TOWARDS OPTIMUM COUNTERFORENSICS OF MULTIPLE SIGNIFICANT DIGITS USING MAJORISATION-MINIMISATION

Félix Balado and Guénolé C. Silvestre

School of Computer Science
University College Dublin, Ireland

ABSTRACT

Optimum counterforensics of the first significant digits entails a forger minimally modifying a forgery in such a way that its first significant digits follow some preselected authentic distribution, e.g., Benford's law. A solution to this problem based on the simplex algorithm was put forward by Comesaña and Pérez-González. However their approach requires scaling up the dimensionality of the original problem. As simplex has exponential worst-case complexity, simplex implementations can struggle to cope with medium to large scale problems. These computational issues get compounded by upscaling the problem dimensionality. Furthermore, Benford's law applies beyond the first significant digit, but no counterforensics method to date offers a solution to handle an arbitrary number of significant digits. As the use of simplex would only aggravate the computational issues in this case, we propose a more scalable approach to counterforensics of multiple significant digits informed by the Majorisation-Minimisation optimisation philosophy.

Index Terms— Counterforensics, multiple significant digits, Benford's law, Majorisation-Minimisation.

1. INTRODUCTION

The field of counterforensics (also called antiforensics) studies how to confuse digital forensic detection tests, whose goal is determining the authenticity of digital assets. In this paper we will first revisit and then extend the problem of optimum counterforensics of the first significant digit (FSD). The main motivation for this problem is Benford's law [1], which applies to the first significant digits of datasets from diverse sources. Most multimedia security research in this area has focused on the fact that Benford's law (or its stronger version) applies to the DCT coefficients of uncompressed images, but not to those of their JPEG-compressed counterparts. As this enables compression (or recompression) detection tests, one would like to remove all deviation of the distribution of the FSD from Benford's law while minimally altering the image quality. The problem of FSD counterforensics was first studied in [2], but without attempting optimality. Soon afterwards a second algorithm was proposed in [3], which did not perfectly enforce the target distribution either. This issue was solved in [4], but the distortion minimisation algorithm therein was heuristic. Finally, a completely general solution to the problem of optimum first-order counterforensics appeared in [5], and its authors showcased their approach by applying it to the problem of optimum FSD counterforensics. Here we present what we believe to be the first contribution towards a low complexity implementation of optimum counterforensics of multiple significant digits.

1.1. Notation and Preliminary Definitions

Boldface lowercase symbols are column vectors. The i-th element of vector $\mathbf{a}$ is a_i. Symbol $\mathbf{1}$ is the all-ones column vector, of length given by the context. Capital Greek letters denote matrices; the entry at row i and column j of A is $(\mathrm{A})_{i,j}$. $(\cdot)^t$ is the transpose operator. $\mathrm{vec}(\mathrm{A})$ is the vectorisation of A by stacking its columns. $\otimes$ is the Kronecker product. $\mathrm{diag}(\mathbf{a})$ is a diagonal matrix with $\mathbf{a}$ in its diagonal. I_n is the $n\times n$ identity matrix.. The 2-norm of $\mathbf{a}$ is $\|\mathbf{a}\| = \sqrt{\mathbf{a}^t\mathbf{a}}$. Calligraphic letters are sets, and $|\mathcal{V}|$ is the cardinality of set $\mathcal{V}$. The indicator function is defined as $\mathbb{1}_{\{\theta\}} = 1$ if logical expression θ is true, and zero otherwise.

Let $\mathcal{S}_n$ be the symmetric group, namely, the group of all permutations of $\{1, 2, \ldots, n\}$. We denote a permutation $\boldsymbol{\sigma} \in \mathcal{S}_n$ by means of a vector $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_n]^t$ where $\sigma_i \in \{1, 2, \ldots, n\}$ and $\sigma_i \neq \sigma_j$ for all $i \neq j$. This vector defines in turn a permutation matrix $\Pi_{\boldsymbol{\sigma}}$ with entries $(\Pi_{\boldsymbol{\sigma}})_{i,j} = \mathbb{1}_{\{\sigma_i=j\}}$. We will just write Π wherever a generic permutation matrix is considered. The reordering of an n-vector $\mathbf{x}$ using $\boldsymbol{\sigma}$ is the vector $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\,\mathbf{x}$, for which $y_i = x_{\sigma_i}$ for $i = 1, 2, \ldots, n$. We will call a *rearrangement* of $\mathbf{x}$ a unique reordering of its elements. The rearrangement of $\mathbf{x}$ in nondecreasing order is denoted by $\mathbf{x}^{\uparrow}$, with elements $x_1^{\uparrow} \leq x_2^{\uparrow} \leq \cdots \leq x_n^{\uparrow}$, and the rearrangement of $\mathbf{x}$ in nonincreasing order is denoted by $\mathbf{x}^{\downarrow}$.

