Caching of Bivariate Gaussians with Non-Uniform Preference Probabilities

Giel J. Op 't Veld Michael C. Gastpar
School of Computer and Communication Sciences, EPFL
Lausanne, Switzerland
giel.optveld@epfl.ch michael.gastpar@epfl.ch

Abstract

Caching is technique that alleviates networks during peak hours by transmitting partial information before a request for any is made. We study this method in a lossy source coding setting with Gaussian databases. A good caching strategy minimizes the data still needed on average once the user requests a file. We identify two important parameters: the prior preference for a file and the correlation among files. This paper characterizes the trade-off between cache and average update communication rate to meet a user's demand using Gaussian codebooks. It is argued that what information needs to be cached not only depends on preference and correlation, but also on the size of the cache.

1 Introduction

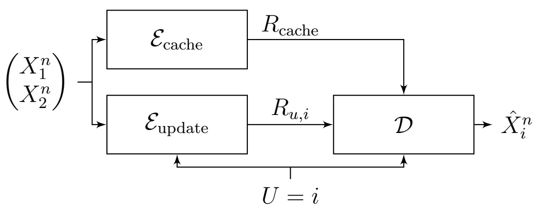
'On-demand' is key in the current state of communication technology. In this millennium, downloading videos of one's own choice replaced (or at least offered an alternative to) traditional broadcast of TV and radio. Then, instantaneous streaming replaced downloading. While providing great flexibility for the user, this demand for personal and instantaneous data streams also greatly increased the load on the network.

A second cost is often overlooked: on-demand services also increase the *imbalance* of network load. Notoriously data-heavy applications like Netflix and Amazon Prime are hardly popular during the day; almost all users use the service sometime between dinner and their bedtime. Network and server capacity suffer from this imbalance; they are installed to withstand *peak* traffic, not average.

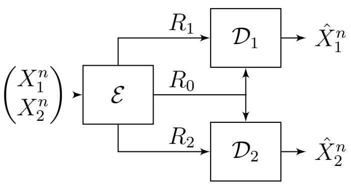
A challenge for IT is to combine the user experience of on-demand streaming with a balanced network load; caching can be the tool to break that impasse. The key of caching is that a server does not wait for a user to make a request for data. Instead the server tries to *anticipate* what data will be requested and sends it in advance. Netflix, for example, could transmit parts of the next episode of your favorite series at night, assuming that you will continue your viewing habits tomorrow. Imperfect prediction of the user's request will increase the overall need for data, but a well designed system can reduce the average network load during peak hours. This is the trade-off we study.

We model this caching problem in a way that resembles the Gray–Wyner network [1]. The same model was introduced in a lossless discrete setting before [2] and was extended to lossy in [3, 4] Also others studied the lossy setting, but took a worst-case metric [5]. This paper has its focus on Gaussian sources. An effective caching strategy weighs two parameters: the amount of correlation between the elements of the database, and the user's preference for one of them. The case of correlation was presented before, but not in combination with preference [3].

The end goal is twofold: To obtain a plot of the optimal rate trade-off between caching and updating, and to understand the coding strategy that achieves it. The latter question shows the most interesting insight: what is the best caching strategy relies heavily on knowledge of the user's preference, but surprisingly also on the size of the cache. As cache size grows larger, the encoder should care less about correlation and more about preference.



(a) The Caching set-up.



(b) The Gray–Wyner network.

Figure 1: Caching is equivalent to the Gray–Wyner network by drawing the update events for X_1 and X_2 as two separate links.

2 Problem Statement

2.1 Definitions

We study the network in Figure 1a in the classical length-N block coding sense. We model caching with a database that does not consist of 'Netflix series', but equally exciting Gaussian random variables. The i.i.d. sequence of vectors $\mathbf{X}^n$ is the database, whereas X_1^n and X_2^n refer to its 'files'. $\mathbf{X}$ is Gaussian distributed $\sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{X}})$, with correlation ρ and both X_i having unit variance. The decoder, though, only needs a lossy description of one of the X_i. Which one is announced in the user's request, modeled as side information U that is a simple Bernoulli random variable with $p(U = 1) = p$.

