) = o(1)$ and $I(X^n; D|\Delta) = H(\mathbf{v}) + o(1)$.

Next, we consider $I(X^n; Z|\Delta, D)$. In this case, we have

$$I(X^n; Z|\Delta, D) = \sum_{d=1}^{d_{\max}} v_d I(X^n; Z|\Delta, D = d). \quad (38)$$

[8]An event A happens a.s. (almost surely) if and only if $\mathbb{P}[A] = 1$.

In order to analyze, we need to consider two situations. First, let $H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n$. We have

$$I(X^n; Z|\Delta, D = d) = H(Z|\Delta, D = d) - H(Z|X^n, \Delta, D = d). \quad (39)$$

Clearly, $H(Z|\Delta, D = d) = H(\hat{\mathbf{w}}_d)$ by definition. Furthermore, the maximum likelihood estimator for the source parameter vector almost surely converges to the true θ in mean square with variance $O\left(\frac{1}{n}\right)$. On the other hand, if $H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n$, let $A(n)$ contain all Δ where there exist two parameter vectors such that $||\theta^{(i)} - \theta^{(j)}|| = O\left(\frac{1}{\sqrt{n}}\right)$. It is straightforward to see that the volume of such set shrinks to zero as $n \to \infty$. Now, we only consider the set $\Lambda'(n)$ defined as

$$\Lambda'(n) = \Lambda \setminus A(n). \quad (40)$$

Then, for $\Delta \in \Lambda'(n)$, we have for any parameter vector $\theta^{(i)} \in \Delta$, all other parameter vectors are asymptotically such that $||\theta^{(i)} - \theta^{(j)}|| = \omega\left(\frac{1}{\sqrt{n}}\right)$. Hence, by picking the closest parameter vector to the maximum likelihood estimate, asymptotically we can determine Z almost surely. Hence, we deduce deduce that $H(Z|X^n, \Delta, D = d) = o(1)$ $a.s.$ Therefore, if $H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n$, then

$$I(X^n; Z|\Delta, D = d) = H(\hat{\mathbf{w}}_d) + o(1). \quad (41)$$

To complete the proof of the theorem, we need to show that if $H(\hat{\mathbf{w}}_d) \gtrsim \frac{d}{2}\log n$, we have $I(X^n; Z|\Delta, D = d) = \bar{R}_{n,d} + o(1)$. In this case, $K \to \infty$ as $n \to \infty$, and hence, for any $\epsilon > 0$, there exists a subset Δ'_d of the K_d vectors of the d-dimensional parameter vectors indexed with K'_d, with normalized weight vector $\hat{\mathbf{u}}_d$, such that

$$(1 - 2\epsilon)\bar{R}_{n,d} < H(\hat{\mathbf{u}}_\mathbf{d}) < (1 - \epsilon)\bar{R}_{n,d}. \quad (42)$$

Let $\mathbb{I}_{\Delta'_d}$ denote the indicator function of the subset Δ'_d. It is straightforward to show that

$$I(X^n; Z|\Delta, D = d) \geq I(X^n; Z|\mathbb{I}_{\Delta'_d}, \Delta, D = d) \quad (43)$$
$$\geq I(X^n; Z|\mathbb{I}_{\Delta'_d} = 1, \Delta, D = d). \quad (44)$$

Note that $\bar{R}_{n,d} \sim \frac{d}{2}\log n$ and hence $H(\hat{\mathbf{u}}_\mathbf{d}) \lesssim \frac{d}{2}\log n$. Therefore, we have $I(X^n; Z|\mathbb{I}_{\Delta'_d} = 1, \Delta, D = d) \geq (1 - 2\epsilon)\bar{R}_{n,d} + o(1)$ almost surely. On the other hand, we also have

$$I(X^n; Z|\Delta, D = d) \leq I(X^n; \theta^{(Z)}|D = d) = \bar{R}_{n,d}. \quad (45)$$

