Optimal Order Reduction of Probability Distributions by
Maximizing Mutual Information

M. Vidyasagar

Abstract— In a companion paper [16], we defined a metric distance between two probability distributions \( \phi, \psi \) defined on sets of different cardinality, called the variation of information metric \( d \). In this paper we study of problem of finding an optimal reduced-order approximation in the variation of information metric. Let \( \phi \) denote the probability distribution of high dimension that is to be approximated. It is shown first that any optimal approximation of \( \phi \) must be an aggregation of \( \phi \). Then it is shown that any optimal aggregation of \( \phi \) is one that has maximum entropy. Using these two results, we then formulate the problem of optimal order reduction as a nonstandard bin-packing problem with overstuffing. Unfortunately this problem is NP-hard. So a greedy algorithm is presented to solve this problem, and an upper bound on its performance is presented. The application of the greedy algorithm is illustrated via a large example.

I. INTRODUCTION

In a companion paper [16], we defined a metric between two probability distributions, say \( \phi, \psi \), defined on sets of different cardinalities, called the variation of information metric \( d \). If there is a way of measuring the ‘distance’ between a high-dimensional probability distribution and a lower-dimensional probability distribution, it is natural to study the problem of optimal order reduction. Specifically, suppose \( \phi \) is an \( n \)-dimensional probability distribution, and \( m < n \) is a specified integer. Then one can ask: What is the (or an) ‘optimal’ \( m \)-dimensional approximation \( \psi \) to \( \phi \) in the sense of minimizing the variation of information metric \( d(\phi, \psi) \)? That is the question studied in this paper. The question is answered via several steps. First, it is shown that any optimal approximation of \( \phi \) must in fact be an aggregation of \( \phi \). Then it is shown that an aggregation of \( \phi \) is optimal if and only if it has maximum entropy amongst all aggregations. Thus the optimal order reduction problem is equivalent to maximizing entropy via aggregation. This problem is then reformulated as a nonstandard bin-packing problem with overstuffing. Unfortunately this problem is NP-hard. So we propose a greedy algorithm to construct a suboptimal solution, and also prove an upper bound on its performance. The approach is illustrated through a large example.

II. PRELIMINARIES

In this section we reprise some relevant results from [16].

A. Notation

Let \( e \) denote the column vector of all one’s, and the subscript denote its dimension. Thus \( e_n \) is a column vector of \( n \) one’s. A matrix \( P \in [0,1]^n \times m \) is said to be stochastic if \( Pe_m = e_n \), that is, all row sums of \( P \) are equal to one. Note that \( P \) need not be a square matrix; but this definition is consistent with the more familiar usage for square matrices. The set of \( n \times m \) stochastic matrices is denoted by \( \mathbb{S}_{n \times m} \). If we take the degenerate case of \( m = 1 \), then the symbol \( \mathbb{S}_n \) denotes the set of nonnegative (row) vectors that add up to one. Clearly \( \mathbb{S}_n \) can be identified with the set \( \mathcal{M}(\mathbb{A}) \) of all probability distributions on \( \mathbb{A} \) on a set of cardinality \( n \).

Throughout, the function \( h : [0,1] \rightarrow \mathbb{R}_+ \) is defined by \( h(r) = -r \log r \), with the standard convention that \( h(0) = 0 \). Note that \( h \) is continuously differentiable except at \( r = 0 \), and that \( h'(r) = -(1 + \log r) \). Throughout, we use the symbol \( H \) to denote the Shannon entropy of a probability distribution. Thus if \( \phi \in \mathbb{S}_n \), then

\[
H(\phi) = -\sum_{i=1}^{n} \phi_i \log \phi_i = \sum_{i=1}^{n} h(\phi_i).
\]