The first encoder, the *cache*, has no knowledge of the user's request and produces a messages using NR_{cache} bits. The second encoder, the *update*, learns the request $U = i \in \{1, 2\}$ and proceeds to code a message using $NR_{u,i}$ bits. The decoder should be able to combine the cache and update messages to reconstruct $\hat{X}_i$ at the desired final distortion level D_{final}. A rate-distortion quadruple $(R_{\text{cache}}, R_{u,1}, R_{u,2}, D_{\text{final}})$ is said to be achievable if there exists such encoders and a decoder that for both $i = 1, 2$

$$\limsup_{N\to\infty} \mathbb{E}\Big(\frac{1}{N}\sum_{n=1}^{N}\Big(X_i(n) - \hat{X}_i(n)\Big)^2\Big) \leq D_{\text{final}}$$

The main question we pose is: What does one need to cache in order to minimize the update rate that is needed on average? To that end define the *average* update rate:

$$\overline{R}_{\text{update}} = pR_{u,1} + (1-p)R_{u,2}.$$

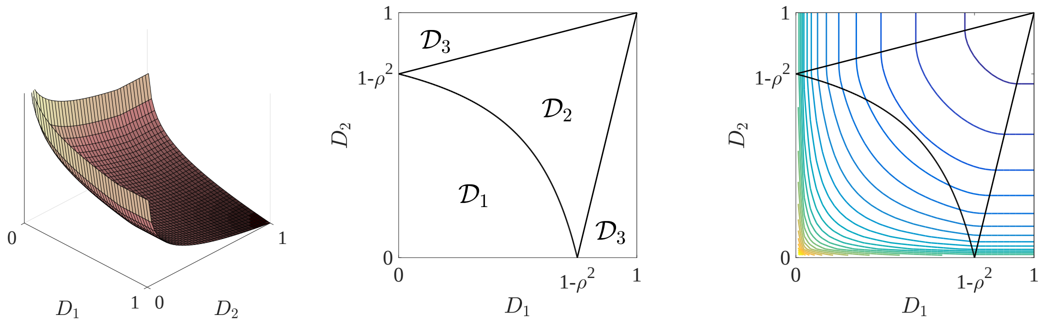
We say $(R_{\text{cache}}, \overline{R}_{\text{update}}, D_{\text{final}})$ is achievable if there is at least one achievable tuple $(R_{\text{cache}}, R_{u,1}, R_{u,2}, D_{\text{final}})$ whose average update rate is exactly $\overline{R}_{\text{update}}$.

An important tool will be the rate-distortion function for Gaussian sources:

$$R(D) = \frac{1}{2}\log\frac{1}{D},$$

but also the more important joint rate-distortion function of two Gaussians under *individual* mean squared error distortion constraints [6]:

$$R(D_1, D_2) = \begin{cases} \frac{1}{2}\log\left(\frac{1-\rho^2}{D_1 D_2}\right) & \text{if } (D_1, D_2) \in \mathcal{D}_1, \\ \frac{1}{2}\log\left(\frac{1-\rho^2}{D_1 D_2 - \left(\rho - \sqrt{(1-D_1)(1-D_2)}\right)^2}\right) & \text{if } (D_1, D_2) \in \mathcal{D}_2, \\ \frac{1}{2}\log\left(\frac{1}{\min(D_1, D_2)}\right) & \text{if } (D_1, D_2) \in \mathcal{D}_3, \end{cases}$$



(a) $R(D_1, D_2)$. (b) The distortion plane as defined by (1)-(3). (c) The distortion plane with contour lines of $R(D_1, D_2)$.