2. OPTIMUM COUNTERFORENSICS OF THE FIRST SIGNIFICANT DIGIT

We will first describe the problem and establish nomenclature. Consider a *forgery* $\underline{\mathbf{z}} = [\underline{z}_1, \ldots, \underline{z}_n]^t \in \mathbb{R}^n$ such that $\underline{z}_i \neq 0$ for $i = 1, 2, \ldots, n$. The first significant digit of $\underline{z}_i$ is given by $z_i = \varphi(\underline{z}_i) \triangleq \lfloor |\underline{z}_i|\, 10^{-\lfloor \log_{10} |\underline{z}_i| \rfloor} \rfloor$, where $z_i \in \mathcal{V} = \{1, 2, \ldots, 9\}$. With a slight abuse of notation, $\mathbf{z} = \varphi(\underline{\mathbf{z}})$ is the vector containing all first significant digits of $\underline{\mathbf{z}}$. Let also $\gamma_i = \mathrm{sign}(\underline{z}_i)\, 10^{\lfloor \log_{10} |\underline{z}_i| \rfloor}$, i.e., 10 to the order of magnitude of the first significant digit of $\underline{z}_i$, multiplied by its sign. The forger wishes to modify the forgery $\underline{\mathbf{z}}$ to produce a *post-processed forgery* $\underline{\mathbf{y}} \in \mathbb{R}^n$ whose first significant digits exactly follow some target empirical distribution (histogram), while simultaneously minimising $\|\underline{\mathbf{z}} - \underline{\mathbf{y}}\|^2$. Assume that the target histogram is denoted by $\mathbf{h}^{\mathbf{x}} = [h_1^{\mathbf{x}}, h_2^{\mathbf{x}}, \ldots, h_q^{\mathbf{x}}]^t$, where $q = |\mathcal{V}|$. The bins corresponding to this histogram are given by a vector $\mathbf{v} = [v_1, v_2, \ldots, v_q]^t$ whose components $v_1 < v_2 < \cdots < v_q$ are the elements of $\mathcal{V}$. This histogram must fulfill $\mathbf{1}^t\mathbf{h}^{\mathbf{x}} = n$, so that it corresponds to an n-dimensional vector. The constraint on the distribution of the post-processed forgery can be stated as $\mathbf{h}^{\mathbf{y}} = \mathbf{h}^{\mathbf{x}}$, where the histogram of $\mathbf{y} = \varphi(\underline{\mathbf{y}})$ is computed as $h_k^{\mathbf{y}} = \sum_{i=1}^{n} \mathbb{1}_{\{y_i=v_k\}}$, for $k = 1, 2, \ldots, q$. Without loss of generality, we denote by $\mathbf{x}$ an

arbitrary vector possessing the target histogram $\mathbf{h}^{\mathbf{x}}$, for instance

$$\mathbf{x} = \mathbf{x}^{\uparrow} = [\underbrace{v_1, \dots v_1}_{h_1^{\mathbf{x}}}, \underbrace{v_2 \dots, v_2}_{h_2^{\mathbf{x}}}, \dots, \underbrace{v_q \dots, v_q}_{h_q^{\mathbf{x}}}]^t, \tag{1}$$

Instead of starting with $\mathbf{h}^{\mathbf{x}}$, the forger can pick some authentic signal $\underline{\mathbf{x}}$ and then use the histogram of its first significant digits for post-processing its forgery. We will call both $\underline{\mathbf{x}}$ and $\mathbf{x} = \varphi(\underline{\mathbf{x}})$ *decoys*, whether they correspond to authentic signals or whether they are synthetic as in (1). In any case, the constraint $\mathbf{h}^{\mathbf{y}} = \mathbf{h}^{\mathbf{x}}$ is completely equivalent to saying that $\mathbf{y}$ is a rearrangement of $\mathbf{x}$. The number of rearrangements of $\mathbf{x}$ is given by the multinomial coefficient $\binom{n}{\mathbf{h}^{\mathbf{x}}} = n!/(h_1^{\mathbf{x}}! \cdots h_q^{\mathbf{x}}!)$, and we will denote by $\mathcal{S}_{\mathbf{x}} \subset \mathcal{S}_n$ any set of permutations leading to all rearrangements of $\mathbf{x}$. Continuing with our discussion, it must hold that $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$, with $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$. Therefore the problem of finding an optimum post-processed forgery $\underline{\mathbf{y}}^*$ with the desired FSD distribution can be expressed as

$$\underline{\mathbf{y}}^* = \underset{\substack{\underline{\mathbf{y}} \in \mathbb{R}^n \\ \mathbf{y} = \Pi\mathbf{x}}}{\arg\min} \|\underline{\mathbf{z}} - \underline{\mathbf{y}}\|^2. \tag{2}$$