B. Reprise of Relevant Results from [16]

Suppose \( \phi \in \mathbb{S}_n, \psi \in \mathbb{S}_m \). Then we can think of \( \phi, \psi \) as probability distributions on some sets \( \mathbb{A}, \mathbb{B} \) respectively where \( |\mathbb{A}| = n, |\mathbb{B}| = m \). We can also think of \( \phi, \psi \) as probability distributions of some random variables \( X, Y \) assuming values in \( \mathbb{A}, \mathbb{B} \) respectively. Let \( \theta \) denote the joint distribution of \( (X,Y) \), so that \( \theta \in \mathcal{M}(\mathbb{A} \times \mathbb{B}) \), and let \( \theta_{\mathbb{A}}, \theta_{\mathbb{B}} \) denote its marginals on \( \mathbb{A}, \mathbb{B} \) respectively. With this convention, we define the following quantities:

\[
W(\phi, \psi) := \min_{\theta \in \mathcal{M}(\mathbb{A} \times \mathbb{B})} \{ H(\theta) : \theta_{\mathbb{A}} = \phi, \theta_{\mathbb{B}} = \psi \},
\]

\[
V(\phi, \psi) := W(\phi, \psi) - H(\phi).
\]

Compare with [16], Equations (5) and (6). Then it is possible to define a metric between \( \phi, \psi \).

Definition 1: Suppose \( \phi \in \mathbb{S}_n, \psi \in \mathbb{S}_m \) and let \( W(\phi, \psi) \) be as in (6). Then

\[
d(\phi, \psi) := V(\phi, \psi) + V(\psi, \phi)
\]

is called the variation of information metric between \( \phi \) and \( \psi \).
The salient properties of the function $d$ are brought out next.

**Theorem 1:** The functions $d$ defined in (3) is a pseudo-metric.

The computation of the quantity $d$ relies on the computation of $V(\phi, \psi)$. This can be achieved by formulating an associated optimization problem. Given $\phi \in S_n$, define the function $J_\phi : S_{n \times m} \rightarrow \mathbb{R}_+$ by

$$J_\phi(P) = \sum_{i=1}^n \phi_i H(p_i),$$

where $p_i$ is the $i$-th row of $P$. Then

$$V(\phi, \psi) = \min_{P \in S_{n \times m}} J_\phi(P) \text{ s.t. } \phi P = \psi.$$

The solution of minimizing $J_\phi(P)$ with respect to $P$ is the main topic of [16].

**III. ALL OPTIMAL REDUCED-ORDER APPROXIMATIONS ARE AGGREGATIONS**

Once we have a way of quantifying the distance between probability distributions having different dimensions, it is natural to examine the problem of approximating a distribution $\phi \in S_n$ by another $\psi \in S_m$ where $m < n$, such that the distance between them is as small as possible. This may be referred to as the ‘order reduction’ problem. This is precisely the problem studied in the present paper, namely: Given a distribution $\phi \in S_n$, and an integer $m < n$ (perhaps $m \ll n$), find a $\psi \in S_m$ such that $d(\phi, \psi)$ is as small as possible.

Given $\phi \in S_n$, let us refer to $\phi^{(a)}$ as an aggregation of $\phi$ if it can be obtained by aggregating the components of $\phi$. In other words, $\phi^{(a)}$ is an aggregation of $\phi$ if there exists a partition of $\{1, \ldots, n\}$ into $m$ pairwise disjoint sets $I_1, \ldots, I_m$ such that

$$\phi_{i_j}^{(a)} = \sum_{i \in I_j} \phi_i, j = 1, \ldots, m.$$ 

An equivalent way of saying the same thing is the following: Suppose $m < n$. Then $\psi \in S_m$ is an aggregation of $\phi \in S_n$ if and only if there exists a matrix $P \in S_{n \times m}$ such that $\psi = \phi P$, and in addition, $p_{ij} = 0$ or 1 for all $i, j$. Note that the two conditions $P \in S_{n \times m}$ and $p_{ij}$ equals 0 or 1 ensure that every row of $P$ consists of a degenerate probability distribution, with a solitary component equal to 1 and the rest equal to zero.

In this section, it is shown that, given a distribution $\phi \in S_n$, an optimal approximation of $\phi$ in $S_m, m < n$, in terms of the variation of information metric, must be an aggregation of $\phi$. Unfortunately the proof of this intuitive result is very long.

**Theorem 2:** Suppose $\phi \in S_n, \psi \in S_m, m < n$, and that $\psi$ is not an aggregation of $\phi$. Then there exists a $\psi' \in S_m$ such that $d(\phi, \psi') < d(\phi, \psi)$.