Figure 2: Visualization of the Gaussian joint rate-distortion function $R(D_1, D_2)$.

where different combinations (D_1, D_2) are grouped into subsets of $\mathcal{D} = [0, 1] \times [0, 1]$:

$$\mathcal{D}_1 = \{D_1, D_2 : \ (1 - D_1)(1 - D_2) \geq \rho^2\}, \tag{1}$$

$$\mathcal{D}_2 = \{D_1, D_2 : \ (1 - D_1)(1 - D_2) \leq \rho^2 \leq \min\left(\frac{1 - D_1}{1 - D_2}, \frac{1 - D_2}{1 - D_1}\right)\}, \tag{2}$$

$$\mathcal{D}_3 = \mathcal{D}_1^c \cap \mathcal{D}_2^c. \tag{3}$$

Figure 2 makes $R(D_1, D_2)$ tangible through visualization. The plane of distortion levels for $\hat{X}_1$ and $\hat{X}_2$ is cut into different regions in which the rate-distortion function exhibits different behavior. At each coordinate, there will be a 2×2 error matrix

$$\mathbf{D} = \mathbb{E}[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]. \tag{4}$$

In $\mathcal{D}_1$ this matrix is diagonal, whereas in $\mathcal{D}_2$ it is correlated. $\mathcal{D}_3$ is degenerate: the distortion D_i on one X_i is so small (in comparison to the other) that the best strategy is to only code that X_i. One can then achieve any distortion in $\mathcal{D}_3$ on the other component by an estimator. We call Figure 2b in its entirety the $\mathcal{D}-$plane.

2.2 Caching as an application of the Gray–Wyner Network

From an operational perspective and the existence of codes, there is a complete equivalence with the Gray–Wyner network [1], as depicted in Figure 1b. Namely, even though only X_1 or X_2 needs to be transmitted, any code should be capable of doing both as the user could request either. Comparing Figures 1a and 1b the equivalence is as follows:

$$R_0 \leftrightarrow R_{\text{cache}}, \quad R_1 \leftrightarrow R_{u,1}, \quad R_2 \leftrightarrow R_{u,2}.$$

The Gray–Wyner network was introduced in [1], but also featured more recently in an explicit lossy source coding setting similar to ours [7, 8] The region of achievable rate-distortion tuples on the Gray–Wyner network is the union over joint densities $p(\mathbf{X}, V, \hat{\mathbf{X}})$ of all $(R_0, R_1, R_2, D_1, D_2)$ satisfying

$$\begin{aligned} R_0 &\geq I(\mathbf{X}; V) \\ R_i &\geq I(X_i; \hat{X}_i | V) \qquad \text{for } i = 1, 2 \\ D_i &\geq \mathbb{E}[d_{X_i}(X_i, \hat{X}_i)] \quad \text{for } i = 1, 2 \end{aligned}$$

for some distortion measure $d_X(\cdot, \cdot)$ (in our case squared error). Hence, by this equivalence one knows which caching strategies are achievable:

Theorem 1. *The caching rate-distortion region is the union over all joint densities* $p(\mathbf{X}, V, \hat{\mathbf{X}})$ *of tuples* $(R_{cache}, \overline{R}_{update}, D_{final})$ *satisfying the following inequalities*

$$\begin{aligned} R_{cache} &\geq I(\mathbf{X}; V) \\ \overline{R}_{update} &\geq pI(X_1; \hat{X}_1|V) + (1-p)I(X_2; \hat{X}_2|V) \\ D_{final} &\geq \mathbb{E}[d_{X_i}(X_i, \hat{X}_i)] \qquad\qquad \text{for } i = 1, 2 \end{aligned}$$

The rest of the paper is dedicated to understanding the boundary of this $(R_{\text{cache}}, \overline{R}_{\text{update}})$ trade-off, to understand which strategies are not only achievable, but are also *good.*