This problem superficially resembles the elementary counterforensics problem solved in [6, 7], whose core formulation is

$$\mathbf{y}^* = \underset{\substack{\mathbf{y} \in \mathcal{V}^n \\ \mathbf{y} = \Pi\mathbf{x}}}{\arg\min} \|\mathbf{z} - \mathbf{y}\|^2. \tag{3}$$

Unlike (2), however, problem (3) has a straightforward solution, easily understood using the so-called rearrangement inequalities [8]

$$(\mathbf{z}^{\downarrow})^t\mathbf{y}^{\uparrow} \leq \mathbf{z}^t\mathbf{y} \leq (\mathbf{z}^{\uparrow})^t\mathbf{y}^{\uparrow}, \tag{4}$$

which hold for any two $\mathbf{z}$ and $\mathbf{y} \in \mathbb{R}^n$. Since $\|\mathbf{y}\| = \|\Pi_{\boldsymbol{\sigma}}\mathbf{x}\| = \|\mathbf{x}\|$ for all $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$, (3) just involves the maximisation of $\mathbf{z}^t\mathbf{y} = \mathbf{z}^t\Pi_{\boldsymbol{\sigma}}\mathbf{x}$ over $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$. From the right-hand side of (4), $\mathbf{z}^t\mathbf{y} \leq (\mathbf{z}^{\uparrow})^t\mathbf{x}^{\uparrow} = \mathbf{z}^t\Pi_{\boldsymbol{\sigma}_{\mathbf{z}}}^t\Pi_{\boldsymbol{\sigma}_{\mathbf{x}}}\mathbf{x}$, where $\Pi_{\boldsymbol{\sigma}_{\mathbf{z}}}$ and $\Pi_{\boldsymbol{\sigma}_{\mathbf{x}}}$ are any two permutation matrices that sort $\mathbf{z}$ and $\mathbf{x}$, respectively, nondecreasingly. Therefore a solution to (3) is $\mathbf{y}^* = \Pi_{\boldsymbol{\sigma}_{\mathbf{z}}}^t\Pi_{\boldsymbol{\sigma}_{\mathbf{x}}}\mathbf{x} = \Pi_{\boldsymbol{\sigma}_{\mathbf{z}}}^t\mathbf{x}^{\uparrow}$. Unfortunately, this simple strategy cannot be used in the special problem (2), as briefly discussed in Section 3 using a simplified setting.

2.1. A Review of the Simplex Solution

Because of this fact, a simplex-based strategy to solve (2) was put forward by Comesaña and Pérez-González [5]. Their strategy is in fact general, and can also solve (3). For the sake of a self-contained paper, we will summarise the approach in [5] in this section. The core idea is transforming the integer nonlinear programming problem in (2) into a binary linear programming problem. The key element in the transformation is a class of $n \times q$ matrices which we denote by the set $\mathcal{L}_{\mathbf{h}^{\mathbf{x}}}$. Any $\Lambda \in \mathcal{L}_{\mathbf{h}^{\mathbf{x}}}$ fulfills three properties:

$$(\Lambda)_{i,k} \in \{0, 1\} \text{ for all } 1 \leq i \leq n, 1 \leq k \leq q, \tag{5}$$

$$\Lambda\mathbf{1} = \mathbf{1}, \tag{6}$$

$$\mathbf{1}^t\Lambda = (\mathbf{h}^{\mathbf{x}})^t. \tag{7}$$

One can see that $|\mathcal{L}_{\mathbf{h}^{\mathbf{x}}}| = \binom{n}{\mathbf{h}^{\mathbf{x}}}$, i.e, the number of rearrangements of $\mathbf{x}$, and that, given any rearrangement $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\mathbf{x}$, there is always a unique $\Lambda \in \mathcal{L}_{\mathbf{h}^{\mathbf{x}}}$ such that $\Lambda\mathbf{v} = \mathbf{y}$. Therefore $\mathcal{L}_{\mathbf{h}^{\mathbf{x}}}$ is isomorphic to the space of solutions of problem (2). Consider next an $n \times q$ cost matrix M such that $(\mathrm{M})_{i,k}$ is the minimum of $(\underline{z}_i - \underline{y}_i)^2$ when $y_i = \varphi(\underline{y}_i) = v_k \in \mathcal{V}$. In order to express these costs, define

$$b_{i,k} \triangleq \begin{cases} 10, & \text{if } z_i > v_k + 4, \\ 0.1, & \text{if } z_i < v_k - 4, \\ 1, & \text{otherwise.} \end{cases}$$