The proof of the theorem makes use of a couple of preliminary lemmas.

**Lemma 1:** Suppose $\mu \in \mathbb{R}_+^m$, $\mu \neq 0$. Then

$$\sum_{j=1}^m h(\mu_j) = cH((1/c)\mu) + h(c),$$

where $c = \mu e_m$ is a normalizing constant.

**Proof:** We have that

$$\sum_{j=1}^m h(\mu_j) = \sum_{j=1}^m \mu_j \log \frac{1}{\mu_j} = \sum_{j=1}^m \mu_j \log \frac{c}{\mu_j} - \left( \sum_{j=1}^m \mu_j \right) \log c = c \sum_{j=1}^m \frac{\mu_j}{c} \log \frac{c}{\mu_j} - c \log c = cH((1/c)\mu) + h(c).$$

This completes the proof.

**Lemma 2:** Suppose $c_1, c_2, b > 0$ with $c_1 + c_2 + b = 1$. For each $\lambda \in [0, 1]$, define $\psi(\lambda) \in S_2$ by

$$\psi(\lambda) = [c_1 + \lambda b \quad c_2 + (1 - \lambda)b],$$

and $G : [0, 1] \rightarrow \mathbb{R}$ by

$$G(\lambda) = bH([\lambda - 1]) - H(\psi(\lambda)).$$

Then

$$G(\lambda) > \min\{G(0), G(1)\} = \min\{-H(\psi(0)), -H(\psi(1))\}.$$  

**Proof:** This follows from elementary calculus. Recall that for the function $h(r) = r \log(1/r)$, we have that $h'(r) = -(1 + \log r)$ for all $r > 0$. Now expand $G(\lambda)$ as

$$G(\lambda) = b(h(\lambda) + h(1 - \lambda)) - h(c_1 + \lambda b) - h(c_2 + (1 - \lambda)b).$$

Then it follows that

$$G'(\lambda) = -b(1 + \log \lambda) + b(1 + \log(1 - \lambda))$$

$$+ b(1 + \log(c_1 + \lambda b) - b(1 + \log(c_2 + (1 - \lambda)b))$$

$$= b \log \left[ \frac{(c_1 + \lambda b) \cdot (1 - \lambda)}{(c_2 + (1 - \lambda)b) \cdot \lambda} \right]$$

$$= b \log \left[ \frac{c_1 - c_1 \lambda + b \lambda - b \lambda^2}{c_2 \lambda + b \lambda - b \lambda^2} \right].$$

From the above, it is clear that $G'(\lambda) = 0$ when

$$c_1 - c_1 \lambda + b \lambda - b \lambda^2 = c_2 \lambda + b \lambda - b \lambda^2,$$

or

$$c_1 - c_1 \lambda = c_2 \lambda, \lambda = \frac{c_1}{c_1 + c_2} =: \lambda^*.$$ 

Now if $\lambda > \lambda^*$, then

$$c_1 < (c_1 + c_2) \lambda, \text{ or } c_1 - c_1 \lambda < c_2 \lambda.$$ 

So the numerator in the fraction above is smaller than the denominator, and as a result $G'(\lambda) < 0$ if $\lambda > \lambda^*$. Similar reasoning shows that $G'(\lambda) > 0$ if $\lambda < \lambda^*$. So $G(\lambda)$ attains its maximum when $\lambda = \lambda^*$, and decreases on either side of
λ∗. So in particular, if λ < λ∗, then G(λ) > G(0), whereas if λ > λ∗, then G(λ) > G(1). In either case, (7) is satisfied.