2.3 The Gaussian Case

One major difficulty is that the source $\mathbf{X}$ being Gaussian does not imply that on the boundary of the achievable region V and $\hat{\mathbf{X}}$ are necessarily jointly Gaussian as well. Whenever it holds that $R_{\text{cache}} + R_{u,1} + R_{u,2} = R(D_{\text{final}}, D_{\text{final}})$, Gaussian auxiliaries are sufficient for optimality. This requires R_{cache} to be large. For small R_{cache}, one necessarily has $R_{\text{cache}} + R_{u,1} + R_{u,2} > R(D_{\text{final}}, D_{\text{final}})^*$. This discussion is, however, beyond the scope of this paper and we will further restrict ourselves to all variables being Gaussian. From here onward, we therefore speak of the *Gaussian* achievable caching rate-distortion region.

Corollary 1. *The boundary of the Gaussian caching rate-distortion region is characterized by*

$$R_{cache}(d, D_{final}) = \min_{D_{final} \leq D_1, D_2 \leq 1} R(D_1, D_2) \quad s.t.\ D_1^p D_2^{1-p} = d. \tag{5}$$

Defined for $d \in [D_{final}, 1]$*; picking* $d = D_{final} 2^{2\overline{R}_{update}}$ *relates the above back to* $\overline{R}_{update}$.

Proof. To characterize the boundary, we fix one rate and minimize the other like

$$R_{\text{cache}}(\gamma) = \min\ R_{\text{cache}} \quad \text{s.t.} \quad \overline{R}_{\text{update}} = \gamma.$$

For Gaussian random variables, the bounds of Theorem 1 are characterized by:

$$R_{\text{cache}} = \frac{1}{2} \log \frac{|\Sigma_{\mathbf{X}}|}{|\mathbf{D}|}, \tag{6}$$

$$\overline{R}_{\text{update}} = \frac{p}{2} \log \frac{D_{1,1}}{D_{\text{final}}} + \frac{1-p}{2} \log \frac{D_{2,2}}{D_{\text{final}}} = \frac{1}{2} \log \frac{D_{1,1}^p D_{2,2}^{1-p}}{D_{\text{final}}}. \tag{7}$$

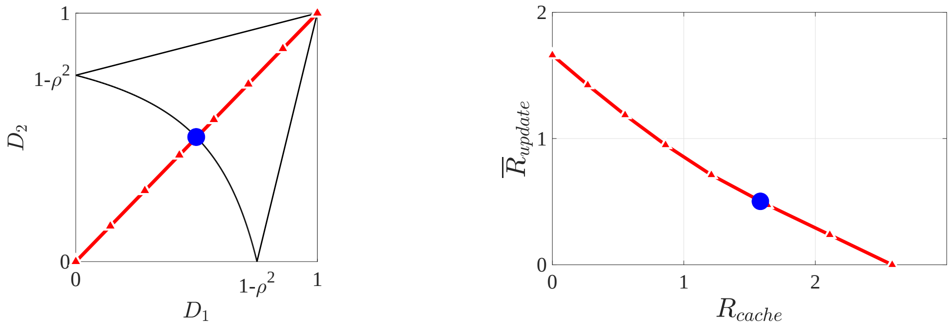
Any positive semidefinite matrix $\mathbf{D}$ that satisfies $\mathbf{D} \preceq \Sigma_{\mathbf{X}}$ can be uniquely associated to a random variable V that is jointly Gaussian with $\mathbf{X}$, and vice versa (see, e.g., [6]). $\mathbf{D}$ is the mean-squared error distortion (as in (4)) after caching, but before the update!

Observe that both rates are completely specified by this $\mathbf{D}$. However, $\overline{R}_{\text{update}}$ only depends on the diagonal entries. Hence, in the cache one can always pick a matrix $\mathbf{D}$ that is rate-distortion optimal w.r.t. $R(D_1, D_2)$ and then minimize over just these two scalar distortions D_1 and D_2.