Hence, we deduce that $I(X^n; Z|\Delta, D = d) = \bar{R}_{n,d} + o(1)$ almost surely, completing the proof. ∎

Proof of Theorem 5: The equivalence of the average minimax redundancy and the average maximin redundancy and the channel capacity above is a direct consequence of Theorem 5 of Gallager in [18]. Next, let $\theta^{(i)}$ and $\theta^{(j)}$ be independently chosen according to Jeffreys' prior on the d_i-dimensional space Λ_{d_i}. Then, if $H(\hat{\mathbf{w}}_d) \lesssim \frac{d}{2}\log n$ almost surely, as $n \to \infty$, we have $\theta^{(i)}$ and $\theta^{(j)}$ are $\omega(\frac{1}{\sqrt{n}})$ apart. Hence, this choice will maximize the mutual information asymptotically almost surely. On the other hand, if $H(\mathbf{w}_d)$, almost surely you have too many parameter vectors that you cannot discriminate them, and hence, the mutual information

is almost surely asymptotically vanishing regardless of how they are distributed. ∎

Proof of Theorem 7: In light of (20), we would need to derive $I(X^n;\Delta)$. Observe that by the chain rule we have

$$\begin{aligned} I(X^n;\Delta,Z,D) &= I(X^n;\Delta) \\ &\quad + I(X^n;D|\Delta) \\ &\quad + I(X^n;Z|D,\Delta) \end{aligned} \tag{46}$$

where D is the random dimension of the source parameter vector. By applying the chain rule in a different order we get

$$\begin{aligned} I(X^n;\Delta,Z,D) &= I(X^n;D) \\ &\quad + I(X^n;Z|D) \\ &\quad + I(X^n;\Delta|Z,D) \end{aligned} \tag{47}$$

Note that $I(X^n;Z|D)=0$ as the random vector X^n would not decrease the uncertainty in the index of the source Z as there is no information about the source parameter vectors. Next, consider $I(X^n;D)$. In light of [33]–[35] D is the random dimension of the signal can be determined uniquely as n grows to infinity, i.e., $\lim_{n\to\infty} H(D|X^n)=0$. Hence,

$$I(X^n;D) = H(D) - H(D|X^n) = H(D) + o(1) = H(\mathbf{v}) + o(1). \tag{48}$$

Similarly, $I(X^n;D|\Delta) = H(\mathbf{v}) + o(1)$ as $H(D|\Delta) = H(D)$. Further, $I(X^n;Z|D,\Delta)$ is calculated in the proof of Theorem 2. Finally, to derive $I(X^n;\Delta|Z,D)$ note that the parameter vectors are chosen independently, and hence, we have $I(X^n;\Delta|Z,D) = I(X^n;\theta^{(Z)}|Z,D)$. On the other hand, as each of the unknown parameter vectors follow Jeffreys' prior, we have $I(X^n;\theta^{(Z)}|Z=z,D=d) = \bar{R}_{n,d}$. Thus,

$$I(X^n;\theta^{(Z)}|Z,D) = \sum_{d=1}^{d_{\max}} v_d \sum_{i=1}^{K} \hat{w}_{d,i} I(X^n;\theta^{(Z)}|Z=z) \tag{49}$$

$$= \sum_{d=1}^{d_{\max}} v_d \bar{R}_{n,d}. \tag{50}$$

By combining (46) and (47) and the above, we arrive at the desired result. ∎

Proof of Theorem 10: In the case of UcompM, we need to derive $I(X^n;\Delta|\mathbf{Y}^{n,T})$. Using the chain rule we have the following.

$$\begin{aligned} I(X^n;\Delta,\mathbf{S},Z,D|\mathbf{Y}^{n,T}) &= I(X^n;\Delta|\mathbf{Y}^{n,T}) \\ &\quad + I(X^n;\mathbf{S},Z,D|\mathbf{Y}^{n,T},\Delta). \end{aligned} \tag{51}$$