**Proof (of Theorem 2):** Now we give a proof of Theorem 2. Suppose φ ∈ S_n, ψ ∈ S_m, m < n, and ψ is not an aggregation of φ. Choose P ∈ S_{n×m} such that φP = ψ and Jφ(P) = V(φ, ψ). Since ψ is not an aggregation of φ, at least one row of P contains at least two nonzero (i.e., positive) elements. Let k be such a row, and without loss of generality permute the components of ψ in such a way that \( p_{k1} > 0, p_{k2} > 0 \). To show that ψ cannot be an optimal approximation of φ in the d metric, we will construct another distribution \( ψ' \in S_m \) that matches ψ from component 3 onwards. We will do this by perturbing only the two elements \( p_{k1}, p_{k2} \) in such a way that \( p'_{k1} + p'_{k2} = p_{k1} + p_{k2} \), and defining \( ψ' = φP' \). This means that many of the quantities common to ψ and ψ′, so in the various equations below, we will just write ‘constant’ or ‘const.’ to avoid notational clutter.

From the manner in which P was chosen, it follows that

\[
V(φ, ψ) = \sum_{i=1}^{n} φ_i H(p_i)
= φ_k H(p_k) + \text{const},
\]

\[
V(ψ, φ) = V(φ, ψ) + H(φ) - H(ψ)
= φ_k [h(p_{k1}) + h(p_{k2})] - [h(ψ_1) + h(ψ_2)] + \text{const}.
\]

Note that the ‘constant’ in the two equations need not be the same. Our use of the phrase ‘constant’ means only that all the ignored summations remain unchanged when we replace ψ by ψ′. Proceeding further, let us write

\[
ψ_1 = \sum_{i=1}^{n} φ_i p_{i1} = \sum_{i \neq k} φ_i p_{i1} + φ_k p_{k1} =: d_1 + φ_k p_{k1}.
\]

Similarly,

\[
ψ_2 = \sum_{i=1}^{n} φ_i p_{i2} = \sum_{i \neq k} φ_i p_{i2} + φ_k p_{k2} =: d_2 + φ_k p_{k2}.
\]

With these definitions, we can write

\[
V(ψ, φ) = φ_k [h(p_{k1}) + h(p_{k2})] - [h(ψ_1) + h(ψ_2)] + \text{const},
\]

\[
(8)
\]

This looks similar to the function \( G(λ) \) in Lemma 2, except that neither \( p_{k1} + p_{k2} \) nor \( d_1 + d_2 + φ_k \) necessarily add up to one. So we proceed as in the proof of Lemma 2 and apply the correction terms from Lemma 1 wherever necessary. Let us define

\[
λ = \frac{p_{k1}}{p_{k1} + p_{k2}}, 1 - λ = \frac{p_{k2}}{p_{k1} + p_{k2}},
β = p_{k1} + p_{k2}, α = d_1 + d_2 + βφ_k,
\]

and note that

\[
ψ_1 + ψ_2 = d_1 + d_2 + βφ_k = α.
\]

With these definitions, and making repeated use of Lemma 1, we get

\[
φ_k [h(p_{k1}) + h(p_{k2})] = βφ_k H([λ 1 - λ]) + φ_k h(β),
\]

\[
h(ψ_1) + h(ψ_2) = α H(γ) + h(α),
\]

where

\[
γ = \left[ \frac{d_1}{α} + λ \frac{βφ_k}{α} \frac{d_2}{α} + (1 - λ) \frac{βφ_k}{α} \right] ∈ S_2.
\]

Hence

\[
V(ψ, φ) = α \left[ \frac{βφ_k}{α} H([λ 1 - λ]) + H(γ) \right] + φ_k h(β) - h(α) + \text{const}.
\]

Now the quantity inside the brackets is like \( G(λ) \) in Lemma 2, with

\[
c_1 = \frac{d_1}{α}, c_2 = \frac{d_2}{α}, b = \frac{βφ_k}{α}.
\]

And these three numbers do add up to one. So we know from Lemma 2 that the quantity inside the brackets can be made smaller by choosing \( λ = 0 \) or \( 1 \). The choice \( λ = 0 \) causes \( p_{k1}, p_{k2}, ψ_1, ψ_2 \) to be replaced by

\[
[p'_{k1} p'_{k2}] = [0 p_{k1} + p_{k2}],
\]

\[
[ψ'_1, ψ'_2] = [d_1 d_2 + φ_k (p_{k1} + p_{k2})],
\]

while the choice \( λ = 1 \) causes \( p_{k1}, p_{k2}, ψ_1, ψ_2 \) to be replaced by

\[
[p'_{k1} p'_{k2}] = [p_{k1} + p_{k2} 0],
\]

\[
[ψ'_1, ψ'_2] = [d_1 + φ_k (p_{k1} + p_{k2}) d_2].
\]

In either case the numbers \( β, α \) remain the same, whence the correction term \( φ_k h(β) - h(α) \) also remains the same. So decreasing the quantity inside the brackets reduces \( V(ψ, φ) \).