With this definition, the optimum value of $\underline{y}_i$ when its first significant figure y_i equals v_k can be put as

$$\underline{y}_{i,k}^* = \begin{cases} \underline{z}_i, & \text{if } z_i = v_k, \\ b_{i,k}\gamma_i(v_k + 0.\dot{9}), & \text{if } z_i > v_k \text{ or } z_i < v_k - 4, \\ b_{i,k}\gamma_i v_k & \text{otherwise,} \end{cases} \tag{8}$$

where a dot over a figure denotes a repeating decimal. So the costs are $(\mathrm{M})_{i,k} = (\underline{z}_i - \underline{y}_{i,k}^*)^2$. Since there is a one-to-one relationship between the elements of $\mathcal{L}_{\mathbf{h}^{\mathbf{x}}}$ and the rearrangements of $\mathbf{x}$, then problem (2) can now be recast using the cost matrix M as

$$\Lambda^* = \underset{\Lambda \in \mathcal{L}_{\mathbf{h}^{\mathbf{x}}}}{\arg\min} \ \mathrm{tr}(\Lambda\mathrm{M}^t). \tag{9}$$

Given the optimum Λ^*, we have that $\mathbf{y}^* = \Lambda^*\mathbf{v}$, and the elements of $\underline{\mathbf{y}}^*$ are obtained as in (8), i.e. if $y_i^* = v_k$ then $\underline{y}_i^* = \underline{y}_{i,k}^*$.

Leaving aside for a moment the binary constraints in (5), (9) is a regular linear programming problem with two sets of linear constraints, which can be solved using the simplex algorithm. The problem can be put in standard vector form using the identity $\mathrm{tr}(\mathrm{AB}^t) = (\mathrm{vec}\,\mathrm{B})^t\,\mathrm{vec}\,\mathrm{A}$ in the objective function, and the identity $\mathrm{vec}(\mathrm{B}\mathrm{A}\Delta) = (\Delta^t \otimes \mathrm{B})\,\mathrm{vec}\,\mathrm{A}$ in the equality constraints (6) and (7), which allow us to rewrite (9) as

$$\mathrm{vec}\,\Lambda^* = \underset{\mathrm{vec}\,\Lambda}{\arg\min} \ (\mathrm{vec}\,\mathrm{M})^t\,\mathrm{vec}\,\Lambda \tag{10}$$

$$\text{s.t. } (\mathbf{1}^t \otimes \mathrm{I}_n)\,\mathrm{vec}\,\Lambda = \mathbf{1}, \quad (\mathrm{I}_q \otimes \mathbf{1})\,\mathrm{vec}\,\Lambda = \mathbf{h}^{\mathbf{x}}.$$

The final twist in the approach in [5] is that, even if (5) is ignored, simplex guarantees that an optimum will be found on a vertex of the feasible polytope defined by the constraints (6) and (7) and the nonnegative orthant, which implies that (5) will be implicitly fulfilled.

2.2. Not so Simple(x): Motivation for a New Approach

It would appear that optimum counterforensics of significant digits stands solved by the strategy just described. However, the problem deserves further attention mainly due to two reasons. First of all, binary linear programming is an NP-hard problem [9]. In connection with this, the simplex algorithm has exponential worst-case complexity [10], even though it has been pragmatically applied to many problems due to its average polynomial complexity under some input distributions. In any case, the fact is that simplex implementations can struggle in practice as the dimensionality of the problem increases. Compounding this issue is the fact that recasting the problem (2) as (9) scales up dimensionality by a factor of q, as we go from n to nq unknown variables. For these reasons, it would be desirable to find an approach alternative to [5] without dimensionality increase and with more scalable complexity.

The second stimulus for further research concerns the distribution of the digits that follow the first significant digit. The central motivation for problem (2) was the situation where $\mathbf{h}^{\mathbf{x}}$ follows Benford's law (up to rounding errors). However Benford's law also applies beyond the first significant digit, to all subsequent figures [1]. This fact was used by Kirchner and Chakraborty [11] to level criticism against the solution in Section 2.1, which is somewhat unfair given that Comesaña and Pérez-González solely set out to settle the then unsolved problem (2). Still, the authors of [11] are right about the approach in [5] creating detectable artifacts: as it can be seen from (8), in Comesaña and Pérez-González's solution the figures after the first significant digit in the elements of $\mathbf{y}^*$ will frequently follow the patterns $000\dots$ or $999\dots$ Clearly this will break the

general version of Benford's law, which states that [1]

$$\Pr(V_d = v) = \log_{10}\left(1 + \frac{1}{v}\right), \tag{11}$$

where random variable V_d models the first d significant digits. The support of this random variable is

$$\mathcal{V}_d \triangleq \{10^{d-1}, 10^{d-1}+1, \ldots, 10^d - 1\}, \tag{12}$$

and we have that $|\mathcal{V}_d| = 9 \cdot 10^{d-1}$. For example, with $d = 2$, $\mathcal{V}_2 = \{10, 11, 12, \ldots, 98, 99\}$ and $|\mathcal{V}_2| = 90$. For the aforementioned reasons, it would be desirable to enforce the distribution of an arbitrary number d of significant digits. An added complication is that (11) implies that significant digits in different positions are statistically dependent [1]: consequently, it is not optimum to consider the distribution of each i-th significant digit separately because in this case the post-processed forgery will be detectable through higher-order strategies [12]. Finally, enforcing the distribution of d significant digits using a simplex approach like in Section 2.1 involves $n\, 9 \cdot 10^{d-1}$ variables, which only adds to the computational woes of simplex.

