Group Membership Verification with Privacy: Sparse or Dense?
Marzieh Gheisari, Teddy Furon, Laurent Amsaleg

To cite this version:
Marzieh Gheisari, Teddy Furon, Laurent Amsaleg. Group Membership Verification with Privacy: Sparse or Dense?. 2019 IEEE International Workshop on Information Forensics and Security (WIFS), Dec 2019, Delft, Netherlands. hal-02489991

HAL Id: hal-02489991
https://hal.archives-ouvertes.fr/hal-02489991
Submitted on 24 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—Group membership verification checks if a biometric trait corresponds to one member of a group without revealing the identity of that member. Recent contributions provide privacy for group membership protocols through the joint use of two mechanisms: quantizing templates into discrete embeddings, and aggregating several templates into one group representation.

However, this scheme has one drawback: the data structure representing the group has a limited size and cannot recognize noisy query when many templates are aggregated. Moreover, the sparsity of the embeddings seemingly plays a crucial role on the performance verification.

This paper proposes a mathematical model for group membership verification allowing to reveal the impact of sparsity on both security, compactness, and verification performances. This models bridges the gap towards a Bloom filter robust to noisy queries. It shows that a dense solution is more competitive unless the queries are almost noiseless.

I. INTRODUCTION

Group membership verification is a procedure checking whether an item or an individual is a member of a group. If membership is positively established, then an access to some resources (a building, a file, . . . ) is granted; otherwise the access is refused. This paper focuses on privacy preserving group membership verification procedures where members must be distinguished from non-members, but where the members of a group should not be distinguished one another.

To this aim, a few recent contributions have proposed to rely on the aggregation and the embedding of several distinctive templates into a unique and compact high dimensional feature representing the members of a group [1], [2]. It has been demonstrated that this allows a good assessment of the membership property at test time. It has also been shown that this provides privacy and security. Privacy is enforced because it is impossible to infer from the aggregated feature which group membership protocols through the joint use of two mechanisms: quantizing templates into discrete embeddings, and aggregating several templates into one group representation.

This paper proposes a mathematical model for group membership verification allowing to reveal the impact of sparsity on both security, compactness, and verification performances. This models bridges the gap towards a Bloom filter robust to noisy queries. It shows that a dense solution is more competitive unless the queries are almost noiseless.

II. DISCRETE SEQUENCES

This section considers the problem of creating a representation $Y$ of a group of $n$ sequences $\{X_1, \ldots, X_n\}$, whose use is to test whether a query sequence $Q$ is a noisy version of one of these $n$ original sequences. This test is done at query time when the original sequences are no longer available and all that remains is the representation $Y$.

The sequences are elements of $\mathcal{X}^m$ where $\mathcal{X}$ is a finite alphabet of cardinality $|\mathcal{X}|$, say $\mathcal{X} := \{0, 1, \ldots, |\mathcal{X}| - 1\}$. The sequence follows a statistical model giving a central role to the symbol 0. The symbols of the sequences are independent and identically distributed with

$$\mathbb{P}(X = s) = \begin{cases} 1 - p(|\mathcal{X}| - 1) & \text{if } s = 0 \\ p & \text{otherwise} \end{cases} \quad (1)$$

for $p \in (0, 1/|\mathcal{X}|]$. Sparsity means that probability $p$ is small, density means that $p$ is close to $1/|\mathcal{X}|$ so that $X$ is uniformly distributed over $\mathcal{X}$. 

A. Structure of the group representation

We impose the following conditions on the aggregation $a(\cdot)$ computing the group representation $Y = a(X_1, \ldots, X_m)$:

- $Y$ is a discrete sequence of the same length $Y \in \mathcal{Y}$.
- Symbol $Y(i)$ only depends on symbols $\{X_1(i), \ldots, X_n(i)\}$.
- The same aggregation is made index-wise: with abuse of notation, $Y(i) = a(X_1(i), \ldots, X_n(i)), \forall i \in [m]$.
- $Y(i)$ does not depend on any ordering of the set $\{X_1(i), \ldots, X_n(i)\}$.

These requirements are well known in traitor tracing and group testing as they usually model the collusion attack or the test results over groups. Here, they simplify the analysis reducing the problem to a single letter formulation where index $i$ is dropped involving symbols $\{X_1, \ldots, X_n\}, Y$ and $Q$.