So the conclusion is that there exists a \( P' ∈ S_{n×m} \) such that, with \( ψ' = φP' \), we have

\[
J_φ(P') + H(φ) - H(ψ') = φ_k [h(p'_{k1}) + h(p'_{k2})] - [h(ψ'_1) + h(ψ'_2)] + \text{const} < φ_k [h(p_{k1}) + h(p_{k2})] - [h(ψ_1) + h(ψ_2)] + \text{const} = V(ψ, φ).
\]

Now, since \( V(ψ, φ') \) is the minimum of the quantity \( J_φ(Q) \) over all \( Q ∈ S_{n×m} \) such that \( φQ = ψ' \), we conclude from the above that

\[
V(φ, ψ') + H(φ) - H(ψ') < V(φ, φ),
\]

or equivalently that

\[
V(ψ', φ) < V(ψ, φ).
\]

718
Similarly, we can compare \( V(\phi, \psi') \) and \( V(\phi, \psi) \). Since 
\[
p'_{k_1} + p'_{k_2} = p_{k_1} + p_{k_2},
\]
and one of \( p'_{k_1}, p'_{k_2} \) is zero, it is obvious that 
\[
\phi_k[h(p'_{k_1}) + h(p'_{k_2})] < \phi_k[h(p_{k_1}) + h(p_{k_2})].
\]
Hence
\[
J_\phi(P') < J_\phi(P) = V(\phi, \psi).
\]
As a consequence, we have as before that
\[
\sum_{i=1}^{n} |\alpha_i - \beta_j| = \sum_{j \in B} \max\{\alpha_j - \beta_j, 0\}
\]
\[
+ \sum_{j \in B} \max\{\beta_j - \alpha_j, 0\}.
\]

IV. FINDING AN OPTIMAL AGGREGATION: A REFORMULATION

Now that we know that any reduced-order approximation in the variation of information metric must be an aggregation, the next logical step is to characterize the distance between a distribution and its aggregations, and choose one aggregation that is closest to the original distribution. That is the objective of this section. We first show that minimizing the distance between a given distribution \( \phi \) and its aggregation \( \phi^{(a)} \) is equivalent to maximizing the entropy of \( \phi^{(a)} \). Since there are \( O(m^n) \) aggregations, finding one with maximum entropy turns out to be NP-hard (not surprisingly). So we reformulate the problem as a bin-packing problem with over-stuffing, and propose a greedy algorithm. A worst-case performance bound for the greedy algorithm is derived.

Now we can ask: What is the best possible aggregation \( \phi^{(a)} \) that is ‘closest’ to \( \phi \)? If \( \phi^{(a)} \) is an aggregation of \( \phi \), it is obvious that \( V(\phi, \phi^{(a)}) = 0 \). By (4) it follows that \( V(\phi^{(a)}, \phi) = H(\phi) - H(\phi^{(a)}) \). Applying the definitions of \( d \) shows that 
\[
d(\phi, \phi^{(a)}) = H(\phi) - H(\phi^{(a)}),
\]
Since \( H(\phi) \) is a part of the data, minimizing \( d(\phi, \phi^{(a)}) \) requires us to maximize the entropy \( H(\phi^{(a)}) \). This leads to the

The Optimal Aggregation Problem to Maximize Entropy: Given \( \phi \in S_n \) and an integer \( m < n \), find an aggregation of \( \phi \) into \( S_m \) with maximum entropy.

Note that if \( m = 2 \), the aggregation \( \phi^{(a)} \) has maximum entropy if and only if it is closest to the uniform vector \( u_2 \) in the total variation metric. In turn, finding an aggregation such that \( \rho(\phi^{(a)}, u_2) \) is minimized (where \( \rho \) denotes the total variation metric) is equivalent to a bin-packing problem with over-stuffing where both bin sizes are equal, and this problem is NP-hard. Hence it is plausible that the above problem is also NP-hard. Moreover, a natural suboptimal algorithm is also not readily available, unless we reformulate the problem, which is the next step.