3. TOWARDS OPTIMUM COUNTERFORENSICS OF MULTIPLE SIGNIFICANT DIGITS

Let us now reconsider (2), keeping in mind the two main goals stated above. The main obstacle for a systematic solution of (2) without dimensionality increase is the lack of convexity of the objective function. This becomes clear when one considers the nonconvexity of (8) with respect to the potential values of y_i. For this reason we will pose an alternative convex version of the problem, which will enable finding a solution even when d significant digits are considered. The main feature of our convex alternative with respect to (2) will be the preservation of the order of magnitude of the first significant digits of $\mathbf{z}$. The absence of this constraint is the ultimate source of nonconvexity in (2). Moreover, if the priority is that the post-processed forgery be undetectable when the detector uses digits beyond the most significant one, then it makes no sense to push for maximum fidelity by frequently forcing the digits after the most significant one to follow the patterns $000\ldots$ or $999\ldots$, as done in [5]. In order to avoid the artifacts due to this strategy, we will leave unchanged all forgery digits after the d first significant ones.

3.1. Problem Formulation

We now formulate the problem addressed in this paper, taking into account the previous discussion. The vector $\mathbf{z} = \varphi_d(\underline{\mathbf{z}})$ now contains d significant digits for each element of the forgery $\underline{\mathbf{z}}$: $z_i = \lfloor |\underline{z}_i| 10^{-\lfloor \log_{10} |\underline{z}_i| \rfloor + d - 1} \rfloor$, where $z_i \in \mathcal{V}_d$. We now define $\gamma_i = \mathrm{sign}(\underline{z}_i) 10^{\lfloor \log_{10} |\underline{z}_i| \rfloor - d + 1}$, which is 10 to the order of magnitude of the d-th significant digit of $\underline{z}_i$, multiplied by its sign. The target histogram $\mathbf{h}^{\mathbf{x}}$ to be imposed on the distribution of the d most significant digits of the post-processed forgery $\underline{\mathbf{y}}$ now has $q = 9 \cdot 10^{d-1}$ elements, and its bins are given by vector $\mathbf{v} = [10^{d-1}, 10^{d-1} + 1, \ldots, 10^d - 1]^t$ (the elements of $\mathcal{V}_d$ sorted in increasing order).

In the conditions discussed above, it is straightforward to verify that $\|\underline{\mathbf{z}} - \underline{\mathbf{y}}\|^2 = \|\Gamma(\mathbf{z} - \mathbf{y})\|^2$, where $\mathbf{y} = \varphi_d(\underline{\mathbf{y}})$ gives the vector of d significant digits of the elements of the post-processed forgery $\underline{\mathbf{y}}$, and where $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_n)$. Therefore the minimisation of $\|\underline{\mathbf{z}} - \underline{\mathbf{y}}\|^2$ conditioned to the histogram of $\mathbf{y}$ being equal to target histogram $\mathbf{h}^{\mathbf{x}}$ (corresponding to decoy $\mathbf{x}$) can now be expressed as

$$\mathbf{y}^* = \underset{\substack{\mathbf{y} \in \mathcal{V}_d^n \\ \mathbf{y} = \Pi\mathbf{x}}}{\arg\min} \|\Gamma(\mathbf{z} - \mathbf{y})\|^2. \tag{13}$$

After solving this problem, the optimum post-processed forgery is simply obtained as $\underline{\mathbf{y}}^* = \underline{\mathbf{z}} + \Gamma(\mathbf{y}^* - \mathbf{z})$. The formulation in (13) resembles the elementary counterforensics problem in (3). Nevertheless, the strategy that solves (3) will not work here due to a simple fact: $\|\Gamma\mathbf{y}\| = \|\Gamma\Pi_{\boldsymbol{\sigma}}\mathbf{x}\|$ is not constant over $\boldsymbol{\sigma} \in \mathcal{S}_{\mathbf{x}}$, which precludes a solution of (13) through direct application of (4). Still, (13) can be solved using (10) by simply choosing $(\mathrm{M})_{i,k} = \gamma_i^2 (z_i - v_k)^2$. For this reason, whenever the dimensionality of the problem allows simplex to finish, we will use this method to gauge the performance of the algorithm that we will propose next.