These conditions motivate a 2-stage construction. The first stage computes the type (a.k.a. histogram or tally) $T$ of the symbols $\{X_1, \ldots, X_n\}$. Denote by $T_{|X|, n}$ the set of possible type values. Its cardinality equals $|T_{|X|, n}| = \frac{(m+|X|-1)}{n}$ which might be too big. The second stage applies a surjective function $r : T_{|X|, n} \rightarrow \mathcal{Y}$, where $\mathcal{Y}$ is a much smaller set.

B. Noisy query

At enrollment time, the system receives $n$ sequences, aggregates them into the compact representation $Y$, and then forgets the $n$ sequences. At query time, the system receives a new sequence $Q$ conforming with one of the following hypotheses:

- $H_1$: $Q$ is a noisy version of one of the enrolled sequences.
- $H_0$: $Q = X_0 + N$, where $X_0$ shares the same statistical model but it is independent of $\{X_1, \ldots, X_n\}$.

We model the source of noise (due to different acquisition conditions) by a discrete communication channel. It is defined by the function $W : X \times X \rightarrow [0, 1]$ with $W(q|x) := \mathbb{P}(Q = q|X = x)$. We impose some symmetry w.r.t. the symbol 0: $W(0|0) = \eta_0$, $W(0|s) = \eta_s, \forall s \in X \setminus \{0\}$.

At query time, the system computes a score $S = s(Q, Y)$ and compares to a threshold: hypothesis $H_1$ is deemed true if $S \geq \tau$. This test leads to two formulations of error:

- $P_{tp}(n, m)$ is the probability of false positive: $P_{tp}(n, m) := \mathbb{P}(S \geq \tau|H_0)$.
- $P_{tn}(n, m)$ is the probability of false negative: $P_{tn}(n, m) := \mathbb{P}(S < \tau|H_1)$.

The emphasis on $(n, m)$ is natural. It is expected that: i) the more sequences are aggregated, the less reliable the test is, ii) the longer the sequences are, the more reliable the test is.

C. Figures of merit ($C, S, V$)

The section presents three information theoretic quantities (expressed in nats) measuring the performances of the scheme. The first two depends on the statistical model of $X$ (especially $p$) and the aggregation mechanism $a$. The last one depends moreover on the channel.

1) Compactness $C$: The compactness of the group representation is measured by the entropy $C := H(Y)$. It roughly means that the number of typical sequences $Y$ scales exponentially as $e^{nH(Y)}$, which can be theoretically compressed to the rate of $H(Y)$ nats per symbol.

2) Security $S$: We consider an insider aiming at disclosing one of the $n$ enrolled sequences. Observing the group representation $Y$, its uncertainty is measured by the equivalent $S := H(X|Y)$. This means that the insider does not know which of the $e^{nH(X|Y)}$ typical sequences the enrolled sequences are.

3) Verification $V$: In our application, the requirement of utmost importance is to have a very small probability of false positive. We are interested in an asymptotical setup where $m \rightarrow +\infty$. This motivates the use of the false positive error exponent as a figure of merit:

$$E_{fp}(n) := \lim_{m \rightarrow +\infty} -\frac{1}{m} \log P_{tp}(n, m). \quad (2)$$

If $E_{fp}(n) > 0$, it means that $P_{tp}(n, m)$ exponentially vanishes as $m$ becomes larger. The theory of test hypothesis shows that $E_{fp}(n)$ is upper bounded by the mutual information $V := I(Y; Q)$ where $Q$ is a symbol of the query sequence, i.e. a noisy version of $X_1$. It means that the necessary length for achieving the requirement $P_{tp}(n, m) < \epsilon$ is $[5]$

$$m \geq \frac{-\log \epsilon}{V}. \quad (3)$$

D. Noiseless setup

The bigger $V$ and $S$, the better the performance in terms of verifiability and security. Yet, they can not be both big at the same time. The noiseless case when the channel introduces no error and $Q = X$ simply illustrates the trade-off:

$$V \leq C \quad (4)$$

$$V + S = H(X) \quad (5)$$

with $H(X) = -\log p_0 + (1 - p_0) \log \frac{1 - p_0}{p_0}$ and $p_0 := \mathbb{P}(X = 0)$. For a given $|X|$, $H(X)$ is maximised by the dense solution: $H(X) \leq \log |X|$ with equality for $p = 1/|X|$.

III. Binary alphabet

This section explores the binary case where $X = \{0, 1\}$. We first set the surjection as the identity function s.t. $Y = T$. Then, the impact of the surjection is investigated.