As a matter of definition, instead of fixing $\overline{R}_{\text{update}}$ one can equivalently fix $D_1^p D_2^{1-p}$ to emphasize that the distortions up to which one caches the sources are truly the intrinsic variables of this problem: D_1, D_2 are both the objective *and* the constraint.

Lastly, $D_1, D_2 \geq D_{\text{final}}$ ensures that $R_{u,1}$ and $R_{u,2}$ in (7) are non-negative. In other words, caching either X_1 or X_2 beyond the end distortion constraint trivially serves no purpose; one should then instead cache the other source. ☐

*for more details on when exactly, please see [7, 4])



(a) The optimal caching strategy requires $D_1 = D_2$ at all time.

(b) An example of the consequent trade-off between R_{cache} and $\overline{R}_{\text{update}}$.

Figure 3: If $p = \frac{1}{2}$, optimal caching strategies must lie on the diagonal line in the $\mathcal{D}-$plane. After the blue dot, the best strategy moves from $\mathcal{D}_2$ into $\mathcal{D}_1$. For illustration purposes, the left is drawn with $\rho = 0.5$ and the right with $\rho = 0.8$.

When we use the term 'caching strategy', we refer to a choice of cache distortions (D_1, D_2). Such a strategy is said to be optimal if it is the solution to (5) for a particular value of $\overline{R}_{\text{update}}$. $R_{\text{cache}}(d, D_{\text{final}})$ is decreasing in d and its extreme ends are the following

$$\begin{aligned} R_{\text{cache}}(D_{\text{final}}, D_{\text{final}}) &\rightarrow (R_{\text{cache}}, \overline{R}_{\text{update}}) = (R(D_{\text{final}}, D_{\text{final}}), 0) && \textit{all cache}, \\ R_{\text{cache}}(1, D_{\text{final}}) &\rightarrow (R_{\text{cache}}, \overline{R}_{\text{update}}) = (0, R(D_{\text{final}})) && \textit{all update}. \end{aligned}$$

3 Caching with either Correlation or Preference

Let us first discuss a few extreme cases. One is obvious: if either p or $1-p$ equals 1, then there is no trade-off; the encoder should always code the desired X_i only. In that case, one can split the bits of $R(D_{\text{final}})$ over the code phases, or implements successive refinability of a single Gaussian [9]. Next, we review two other cases:

1. Preference, but no correlation: $\rho = 0$, but $p \neq 1-p$.
2. Correlation but no preference: $\rho \neq 0$, but $p = 1-p = \frac{1}{2}$.

In the first case, caching strategies are greedy:

Theorem 2. *If $\rho = 0$ the optimal caching strategy is to cache only the most popular X_i until it satisfies the end distortion constraint; then proceed with the other.*

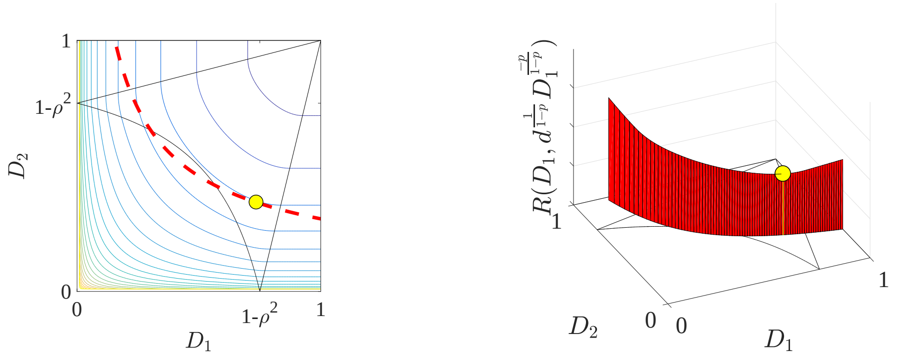
Due to space restrictions we omit the whole proof, but it stems from the fact that $X_1 \perp X_2$ implies $R(D_1, D_2) = R(D_1) + R(D_2)$ and consequently:

$$R_{\text{cache}}(d, D_{\text{final}}) = \min_{D_{\text{final}} \leq D_1, D_2 \leq 1} R(D_1) + R(D_2) \quad \text{s.t. } D_1^p D_2^{1-p} = d.$$

Second, symmetry in the preference probabilities means symmetry in the caching strategy, as we investigated earlier [3, 4]. An example is plotted in Figure 3.