3.2. Majorisation-Minimisation (MM)

The main advantage of (13) is its convexity. If we define the objective function in (13) as

$$f(\mathbf{y}) \triangleq \mathbf{y}^t \Gamma^2 \mathbf{y} - 2\mathbf{z}^t \Gamma^2 \mathbf{y}$$

(removing the constant term), it is straightforward to see that $\nabla^2 f(\mathbf{y}) = 2\Gamma^2$ is positive definite. However (13) is also a problem on integer variables, and so the application of standard convex optimisation techniques needs to be carefully thought out.

Many minimisation problems defy a closed-form solution, even if the objective function $f(\mathbf{y})$ is known to be convex. In this scenario the Majorisation-Minimisation philosophy [13] propounds an iterative optimisation approach based on a surrogate function $g(\mathbf{y}|\mathbf{y}_m)$ with the following two properties:

$$f(\mathbf{y}) \leq g(\mathbf{y}|\mathbf{y}_m), \qquad f(\mathbf{y}_m) = g(\mathbf{y}_m|\mathbf{y}_m). \tag{14}$$

A function $g(\mathbf{y}|\mathbf{y}_m)$ fulfilling properties (14) is said to *majorise* $f(\mathbf{y})$ at $\mathbf{y} = \mathbf{y}_m$. The key to implementing MM is finding a majorisation function easier to minimise than the original objective function. An iterative descent method becomes then possible by solving

$$\mathbf{y}_{m+1} = \underset{\mathbf{y}}{\arg\min}\, g(\mathbf{y}|\mathbf{y}_m), \tag{15}$$

since, from (14) and (15) it follows that

$$f(\mathbf{y}_{m+1}) \leq g(\mathbf{y}_{m+1}|\mathbf{y}_m) \leq g(\mathbf{y}_m|\mathbf{y}_m) = f(\mathbf{y}_m).$$

In our problem, the following standard quadratic majoriser of the objective function may be used:

$$g(\mathbf{y}|\mathbf{y}_m) \triangleq f(\mathbf{y}_m) + \nabla f(\mathbf{y}_m)^t (\mathbf{y} - \mathbf{y}_m) + \frac{1}{2}(\mathbf{y} - \mathbf{y}_m)^t \mathrm{B} (\mathbf{y} - \mathbf{y}_m),$$

provided that $\mathrm{B} - \nabla^2 f(\mathbf{y})$ is positive semidefinite. We can guarantee this by choosing $\mathrm{B} = 2\mu I$, where $\mu \triangleq \max_i \gamma_i^2$. With this choice, as $\nabla f(\mathbf{y}) = 2\Gamma^2(\mathbf{y} - \mathbf{z})$ the majoriser becomes

$$g(\mathbf{y}|\mathbf{y}_m) = f(\mathbf{y}_m) + 2(\mathbf{y}_m - \mathbf{z})^t \Gamma^2 (\mathbf{y} - \mathbf{y}_m) + \mu \|\mathbf{y} - \mathbf{y}_m\|^2. \tag{16}$$

In our problem, the use of a quadratic majoriser of a quadratic objective function is motivated by the fact that, under the problem constraints, (16) has a linear dependence with $\mathbf{y}$ rather than a quadratic one: as $\|\mathbf{y}\|^2$ must be constant when $\mathbf{y} = \Pi\mathbf{x}$, the only dependence of $g(\mathbf{y}|\mathbf{y}_m)$ on $\mathbf{y}$ is $g'(\mathbf{y}|\mathbf{y}_m) \triangleq \mathbf{w}_m^t \mathbf{y}$, where

$$\mathbf{w}_m \triangleq \Gamma^2(\mathbf{y}_m - \mathbf{z}) - \mu \mathbf{y}_m. \tag{17}$$

Therefore, an equivalent but simpler formulation of the optimum update problem (15) is $\mathbf{y}_{m+1} = \arg\min_{\mathbf{y}} g'(\mathbf{y}|\mathbf{y}_m)$.

3.3. Discrete MM

An MM iteration requires minimising $\mathbf{w}_m^t\mathbf{y}$ constrained to $\mathbf{y} = \Pi\mathbf{x}$. From the inequality on the left-hand side of (4) we have that

$$\mathbf{w}_m^t\mathbf{y} \geq (\mathbf{w}_m^{\downarrow})^t\mathbf{x}^{\uparrow},$$

and therefore a solution to (15) is $\mathbf{y}_{m+1} = \Pi_{\boldsymbol{\sigma}_\mathbf{w}}^t\mathbf{x}^{\uparrow}$, where $\Pi_{\boldsymbol{\sigma}_\mathbf{w}}$ is any permutation matrix that sorts $\mathbf{w}_m$ nonincreasingly. As $g(\mathbf{y}_{m+1}|\mathbf{y}_m) = g(\mathbf{y}_m|\mathbf{y}_m) + 2\mathbf{w}_m^t(\mathbf{y}_{m+1} - \mathbf{y}_m)$ a critical question is avoiding that $\mathbf{y}_{m+1} - \mathbf{y}_m$ becomes orthogonal to $\mathbf{w}_m$ before reaching the minimum, as this will stall the iterative descent. The reason for this potential issue is that the space of solutions is discrete, and the optimum update $\mathbf{y}_{m+1}$ not unique whenever there are sorting ties in $\mathbf{w}_m^{\downarrow}$ (the usual case). Even though all possible updates yield the same minimum of the majoriser in the m-th iteration, $\mathbf{w}_{m+1}$ will vary depending on the choice of $\mathbf{y}_{m+1}$, and thus influence the future course of the iterations.