On the other hand, (X^n,Z,D) is independent of $(\mathbf{Y}^{n,T},\mathbf{S})$ given Δ. Hence,

$$I(X^n;\mathbf{S},Z,D|\mathbf{Y}^{n,T},\Delta) = I(X^n;Z,D|\Delta), \tag{52}$$

which has been characterized in the proof of Theorem 7. Applying the chain rule in a different order, we get

$$\begin{aligned} I(X^n;\Delta,\mathbf{S},Z,D|\mathbf{Y}^{n,T}) &= I(X^n;\mathbf{S},Z,D|\mathbf{Y}^{n,T}) \\ &\quad + I(X^n;\Delta|\mathbf{Y}^{n,T},\mathbf{S},Z,D). \end{aligned} \tag{53}$$

Now, considering $I(X^n;\mathbf{S},Z,D|\mathbf{Y}^{n,T})$ observe that

$$\begin{aligned} I(X^n;\mathbf{S},Z,D|\mathbf{Y}^{n,T}) &= I(X^n;Z,D|\mathbf{Y}^{n,T}) \\ &\quad + I(X^n;\mathbf{S}|\mathbf{Y}^{n,T},Z,D). \end{aligned} \tag{54}$$

Observe that $I(X^n;Z,D|\mathbf{Y}^{n,T})$ can be made arbitrarily close to $I(X^n;Z,D|\Delta)$ with T, i.e., $\forall\delta\ \exists T_0$ such that for $T>T_0$,

$$I(X^n;Z,D|\mathbf{Y}^{n,T}) - I(X^n;Z,D|\Delta) < \delta, \tag{55}$$

and $I(X^n;Z,D|\Delta)$ is characterized in the proof of Theorem 7. Further note that $I(X^n;\mathbf{S}|\mathbf{Y}^{n,T},Z,D)$ can be made arbitrarily small with T, i.e., $\forall\delta\ \exists T_1$ such that $I(X^n;\mathbf{S}|\mathbf{Y}^{n,T},Z,D)<\delta$ for $T>T_1$.

Now, we only need to derive $I(X^n;\Delta|\mathbf{Y}^{n,T},\mathbf{S},Z,D)$ in (53). We have

$$\begin{aligned} I(X^n;\Delta|\mathbf{Y}^{n,T},\mathbf{S},Z,D) &= I(X^n;\theta^{(Z)}|\mathbf{Y}^{n,T},\mathbf{S},Z,D) \\ &\quad + \sum_{\imath\neq Z} I(X^n;\theta^{(i)}|\mathbf{Y}^{n,T},\mathbf{S},Z,D,\theta^{(Z)},\theta^{(1)},\ldots,\theta^{(i-1)}) \end{aligned} \tag{56}$$

All the summands of the second term are zero as X^n is independent of all $\theta^{(i)}$ $(i\neq Z)$ given Z. On the other hand, observe that

$$I(X^n;\theta^{(Z)}|\mathbf{Y}^{n,T},\mathbf{S},Z,D) = I(X^n;\theta^{(Z)}|\{Y^n(t)\}_{S(t)=Z},Z,D). \tag{57}$$

The size of $\frac{1}{T}|\{Y^n(t)\}_{S(t)=Z}|$ can be made arbitrarily close to $\hat{w}_{D,Z}$ for sufficiently large T. On the other hand, in (57) the side information is from a single source. Hence, the mutual information can be obtained using Theorem 2 of [20]. Combining all these pieces results in the desired result. ∎

Proof of Theorem 11: Observe that

$$\begin{aligned} I(X^n;\Delta,\mathbf{S},Z,D|\mathbf{Y}^{n,T}) &= I(X^n;\Delta|\mathbf{Y}^{n,T}) \\ &\quad + I(X^n;\mathbf{S},Z,D|\mathbf{Y}^{n,T},\Delta). \end{aligned} \tag{58}$$