Theorem 3 (Op 't Veld, Gastpar, [3, 4])**.** *When $p = 1-p = \frac{1}{2}$, the best caching strategy is to cache X_1 and X_2 equally:*

$$R_{\textit{cache}}(d, D_{\textit{final}}) = R(d, d).$$



(a) The optimal caching strategy is where a contour line of $R(D_1, D_2)$ is tangential to the line $D_1^p D_2^{1-p} = d$.

(b) A slice of $R(D_1, D_2)$ that is strictly convex in (D_1, D_2) has one unique minimum.

Figure 4: Example of Lemma 1 for $p = \frac{2}{5}$ and $d = 0.475$.

4 Caching with both Correlation and Preference

Consider now any $0 < \rho < 1$ and $p \neq \frac{1}{2}$. Above all, it should be clear that if $p < 1 - p$ then a good caching strategy should mostly cache information on X_2 and must thus result in $D_1 > D_2$. Let us therefore cut up the $\mathcal{D}-$plane in an upper and lower triangle:

$$\mathcal{D}_{i,1} = \mathcal{D}_i \cap \{D_1, D_2 :\ D_2 \geq D_1\} \quad \leftrightarrow \quad \mathcal{D}_{i,2} = \mathcal{D}_i \cap \{D_1, D_2 :\ D_2 \leq D_1\}\,.$$

To the best of our knowledge, there is no simple closed-form analytic expression for the optimal caching strategy (D_1, D_2) in terms of the parameters ρ, p and d. Numerically, however, the optimization of $R_{\text{cache}}(d, D_{\text{final}})$ is not hard as we lay out in two steps:

Lemma 1. *If $D_{final} = 0$, then the cache-update trade-off has one unique minimum on the $\mathcal{D}-$plane, which is the solution to*

$$R_{cache}(d, 0) = \min_{D_1}\ R(D_1, d^{\frac{1}{1-p}} D_1^{-\frac{p}{1-p}}).$$

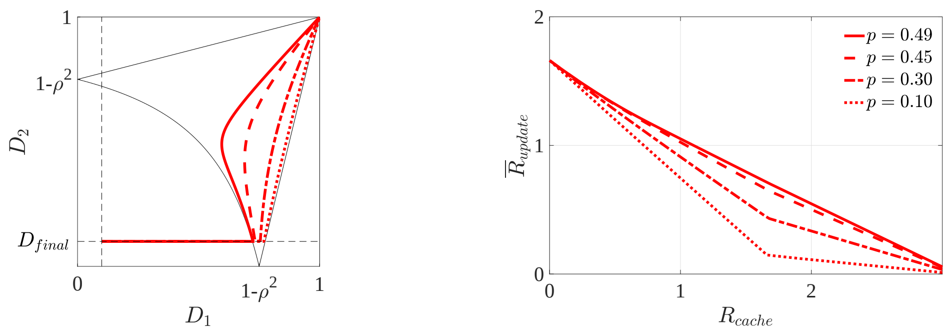
Proof. Neglect the constraint involving D_{final}. In (5) one does not evaluate $R(D_1, D_2)$ over *all* $(D_1, D_2) \in \mathcal{D}$, but only along a 'slice' defined by the constraint $D_1^p D_2^{1-p} = d$:

$$D_2 = d^{\frac{1}{1-p}} D_1^{-\frac{p}{1-p}}. \tag{8}$$

This slice is strictly convex with respect to (D_1, D_2). The contour lines (or isolines) of $R(D_1, D_2)$ are also convex (and continuous!) on the $\mathcal{D}-$plane. More importantly, though, these contour lines end straight ($\frac{dD_2}{dD_1} = 0$ in $\mathcal{D}_{3,2}$ and $\frac{dD_1}{dD_2} = 0$ in $\mathcal{D}_{3,1}$). Consequently, the minimum of $R(D_1, D_2)$ on a *strictly* convex curve is where that curve is tangential with a contour line; it cannot be at a simple crossing.