We address next the enumeration and generation of all optimum updates. In the remainder of this section we will drop the subindex m from $\mathbf{w}_m$ to simplify the notation. Let $\mathbf{h}^{\mathbf{w}}$ be the histogram of $\mathbf{w}$ on the bins defined by its q_w unique values. Define next vectors $\mathbf{x}_k^{\uparrow}$ of length $h^{\mathbf{w}}_{q_w-k+1}$, for $k = 1, 2, \ldots, q_w$, such that $[(\mathbf{x}_1^{\uparrow})^t, (\mathbf{x}_2^{\uparrow})^t, \ldots, (\mathbf{x}_{q_w}^{\uparrow})^t]^t = \mathbf{x}^{\uparrow}$. If $\mathbf{h}^{\mathbf{x}_1^{\uparrow}}, \mathbf{h}^{\mathbf{x}_2^{\uparrow}}, \ldots, \mathbf{h}^{\mathbf{x}_{q_w}^{\uparrow}}$ are their histograms, then the number s of different optimum solutions to (15) is given by the following product of multinomial coefficients:

$$s = \prod_{k=1}^{q_w} \binom{h^{\mathbf{w}}_{q_w-k+1}}{\mathbf{h}^{\mathbf{x}_k^{\uparrow}}}. \tag{18}$$

We can spell out each of s solutions using block-diagonal permutation matrices $\Xi_{\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_{q_w}} = \mathrm{diag}(\Pi_{\boldsymbol{\sigma}_1}, \Pi_{\boldsymbol{\sigma}_2}, \ldots, \Pi_{\boldsymbol{\sigma}_{q_w}})$, where $\boldsymbol{\sigma}_k \in \mathcal{S}_{\mathbf{x}_k^{\uparrow}}$. There are s different Ξ-matrices, because $|\mathcal{S}_{\mathbf{x}_k^{\uparrow}}|$ equals the k-th multinomial in (18), and for any of them $\Xi_{\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_{q_w}}\mathbf{w}^{\downarrow} = \mathbf{w}^{\downarrow}$ (as $\Xi_{\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_q}$ only permutes elements with equal value in $\mathbf{w}^{\downarrow}$). Thus, the optimum update associated to $\Xi_{\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_{q_w}}$ is

$$\mathbf{y}_{m+1}^{(\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_{q_w})} = \Pi_{\boldsymbol{\sigma}_\mathbf{w}}^t\Xi_{\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_{q_w}}\mathbf{x}^{\uparrow}.$$

We have been unable to analytically determine the best choice among the s possible updates in order to guarantee no stalling. However, we have empirically found that a good strategy for selecting $\mathbf{y}_{m+1}$ is to use $\Pi_{\boldsymbol{\sigma}_\mathbf{w}}$ corresponding to stable sorting [14] of $\mathbf{w}$ and $\Xi_{\boldsymbol{\sigma}_1\cdots\boldsymbol{\sigma}_{q_w}}$ chosen uniformly at random. This procedure works well when the method is initialised close to the optimum, which is what we deal with in the following section.

3.4. Continuous MM

In order to find a good initialisation for the method in Section 3.3 we will solve a continuous version of (13), which we can state as

$$\mathbf{y}_c^* = \underset{\mathbf{y}\in\mathbb{R}^n}{\arg\min}\, f(\mathbf{y}) \quad \text{s.t. } \|\mathbf{y}\|^2 = \|\mathbf{x}\|^2,\ \mathbf{y}^t\mathbf{1} = \mathbf{x}^t\mathbf{1}. \tag{19}$$

Here we let $\mathbf{y} \in \mathbb{R}^n$, but we constrain the solution to lie on the same geometric loci as all $\mathbf{y} = \Pi\mathbf{x}$ (i.e., the permutation sphere $\|\mathbf{y}\|^2 = \|\mathbf{x}\|^2$ and the permutation plane $\mathbf{y}^t\mathbf{1} = \mathbf{x}^t\mathbf{1}$). The solution of (19) is $\mathbf{y}_c^* = (\Gamma^2 - \alpha I)^{-1}(\Gamma^2\mathbf{z} + (\beta/2)\mathbf{1})$. Nevertheless this is not a closed-form solution, because the Lagrange multipliers α and β must be numerically computed. More critically, the numerical issues for their determination are highly dependent on Γ and $\mathbf{z}$.