We observe that amongst all distributions in \( S_m \), the uniform distribution has the maximum entropy. Thus we attempt to aggregate \( \phi \) in such a way that every component of \( \phi^{(a)} \) is as close as possible to \( 1/m \). That problem is a special case of aggregating \( \phi \) in such a way that every component of \( \phi^{(a)} \) is as close as possible to the corresponding component of a given distribution \( \psi \in S_m \), which need not be the uniform distribution. This more general problem is formulated in [14], as a follow up to earlier work in [6], [7], [8], and can be stated as follows.

Optimal Aggregation in Total Variation Metric to a Desired Distribution: Given \( \phi \in S_n \), \( \psi \in S_m \), find an aggregation \( \phi^{(a)} \) of \( \phi \) such that the total variation metric \( \rho(\phi^{(a)}, \psi) \) is as small as possible, where the total variation metric \( \rho \) between \( \alpha, \beta \in S_m \) is defined as:
\[
\rho(\alpha, \beta) = \frac{1}{2} \sum_{j \in B} |\alpha_j - \beta_j| = \sum_{j \in B} \max\{\alpha_j - \beta_j, 0\}
\]
\[
+ \sum_{j \in B} \max\{\beta_j - \alpha_j, 0\}.
\]

V. A GREEDY ALGORITHM FOR OPTIMAL AGGREGATION

As shown below, the optimal aggregation problem can be formulated as a non-standard bin-packing problem. Specifically, the problem of optimal aggregation in the total variation metric can be thought of as a bin-packing problem with the longer probability distribution \( \phi_1, \ldots, \phi_n \) as the ‘list’ to be packed, and the shorter distribution \( \psi_1, \ldots, \psi_m \) as the capacity of the ‘bins’, while minimizing the unutilized capacity. This problem differs from the conventional bin-packing problem in at least three respects:

- In the standard bin-packing problem, all bins have the same capacity, whereas here they need not.
- In the standard bin-packing problem, if a list item does not fit any bin, then a new bin is created; here the number of bins is fixed.
- Since \( \sum_i \phi_i = \sum_j \psi_j = 1 \), if all list items have to be put into the available bins, then some bins need to be ‘overstuffed’, that is, have their capacity exceeded.

Fortunately, thanks to the propensity of the research community to study every possible variation of a problem, this very situation has been studied in [18]. We don’t use their results directly; rather, we adapt their method of proof to the situation at hand. First we adapt the LS algorithm to the situation where the number of bins is fixed but overstuffing is allowed.

1) Sort the elements of \( \phi, \psi \) into descending order of magnitude.
2) Set \( i \) (the round counter) to 0, and set the initial bin capacities as \( c_j = \psi_j \) for \( j = 1, \ldots, m \).
3) Increment the counter \( i \) by one until \( i = n \). Include the element \( \phi_i \) into the bin with the greatest capacity \( c_j \), and then replace \( c_j \) by \( c_j - \phi_i \). If \( c_j - \phi_i < 0 \) then put no more elements in bin \( j \). End when \( i = n \).
Theorem 3: For the LS algorithm described above, we have
\[ \rho(\phi(a), \psi) \leq 0.25m\phi_{\max}. \] (9)
where \( \phi(a) \) is the aggregation produced by the algorithm, and \( \phi_{\max} = \max_i \{\phi_i\} \).