This tangential part is either a unique point or a (set of) closed interval(s), the latter if and only if $\exists$ a contour line that is described by the same curve as (8) on some interval(s). This happens in $\mathcal{D}_1$ when $p = \frac{1}{2}$, but is not the case for other values of p. Hence, there can be only one minimum. $\square$

An illustration of the 'slicing' of Lemma 1 is depicted in Figure 4.



(a) Optimal strategies move away from the diagonal as p decreases.

(b) The resulting trade-off between R_{cache} and $\overline{R}_{\text{update}}$.

Figure 5: Progression of optimal strategies and rates as one evaluates $R_{\text{cache}}(d, D_{\text{final}})$ for different d. Correlation is fixed to be $\rho = \frac{1}{2}$ and $p = 0.49, 0.45, 0.3$ and 0.1.

Theorem 4. *Without loss of generality, assume $p < \frac{1}{2}$. Then,*

$$R_{cache}(d, D_{final}) = R(\bar{D}_1, d^{\frac{1}{1-p}} \bar{D}_1^{-\frac{p}{1-p}})$$

where $\bar{D}_1 = \max(D_{final}, D_1^*)$ and D_1^* is the solution to

$$\frac{d}{dD_1}\left(-1 + D_1 + d^{\frac{1}{1-p}} D_1^{\frac{-p}{1-p}} + 2\rho\sqrt{(1-D_1)(1-d^{\frac{1}{1-p}} D_1^{\frac{-p}{1-p}})}\right) = 0, \tag{9}$$

over $(D_1, d^{\frac{1}{1-p}} D_1^{-\frac{p}{1-p}}) \in \mathcal{D}_{2,2}$.

Proof. First, assume D_{final} plays no restricting role. Then, the minimum of Lemma 1 lies necessarily in $D_{2,2}$. Namely, it cannot lie in $\mathcal{D}_{3,2}$ since its boundary is strictly superior. Second, in $\mathcal{D}_{1,2}$ the equipotential lines of $R(D_1, D_2)$ behave as $D_1 D_2 = \text{constant}$. Hence, they cannot be tangential to the curve $D_1^p D_2^{1-p} = d$, whose derivative is 'less steep' everywhere. That leaves $\mathcal{D}_{2,2}$, where $R(D_1, D_2)$ is minimized by maximizing $D_1 D_2 - \left(\rho - \sqrt{(1-D_1)(1-D_2)}\right)^2$. By restricting $(D_1, D_2) = (D_1, d^{\frac{1}{1-p}} D_1^{-\frac{p}{1-p}}) \in \mathcal{D}_{2,2}$ first and only then setting this derivative w.r.t. D_1 to 0, one finds the optimum.

Finally, should any D_i drop to D_{final}, then $R_{\text{cache}}(d, D_{\text{final}})$ is minimized at the intersection of $D_i = D_{\text{final}}$ and $D_1^p D_2^{1-p} = d$, since $R(D_1, D_2)$ is monotonic along that curve on one side of the aforementioned unconstrained minimum. □

5 Simulations and Discussion

We include two plots of the evolution of optimal caching strategies (Figure 5a) and the resulting trade-off in $(R_{\text{cache}}, \overline{R}_{\text{update}})$ (Figure 5b). The following points match:

$$\begin{aligned} (D_1, D_2) = (1, 1) \quad &\Leftrightarrow \quad (R_{\text{cache}}, \overline{R}_{\text{update}}) = (0, R(D_{\text{final}})) \\ (D_1, D_2) = (D_{\text{final}}, D_{\text{final}}) \quad &\Leftrightarrow \quad (R_{\text{cache}}, \overline{R}_{\text{update}}) = (R(D_{\text{final}}, D_{\text{final}}), 0) \end{aligned}$$