A. Working with types

In the binary case, there are $n + 1$ type values. There can be uniquely labelled by the number of symbols ‘1’ in $\{X_1, \ldots, X_n\}$, i.e. $T = \sum_{i=1}^{n} X_i \sim B(n, p)$.

1) Verification: In the noiseless case, after some rewriting:

$$V = h(p) - \sum_{t=0}^{n} \mathbb{P}(T = t) h \left( \frac{t}{n} \right), \quad (6)$$

with $h(p) := -p \log p - (1-p) \log (1-p)$, the entropy of a Bernoulli r.v. $B(p)$. If $p = 1/2$ and $n$ is large:

$$V = \frac{1}{2n} + o \left( \frac{1}{n} \right). \quad (7)$$
This is not the maximum of this quantity. For large \( n \), the best option is to set
\[
p = \frac{\alpha}{n}, \quad \mathbb{V} = \frac{\beta}{n} + o \left( \frac{1}{n} \right),
\]
with \( \alpha = 1.338 \) and \( \beta = 0.580 \). This was proven in the totally different application of traitor tracing [6, Prop. 3.8].

This section outlines two setups: the dense setup where \( p = 1/2 \), and the sparse setup where \( p \) goes to 0 when more sequences are packed in the group representation. Both setups share the asymptotical property that \( \mathbb{V} \approx \kappa/n \) for large \( n \).

According to (3), we can pack a big number \( n \) of sequences into one group representation provided that their length \( m \) scales proportionally to \( n \).

2) Compactness: The figure of merit for compactness for types is just \( C = H(T) \) where \( T \) follows a binomial distribution: \( T \sim B(n, p) \). In the dense setup \( p = 1/2 \), the binomial distribution is approximated by a Gaussian distribution \( \mathcal{N}(n/2; n/4) \) providing:
\[
C = \frac{1}{2} \log \left( \frac{\pi en}{2} \right) + O \left( \frac{1}{n} \right).
\]
In the sparse setup \( p = \alpha/n \), the binomial distribution is approximated by a Poisson distribution \( \mathcal{P}(\alpha) \) [7]:
\[
C \approx \alpha(1 - \log(\alpha)) + e^{-\alpha} \sum_{j=0}^{+\infty} \frac{\alpha^j \log(j!)}{j!}.
\]
This shows that the types are not compact in the dense setup; it approximatively remains constant in the sparse setup.

3) Security: Thanks to (5), we only need to calculate \( H(X) = h(p) \). In the dense setup, \( H(X) = \log(2) \) and \( S \) converges to \( H(X) \) as \( n \) increases. Merging into a single representation protects an individual sequence. If sparse,
\[
H(X) = \frac{\alpha}{n} \left( 1 - \log \frac{\alpha}{n} \right) + o \left( \frac{1}{n} \right).
\]
Therefore, \( S \) converges to zero as \( n \) increases, contrary to the dense setup. It might be more insightful to see that the ratio of uncertainties before and after observing \( T \), i.e. \( H(X)/H(X|T) \), converges to 1 in both cases. Merging does provide some security but sparsity is more detrimental.

B. Adding a surjection

The motivation of the surjection onto a smaller set \( Y \) is to bound \( C \) as \( C \leq \log |Y| \), \forall n. The Markov chain \( Q \rightarrow X_1 \rightarrow T \rightarrow Y \) imposes that \( V \leq I(T; Q) \). The surjection thus provokes a loss in verification as depicted in Fig. 1.

App. A shows that for \(|Y| = 2\), this loss is minimized for:
\[
r(t) = \begin{cases} 
0 & \text{if } t < t_p \\
1 & \text{otherwise}
\end{cases}
\]
where \( t_p \) is a threshold depending on \( p \). In the dense setup, \( t_p = n/2 \) and the surjection corresponds to a majority vote collusion in traitor tracing (a threshold model in group testing). Hence, by [6, Prop. 3.4]:
\[
V = \frac{1}{n^2} + o \left( \frac{1}{n} \right).
\]
In the sparse setup \( t_p = 1 \) which corresponds to an ‘All-1’ attack in traitor tracing (a the perfect model in group testing). Then the best option is to set \( p = \log(2)/n \) and [6, Prop. 3.3]:
\[
V = \frac{\log(2)^2}{n} + o \left( \frac{1}{n} \right).
\]
From (3), the necessary length is \( m \geq -n \log(\epsilon)/(\log(2))^2 \).

The main property \( V \approx \kappa/n \) still holds but the surjection lowers \( \kappa \) from 0.5 to 0.32 (dense), from 0.84 to 0.48 (sparse). The sparse setup is still the best option w.r.t. \( V \).