Proof: The steps in the proof follow the corresponding steps in [18]. Once the greedy algorithm is completed, let us denote the resulting aggregation \( \phi(a) \) by \( \alpha \) to reduce clutter. Let us refer to bin \( j \) as ‘heavy’ if \( \alpha_j > \psi_j \), and ‘light’ if \( \alpha_j \leq \psi_j \). Suppose there are \( k \) heavy bins. Without loss of generality, renumber the bins such that the first \( k \) bins are heavy and the rest are light. Let \( e_1, \ldots, e_k \) denote the excess and \( s_{k+1}, \ldots, s_m \) denote the slack. In other words,
\[ e_j = \alpha_j - \psi_j, \quad j = 1, \ldots, k, \]
and
\[ s_j = \psi_j - \alpha_j, \quad j = k + 1, \ldots, m. \]
For \( j = 1, \ldots, k \), let \( r_j \) denote its excess capacity just before the last item was placed into it (making it heavy). Then two things are obvious. First, \( r_j + e_j \) equals the last component of \( \phi \) that was placed into this bin, and as a result \( r_j + e_j \leq \phi_{\max} \). Second, the nature of the LS algorithm implies that \( r_j \) is at least equal to the capacity of all the other bins at the time this item was placed into bin \( j \). Since bin capacity can only decrease as the algorithm is run, in particular this implies that
\[ r_j \geq s_{k+1}, \ldots, s_m, \quad j = 1, \ldots, k. \]
Therefore
\[ \frac{1}{k} \sum_{j=1}^{k} r_j \geq \min_{j=1,\ldots,k} r_j \geq \max_{j=k+1,\ldots,m} s_j \geq \frac{1}{m-k} \sum_{j=k+1}^{m} s_j. \]
Rearranging gives
\[ (m-k) \sum_{j=1}^{k} r_j \geq k \sum_{j=k+1}^{m} s_j. \]
Since both \( \phi, \psi \) are unit vectors, it follows that
\[ k \sum_{j=1}^{k} e_j = m \sum_{j=k+1}^{m} s_j. \]
Therefore
\[ (m-k) \sum_{j=1}^{k} (r_j + e_j) \geq m \sum_{j=k+1}^{m} s_j. \]
Note that the right side is precisely \( m\rho(\alpha, \psi) \). Hence
\[ \rho(\alpha, \psi) \leq \frac{m-k}{m} \sum_{j=1}^{k} (r_j + e_j) \leq \frac{k(m-k)}{m} \phi_{\max} \leq \frac{m \phi_{\max}}{4}, \] (10)
which follows from the obvious observation that \( k(m-k) \leq m^2/4 \) no matter what \( k \) is.

It is quite easy to show that the performance of the algorithm is bounded by \( 0.5m\phi_{\max} \). This is because no bin can be overstuffed by more than \( \phi_{\max} \), and no bin can have unutilized capacity more than \( \phi_{\max} \). Since the totals of over- and under-capacity have to balance out, the bound \( 0.5m\phi_{\max} \) follows. Thus the real essence of the theorem is to gain an extra factor of 0.5.

The specific result of [18] bounds the total weight of all the bins (call it \( A \)) and shows that \( A_{LS} \leq 1.25A_{opt} \), where LS is the online list-scheduling algorithm. Moreover, they also require an extra assumption that \( \phi_{\max} \leq \psi_{\min} \), something that is not needed here. It is easy to verify that the weight of an algorithm equals \( 1 + \rho(\phi(a), \psi) \) achieved by that algorithm. Hence a direct application of the results of [18] would imply that
\[ \rho_{LS}(\phi(a), \psi) \leq 1.25\rho_{opt}(\phi(a), \psi) + 0.25. \]
Because of the additive constant of 0.25, this bound is less useful than the bound (9) given by Theorem 3.

The above analysis works also when the bins are nonuniform in size. Moreover, as pointed out in [18], the LS (in this case BF) algorithm actually works better when the bin sizes are widely disparate. However, the problem of optimal aggregation to maximize entropy is a conventional bin-packing problem with equal bin sizes of \( 1/m \) and overstuffing permitted. The bound given in Theorem 3 holds in this case as well.