Tracing a curve in Figure 5a from the top-right corner to $(D_{\text{final}}, D_{\text{final}})$ corresponds to tracing Figure 5b from left to right. The differently patterned lines show how strategies change as p changes. As p moves from $\frac{1}{2} \to 0, 1$ the optimal caching distortion pairs move from the diagonal (implying the caching of an equal mixture of X_1 and X_2) towards the border of $\mathcal{D}_2$ and $\mathcal{D}_3$ (implying to cache only the most popular X_i).

Corollary 2. *The optimal caching strategy depends on the size of the cache.*

Every point in $\mathcal{D}_2$ is obtained by encoding a single Gaussian random variable that is a mixture of the two sources [6]: $\alpha X_1 + \beta X_2 + W$, with α, β some constants and W independent Gaussian noise. Had the optimal caching strategies lain on a straight line, then α and β would have been constant regardless of R_{cache}. Since this is not the case in general, these mixing coefficients change as R_{cache} changes.

Corollary 3. *Caching is* not *successively refinable unless* $p \in \{0, \frac{1}{2}, 1\}$ *or* $\rho = 0$.

Successive refinability [9] from (D_1, D_2) to (D_1', D_2') requires these coordinates to lie on a straight line originating from $(1, 1)$. Instead, optimal strategies lie on a curve (unless they are pushed to the border of $\mathcal{D}_{2,i}$ which are the exceptions mentioned). Hence, the encoder cannot spontaneously decide to cache more without losing efficiency.

Corollary 4. *For* $D_{\mathit{final}} \to 0$ *and* $R_{\mathit{cache}} \to \infty$*, the optimal strategy would be to only cache the most popular* X_i.

$\mathcal{D}_2$ ends in two corners, i.e., $(0, 1-\rho^2)$ or $(1-\rho^2, 0)$. if R_{cache} grows very large and D_{final} plays no restricting role, the optimal caching strategy is necessarily squeezed into one of these corners. These points are associated to a perfect description of one X_i and the resulting MSE-estimator of the other. In other words: for very large R_{cache} the best caching strategy cares more about the most popular component and less about the correlation between the two, irrespective of the value of p.

References

[1] R. Gray and A. Wyner, "Source coding for a simple network," *Bell System Technical Journal, The*, vol. 53, no. 9, pp. 1681–1721, Nov 1974.

[2] C. Y. Wang, S. H. Lim, and M. Gastpar, "Information-theoretic caching: Sequential coding for computing," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6393–6406, Nov 2016.

[3] G. J. Op 't Veld and M. C. Gastpar, "Caching (a pair of) gaussians," in *36th WIC Symposium on Information Theory in the Benelux*, May 2015, pp. 4–11.

[4] ——, "Caching gaussians: Minimizing total correlation on the gray-wyner network," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 478–483.

[5] R. Timo, S. S. Bidokhti, M. A. Wigger, and B. C. Geiger, "A rate-distortion approach to caching," *CoRR*, vol. abs/1610.07304, 2016. [Online]. Available: http://arxiv.org/abs/1610.07304

[6] J. Xiao and Q. Luo, "Compression of correlated gaussian sources under individual distortion criteria," in *43rd Allerton Conference on Communication, Control, and Computing*, 2005, pp. 438–447.

[7] K. Viswanatha, E. Akyol, and K. Rose, "The lossy common information of correlated sources," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3238–3253, June 2014.

[8] G. Xu, W. Liu, and B. Chen, "A lossy source coding interpretation of wyner's common information," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 754–768, Feb 2016.

[9] W. Equitz and T. Cover, "Successive refinement of information," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, 1991.