C. Relationship with the Bloom filter

A Bloom filter is a well-known data structure \( Y \in \{0, 1\}^m \) designed for set membership, embedding items to be enrolled into \( Y \) thanks to \( k \) hash functions. Its probability of false negative is exactly 0, whereas the probability of false positive is not null. The number of hash functions minimizing \( P_{fp}(n, m) \) is \( k = \lfloor \log(2)m/n \rfloor \). Then, the necessary length to meet a required false positive level \( \epsilon \) is \( m \geq -n \log(\epsilon)/(\log(2))^2 \).

These numbers show the connection with our scheme (14). At the enrollment phase, the hash functions indeed associate to the \( j \)-th item a binary sequence \( X_j \), indicating which bits of \( Y \) have to be set. This sequence is indeed sparse with \( k/n \approx \log(2)/n \). The necessary length is the same. Indeed, the enrollment phase of a Bloom filter is nothing more than the ‘All-1’ surjection.

The only difference resides in the statistical model. There is at most \( k \) symbols ‘1’ in sequence \( X_j \) whereas, in our model, that follows a binomial distribution \( B(m, p) \). Yet, asymptotically as \( m \rightarrow \infty \), by some concentration phenomenon, the two models get similar. This explains why we end up with similar optimal parameters. Yet, the Bloom filter only works
when the query object is exactly one enrolled item, whereas
the next section shows that our scheme is robust to noise.

IV. REAL VECTORS

This section deals with real vectors: \( n \) vectors to be enrolled \( \{\vec{x}_1, \ldots, \vec{x}_n\} \subset \mathbb{R}^d \), and the query vector \( \vec{q} \in \mathbb{R}^d \). All have unit norm. An embedding mechanism \( \mathcal{E} : \mathbb{R}^d \to \mathcal{X}^m \) makes the connection with the previous section. As in [8], this study models the embedding as a probabilistic function.

A. Binary embedding

For instance, for \( \mathcal{X} = \{0, 1\} \), a popular embedding is:

\[
X(i) = [\vec{x}^T \vec{U}_i > \lambda_x], \forall i \in [m]
\]  

(15)

where \( \vec{U}_i \sim \mathcal{N}(\vec{0}, I_d) \). This in turn gives i.i.d. Bernoulli symbols \( \{X(i)\} \) with \( p = 1 - \Phi(\lambda_x) \) if \( ||\vec{x}|| = 1 \).

At the query time, the embedding mechanism uses the same random vectors but a different threshold:

\[
Q(i) = [\vec{q}^T \vec{U}_i > \lambda_q], \forall i \in [m].
\] 

(16)

Under \( \mathcal{H}_1 \), suppose that \( \vec{q}^T \vec{x}_i = c < 1 \). This correlation defines the channel \( X \to Q \) with the error rates:

\[
\eta_0 = \Pr(\vec{q}^T \vec{U} > \lambda_q | \vec{x}^T \vec{U} \leq \lambda_x),
\]

\[
\eta_1 = \Pr(\vec{q}^T \vec{U} \leq \lambda_q | \vec{x}^T \vec{U} > \lambda_x).
\] 

(17)

(18)

The error rate \( \eta_0 \) has the expression (and similarly for \( \eta_1 \)):

\[
\eta_0 = 1 - (1-p)\sqrt{2\pi} \int_{-\infty}^{\lambda_x} \Phi \left( \frac{\lambda_q - cx}{\sqrt{1-c^2}} \right) e^{-\frac{x^2}{2}} dx.
\] 

(19)

B. Induced channel

For this embedding, the parameters \( (\lambda_x, \lambda_q, c, d) \) for the vectors define the setup \( (p, \eta_0, \eta_1) \) for the sequences. It is a priori difficult to find the best tuning \( (\lambda_x, \lambda_q) \). For a fixed \( \lambda_x \), \( \eta_0 \) decreases with \( \lambda_q \), while \( \eta_1 \) increases. App B reveals that \( V \) is sensitive to \( \eta_0 \) especially with the ‘All-1’ surjection of the sparse solution. Fig. 2 shows indeed that the dense solution \( (\lambda_x, \lambda_q) = (0, 0) \) is more robust, unless \( c \) is very close to 1. Here, we enforce a surjection (identity, All-1, or majority vote) and make a grid search to find the optimum \( (\lambda_x, \lambda_q) \) for a given \( c \). It happens that these parameters are better set to 0, i.e. dense solution, for the identity and majority vote. As for the ‘All-1’ surjection, we observe that \( \lambda_x \) is s.t. \( p \approx 1/n \) and \( \lambda_q \) is slightly bigger than \( \lambda_x \) to lower \( \eta_0 \). Yet, this sparse solution is not as good as the dense solution unless \( c \) is close to 1, i.e. the query vector is very close to the enrolled vector.