VI. EXAMPLE OF AGGREGATION USING THE GREEDY ALGORITHM

Example 1: To illustrate the above algorithm, we solve a 40 \( \times \) 10 problem. First two uniformly distributed random vectors \( x \in [0, 1]^{40}, y \in [0, 1]^{10} \) were generated using the rand command of MATLAB. Then these were stretched out via the transformation
\[ \phi_i = \exp(x_i)/s_1, \quad \psi_j = \exp(y_j)/s_2, \]
where \( s_1, s_2 \) are scaling constants to make the sums come out equal to one. Then only the smaller vector is sorted in descending order. The results are shown below. For display purposes the resulting \( \phi \) and \( \psi \) are shown as a matrices, though in reality both are row vectors.

\[
\phi = \begin{bmatrix}
0.0304 & 0.0333 & 0.0153 & 0.0335 & 0.0253 \\
0.0148 & 0.0178 & 0.0232 & 0.0350 & 0.0353 \\
0.0157 & 0.0355 & 0.0350 & 0.0219 & 0.0299 \\
0.0155 & 0.0205 & 0.0336 & 0.0297 & 0.0351 \\
0.0259 & 0.0139 & 0.0314 & 0.0342 & 0.0265 \\
0.0287 & 0.0283 & 0.0199 & 0.0259 & 0.0160 \\
0.0273 & 0.0139 & 0.0177 & 0.0141 & 0.0148 \\
0.0307 & 0.0270 & 0.0185 & 0.0348 & 0.0139 \\
\end{bmatrix},
\]

\[
\psi = \begin{bmatrix}
0.1241 & 0.1205 & 0.1192 & 0.1139 & 0.1069 \\
0.0914 & 0.0875 & 0.0869 & 0.0821 & 0.0675 \\
\end{bmatrix}.
\]
Applying the best fit algorithm for aggregation \textit{without} sorting \( \phi \) results in the following grouping and aggregation (shown as a matrix for convenience):

\[
I_1 = \{1, 16, 23, 33\}, I_2 = \{2, 18, 31\}, I_3 = \{3, 12, 24, 40\},
I_4 = \{4, 19, 30, 36\}, I_5 = \{5, 15, 27, 38\}, I_6 = \{6, 11, 17, 25, 34\},
I_7 = \{7, 13, 26, 37\}, I_8 = \{8, 14, 22, 28, 35\},
I_9 = \{9, 20, 32, 39\}, I_{10} = \{10, 21, 29\},
\]

\[
\phi^{(a)} = \begin{bmatrix}
0.0951 & 0.0942 & 0.0989 & 0.1099 & 0.1020 \\
0.0917 & 0.1085 & 0.0938 & 0.1188 & 0.0871
\end{bmatrix}.
\]

We have that \( H(\phi^{(a)}) = 2.2934 \), quite close to the theoretical maximum of 2.3026.

In contrast, if we first sort \( \phi \) before applying the best fit algorithm, the following grouping results:

\[
I_1 = \{1, 20, 21, 39\}, I_2 = \{2, 19, 22, 40\}, I_3 = \{3, 18, 23, 38\},
I_4 = \{4, 17, 24, 37\}, I_5 = \{5, 16, 25, 36\}, I_6 = \{6, 15, 29, 31\},
I_7 = \{7, 14, 27, 34\}, I_8 = \{8, 13, 26, 35\},
I_9 = \{9, 12, 28, 33\}, I_{10} = \{10, 11, 30, 32\}.
\]

The resulting aggregation is

\[
\phi^{(a)} = \begin{bmatrix}
0.1019 & 0.1021 & 0.1016 & 0.1007 & 0.1004 \\
0.0982 & 0.0994 & 0.0993 & 0.0982 & 0.0982
\end{bmatrix},
\]

which is much closer to being uniform than the earlier aggregation.

\section{Conclusions}

In this paper we have studied the problem of finding an optimal lower-order approximation to a higher-order probability distribution, where the metric distance between the original and reduced-order distributions is the variation of information metric introduced in a companion paper [16]. It is first shown that every optimal reduced-order approximation must in fact be an \textit{aggregation} of the original distribution. Thus the optimal order reduction problem is shown to be equivalent to finding an aggregation that has maximum entropy. Since this problem is NP-hard, we have reformulated it as a problem of bin-packing with over-stuffing, which is also NP-hard. However, for the latter problem we are able to give a greedy algorithm and also to prove an upper bound on its performance. The approach has been illustrated by a fairly large example of finding a 10-th order approximation to a 40-th order distribution.

Note that a preprint that combines both [16] as well as the present paper can be found at [15].