The first term was characterized in Theorem 10 and the second term is equal to $I(X^n;Z,D|\Delta)$, which was derived in the proof of Theorem 7. Applying the chain rule in a different order we have

$$\begin{aligned} I(X^n;\Delta,\mathbf{S},Z,D|\mathbf{Y}^{n,T}) &= I(X^n;\mathbf{S},Z|\mathbf{Y}^{n,T}) \\ &\quad + I(X^n;\Delta|\mathbf{Y}^{n,T},\mathbf{S},Z) \\ &\quad + I(X^n;D|\Delta,\mathbf{Y}^{n,T},\mathbf{S},Z). \end{aligned} \tag{59}$$

The second term in the expansion is what we are after while $I(X^n;D|\Delta,\mathbf{Y}^{n,T},\mathbf{S},Z) = 0$. Considering $I(X^n;\mathbf{S},Z|\mathbf{Y}^{n,T})$, we have

$$\begin{aligned} I(X^n;\mathbf{S},Z|\mathbf{Y}^{n,T}) &= I(X^n;Z|\mathbf{Y}^{n,T}) \\ &\quad + I(X^n;\mathbf{S}|\mathbf{Y}^{n,T},Z). \end{aligned} \tag{60}$$

Using similar arguments as in the proof of Theorem 10, we can make $I(X^n;Z|\mathbf{Y}^{n,T})$ and $I(X^n;\mathbf{S}|\mathbf{Y}^{n,T},Z)$ arbitrarily close to $I(X^n;Z|\Delta)$ and zero, respectively, for sufficiently large T. Putting all these facts together, we conclude that

$$I(X^n;\Delta|\mathbf{Y}^{n,T},\mathbf{S},Z) = I(X^n;\Delta|\mathbf{Y}^{n,T}) + \delta, \tag{61}$$

where δ can be made arbitrarily small for sufficiently large T, which completes the proof. ∎

REFERENCES

[1] M. Sardari, A. Beirami, J. Zou, and F. Fekri, "Content-aware network data compression using joint memorization and clustering," in *2013 IEEE Conference on Computer Networks (INFOCOM 2013)*, Apr. 2013.

[2] A. Beirami, M. Sardari, and F. Fekri, "Results on the optimal memory-assisted universal compression performance for mixture sources," in *51st Annual Allerton Conference*, Oct. 2013, pp. 890–895.

[3] A. Beirami, L. Huang, M. Sardari, and F. Fekri, "On optimality of data clustering for packet-level memory-assisted compression of network traffic," in *15th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2014)*, Toronto, Canada, June 2014.

[4] L. Davisson, "Universal noiseless coding," *IEEE Trans. Inf. Theory*, vol. 19, no. 6, pp. 783 – 795, Nov. 1973.

[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[6] F. Willems, Y. Shtarkov, and T. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[7] J. Rissanen and G. Langdon Jr., "Universal modeling and coding," *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 12 – 23, Jan. 1981.

[8] M. Feder and N. Merhav, "Hierarchical universal coding," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1354 –1364, Sept. 1996.

[9] M. Effros, K. Visweswariah, S. Kulkarni, and S. Verdu, "Universal lossless source coding with the Burrows Wheeler transform ," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1061–1081, May 2002.

[10] D. Baron and Y. Bresler, "An O(N) semipredictive universal encoder via the BWT," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 928–937, May 2004.

[11] A. R. Barron and T. M. Cover, "Minimum complexity density estimation," *IEEE Trans. Inf. Theory*, vol. 37, no. 4, pp. 1034–1054, Jul. 1991.

[12] R. E. Krichevsky and V. K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inf. Theory*, vol. 27, no. 2, pp. 199–207, 1981.

[13] A. Beirami and F. Fekri, "Results on the redundancy of universal compression for finite-length sequences," in *2011 IEEE International Symposium on Information Theory (ISIT '11)*, Jul. 2011, pp. 1604–1608.

[14] N. Merhav and M. Feder, "A strong version of the redundancy-capacity theorem of universal coding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 714 –722, May 1995.