This observation holds only for the embedding function (15). Hashing functions less prone to error \( \eta_0 \) may exist.

V. EXPERIMENTAL WORK

We evaluate our scheme with face recognition. Face images are coming from LFW [9], CFP [10] and FEI [11] databases. For each dataset, \( N \) individuals are enrolled into random groups. There is the same number \( N_q \) of positive and negative (impostors) queries.

Labeled Faces in the Wild: These are pictures of celebrities in all sort of viewpoint and under an uncontrolled environment. We use pre-aligned LFW images. The enrollment set consists of \( N = 1680 \) individuals with at least two images in the LFW database. One random template of each individual is enrolled in the system, playing the role of \( \vec{x}_i \). Some other \( N_q = 263 \) individuals were randomly picked in the database to play the role of impostors.

Celebrities in Frontal-Profile: These are frontal and profile views of celebrities taken in an uncontrolled environment. We only use \( N = 400 \) frontal images enrolled in the system. The impostor set is a random selection of \( N_q = 100 \) other individuals.

Faculdade de Engenharia Industrial: The FEI database contains images in frontal view in a controlled environment. We use pre-aligned images. There are \( 200 \) subjects with two frontal images (one with a neutral expression and the other with a smiling facial expression). The database is created by randomly sampling \( N = 150 \) individuals to be enrolled, and \( N_q = 50 \) impostors.

A. Experimental Setup

Face descriptors are obtained from a pre-trained network based on VGG-Face architecture followed by PCA [12]. FEI corresponds to the scenario of employees entering in a building with face recognition, whereas CFP is more difficult, and LFW even more difficult. To equalize the difficulty, we apply a dimension reduction (Probabilistic Principal Component Analysis [13]) to \( d = 128 \) (FEI), 256 (CFP), and 512 (LFW). The parameters of PPCA are learned on a different set of images, not on the enrolled templates and queries. The vectors are also \( L_2 \) normalized. With such post-processing, the average correlation between positive pairs equals 0.83 (FEI), 0.78 (CFP), and 0.68 (LFW) with a standard deviation of 0.01. Despite the dimension reduction, the hardest dataset is LFW and the easiest FEI.

In one simulation run, the enrollment phase makes random groups with the same number \( n \) of members. A user claims she/he belongs to group \( g \). This claim is true under hypothesis \( \mathcal{H}_1 \) and false under hypothesis \( \mathcal{H}_0 \) (i.e. the user is an impostor). Her/his template is quantized to the sequence \( Q \), and \( (Q, g) \) is sent to the system, which compares \( Q \) to the group repre-
Our theoretical study justifies that the dense setup is more interesting in terms of verification performance $V$ and security level 5 unless we are operating in the high-SNR regime where the positive queries are very well correlated with the enrolled templates. This statement holds for any embedding, yet some are certainly more suited than others depending on $d$, $c$, and the geometrical relationship among positive pairs.

**VI. Conclusion**

This work is supported by the project CHIST-ERA ID_IOT 20CH21 167534.
Appendix

Let us first explain how $V$ is computed. Denote $P_i(q, y) := \mathbb{P}(Q = q, Y = y | H_i)$ and channel $W(q|x) := \mathbb{P}(Q = q | X = x)$, $\forall y \in \mathcal{Y}, q \in \mathcal{X}$ and $i \in \{0, 1\}$. Then,

$$V = \sum_{q, y} P_1(q, y) \log \frac{P_1(q, y)}{P_0(q, y)},$$

with $P_0(q, y) = \mathbb{P}(Q = q)\mathbb{P}(Y = y)$ and

$$P_1(q, y) = \sum_{x \in \mathcal{X}} \mathbb{P}(Y = y, X = x)W(q|x).$$

A. Surjection to $\mathcal{Y} = \{0, 1\}$

We assume here the noiseless setup allowing to write $\mathbb{P}(Y = y, X = x) = P_1(x, y)$. Inspired by traitor tracing, we consider a probabilistic surjection where $r_i$ parametrizes the surjection. Denote by $\nabla_\theta V(t)$ the derivative w.r.t. $\theta_i$. After some lengthy calculus:

$$\nabla_\theta V(t) = n^{-1}K_1(p, \theta)(t-nK_2(p, \theta)),$$

$$K_1(p, \theta) = \mathbb{P}(T = t)\Delta,$$

$$K_2(p, \theta) = \frac{h'(P_1(0, 1)) - h'(\mathbb{P}(Y = 1))}{\Delta},$$

$$\Delta = h'(P_1(0, 1)) - h'(P_1(1, 1)).$$

It is not possible to cancel the gradient $\nabla_\theta V$. The optimal $\theta$ thus lies on the boundary of the hypercube $[0, 1]^{n+1}$. This makes the surjection deterministic. Assuming $\mathbb{P}(Y = 1|X = 0) < \mathbb{P}(Y = 1|X = 1)$, then $0 < K_1(p, \theta)$ and $0 < K_2(p, \theta) \leq 1$ because $h'(-)$ is strictly decreasing. This makes $\nabla_\theta V(0) < 0$ and $\theta_0$ must be set to the lowest possible value, i.e. $\theta_0 = 0$, to increase $V$ at most. This is indeed the case for any $\theta_i$ with $t < K_2(p, \theta)$. In the same way, $\theta_n = 1$ and so is $\theta_i$ if $t > K_2(p, \theta)$. Yet, for a given $\theta$, $K_2(p, \theta)$ ranges from 0 to 1 as $p$ increases from 0 to 1. Therefore, $\theta = (0, \ldots, 0, 1, \ldots, 1)$ is optimal only over an interval of $p$.

For $n$ odd and $p = 1/2$, $\theta_i = 0$ if $t \leq (n + 1)/2$, and 1 (i.e. majority vote) otherwise is optimal because $K_2(1/2, \theta) = 1/2$ ($\mathbb{P}(Y = 1) = 1/2$ and $P_1(0, 1) = 1 - P_1(1, 1)$).

The ‘All-1’ surjection: $\theta = (0, \ldots, 1)$ makes $P_1(1, 1) = 1$ so that $\nabla_\theta V(t) = +\infty$ if $t > 0$ and $< 0$ for $t = 0$.

B. Impact of the channel

Suppose that $\eta$ is a parameter of the channel $W(\cdot | \cdot)$. Then

$$\frac{\partial V}{\partial \eta} = \sum_{q, y} \frac{\partial P_1(q, y)}{\partial \eta} \log \frac{P_1(q, y)}{P_0(q, y)},$$

because $\sum_{q, y} \frac{\partial P_1(q, y)}{\partial \eta} = \frac{\partial \sum_{q, y} P_1(q, y)}{\partial \eta} = 0$ and $\sum_{q, y} P_1(q, y) \frac{\partial P_0(q, y)}{\partial \eta} = \sum_{q} \frac{\partial \mathbb{P}(Q = q)}{\partial \eta} = 0$.

Suppose now that $\eta = \eta_0 := W(q|0), \forall q \in \mathcal{X}\setminus\{0\}$. Then,

$$\frac{\partial P_1(q, y)}{\partial \eta_0} = \mathbb{P}(X = 0, Y = y) \forall q \in \mathcal{X}\setminus\{0\}.$$ (24)

Taking (23) around the noiseless channel where $\eta_0 = 0$ and $\mathbb{P}(X = 0, Y = y) = P_1(0, y)$ because $Q = X$:

$$\frac{\partial V}{\partial \eta_0} |_{\eta_0 = 0} = \sum_{q, y} P_1(0, y) \log \frac{P_1(x, y)}{P_0(x, y)} + \cdots$$

We only express the first terms to outline that if $P_1(x, y) = 0$ while $P_1(0, y)$ and hence $P_0(x, y)$ are not null, then this derivative goes to $-\infty$. A small deviation from the noiseless case with $\eta_0 \neq 0$ has a major detrimental impact on $V$. That situation happens for sure when working with type, i.e. $Y = T$: Consider the null type $t_0$ obtained when $X_1 = \ldots = X_n = 0$: $P_1(0, t_0) > 0$ while $P_1(x, t_0) = 0$, $\forall x \neq 0$.

One can prove that the surjection can mitigate this effect if $\exists t \neq t_0 : r(t) = r(t_0)$ and $P_1(0, t) > 0$. This happens with the majority vote of the dense setup, but unfortunately, not with of the ‘All-1’ surjection in the sparse setup.

References