[15] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[16] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[17] T. Berger, Z. Zhang, and H. Viswanathan, "The CEO problem," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 887 –902, May 1996.

[18] R. G. Gallager, "Source coding with side information and universal coding," *unpublished*.

[19] A. Beirami, M. Sardari, and F. Fekri, "Results on the fundamental gain of memory-assisted universal source coding," in *2012 IEEE International Symposium on Information Theory (ISIT '12)*, Jul. 2012, pp. 1087–1091.

[20] A. Beirami and F. Fekri, "On lossless universal compression of distributed identical sources," in *2012 IEEE International Symposium on Information Theory (ISIT '12)*, Jul. 2012, pp. 561–565.

[21] Z. Zhuang, C.-L. Tsao, and R. Sivakumar, "Curing the amnesia: Network memory for the Internet, Tech. Report," 2009. [Online]. Available: http://www.ece.gatech.edu/research/GNAN/archive/tr-nm.pdf

[22] S. Sanadhya, R. Sivakumar, K.-H. Kim, P. Congdon, S. Lakshmanan, and J. P. Singh, "Asymmetric caching: improved network deduplication for mobile devices," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 161–172. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348565

[23] M. Sardari, A. Beirami, and F. Fekri, "Memory-assisted universal compression of network flows," in *2012 International Conference on Computer Communications (INFOCOM '12)*, Mar. 2012, pp. 91–99.

[24] A. Beirami, M. Sardari, and F. Fekri, "Packet-level network compression: Realization and scaling of the network-wide benefits," *arXiv preprint arXiv:1411.6359*, 2014.

[25] N. Krishnan, D. Baron, and M. K. Mihcak, "A parallel two-pass MDL context tree algorithm for universal source coding," in *2014 IEEE International Symposium on Information Theory Proceedings (ISIT '14)*, Jul. 2014.

[26] N. Krishnan and D. Baron, "A universal parallel two-pass mdl context tree compression algorithm," *arXiv preprint arXiv:1407.1514*, 2014.

[27] M. Sardari, A. Beirami, and F. Fekri, "On the network-wide gain of memory-assisted source coding," in *2011 IEEE Information Theory Workshop (ITW '11)*, Oct. 2011, pp. 476–480.

[28] M. Drmota and W. Szpankowski, "Precise minimax redundancy and regret," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2686–2707, Nov. 2004.

[29] W. Szpankowski, "Asymptotic average redundancy of Huffman (and other) block codes ," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2434–2443, Nov. 2000.

[30] B. Clarke and A. Barron, "Information-theoretic asymptotics of Bayes methods," *IEEE Trans. Inf. Theory*, vol. 36, no. 3, pp. 453 –471, May 1990.

[31] K. Atteson, "The asymptotic redundancy of Bayes rules for Markov chains," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2104 –2109, Sept. 1999.

[32] Q. Xie and A. Barron, "Minimax redundancy for the class of memoryless sources," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 646 –657, Mar. 1997.

[33] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1007 –1016, Mar. 2006.

[34] L. Finesso, C.-C. Liu, and P. Narayan, "The optimal error exponent for markov order estimation," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1488–1497, Sept. 1996.

[35] J. C. Kieffer, "Strongly consistent code-based identification and order estimation for constrained finite-state model classes," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 893–902, May 1993.

[36] L. L. Cam and G. L. Yang, *Asymptotics in Statistics: Some Basic Concepts*. Springer, 2000.

[37] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[38] S. Sanadhya, R. Sivakumar, K.-H. Kim, P. Congdon, S. Lakshmanan, and J. Singh, "Asymmetric caching: Improved deduplication for mobile devices," in *Proceedings of the ACM MOBICOM 2012 conference*. ACM, 2012.

[39] L. P. Deutsch, "Gzip file format specification version 4.3," 1996.

[40] S. M. Omohundro, *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.