However, we will see next that an MM approach, still based on majoriser (16), enables an explicit solution at each minimisation step. Considering (17), the Lagrangian for minimising the majoriser can now be put as $\upsilon(\mathbf{y}) = \mathbf{w}_m^t\mathbf{y} - \alpha_m\mathbf{y}^t\mathbf{y} - \beta_m\mathbf{1}^t\mathbf{y}$, where α_m and β_m are Lagrange multipliers. Equating $\nabla\upsilon(\mathbf{y})$ to the null vector and solving for $\mathbf{y}$, we can see that the optimum update is

$$\mathbf{y}_{m+1} = \frac{1}{2\alpha_m}(\mathbf{w}_m - \beta_m\mathbf{1}). \tag{20}$$

Plugging this solution into the two constraints we obtain two equations to solve α_m and β_m. The solutions to these equations are

$$\beta_m = \frac{1}{n}(\mathbf{w}_m^t\mathbf{1}) \pm \frac{1}{n}(\mathbf{x}^t\mathbf{1})\sqrt{\frac{\|\mathbf{w}_m\|^2 - \frac{1}{n}(\mathbf{w}_m^t\mathbf{1})^2}{\|\mathbf{x}\|^2 - \frac{1}{n}(\mathbf{x}^t\mathbf{1})^2}} \tag{21}$$

and $\alpha_m = (\mathbf{w}_m^t\mathbf{1} - n\beta_m)/(2\,\mathbf{x}^t\mathbf{1})$. So we have two possible closed-form possibilities for (20), only one of which decreases the majoriser (by continuity and convexity). Therefore, in this case there is no convergence issue, and the global continuous minimum $\mathbf{y}_c^*$ is always found using any rearrangement of $\mathbf{x}$ to initialise the method.

a)

n	10^3	10^4	10^5	10^6	10^7	
$d=1$	51.94	53.44	52.92	49.27	49.28	MM
	51.97	53.40	52.94	49.27	—	simplex
$d=2$	51.02	50.04	49.98	50.04	50.01	MM
	51.02	50.04	49.98	—	—	simplex
$d=3$	52.61	50.08	50.02	49.99	50.01	MM
	52.61	50.08	—	—	—	simplex

b)

n	10^3	10^4	10^5	10^6	10^7	
$d=1$	59.68	60.33	59.95	55.19	55.22	MM
	65.18	65.62	64.41	60.90	—	simplex
$d=2$	58.55	59.66	59.53	59.45	59.49	MM
	65.05	67.18	66.26	—	—	simplex
$d=3$	57.59	59.82	59.31	59.53	59.52	MM
	60.49	66.46	—	—	—	simplex

Table 1. Quality of post-processed forgery in PSNR (dB) for: a) uniform forgery; and b) Gaussian forgery (standard deviation $\sigma_{\underline{\mathbf{z}}} = 5$). Benford's law is exactly enforced for d significant digits.

4. RESULTS

Table 1 shows empiricals results, using pseudorandom forgeries $\underline{\mathbf{z}}$ of size $n \in \{10^3, 10^4, 10^5, 10^6\}$ drawn from two distributions known not to comply with Benford's law (uniform and Gaussian). A target $\mathbf{h}_\mathbf{x}$ following Benford's law for d significant digits [i.e., (11) up to rounding errors] is enforced for $d \in \{1, 2, 3\}$, as the law flattens out for $d \geq 4$. The continuous MM method (Section 3.4) is used to find $\mathbf{y}_c^*$, and this solution is then used to initialise the discrete MM method (Section 3.3) using $\mathbf{y}_0 = \arg\min_{\mathbf{y}=\Pi\mathbf{x}} \|\mathbf{y}_c^* - \mathbf{y}\|^2$. The quality of the post-processed forgery $\underline{\mathbf{y}}$ is measured using the PSNR $= 10\log_{10}(n(2^b-1)^2/\|\underline{\mathbf{z}} - \underline{\mathbf{y}}\|^2)$ (dB) assuming $b = 8$ bits/sample, and the algorithm stops when the relative PSNR change between iterations is smaller than 10^{-9}. Whenever possible, the simplex approach is also used to solve (13) as described in Section 3.1, using the same $\underline{\mathbf{z}}$ as in MM for each (n, d) pair. As shown in Table 1 (marked with "—"), the simplex approach was not able to provide a solution when $d + \log_{10} n \gtrapprox 8$. The MM approach works optimally with the uniform forgery, but falls behind simplex with the Gaussian forgery due to the algorithm stalling. Still, unlike simplex, MM enforces the desired distribution in all cases tested, always producing relatively high quality post-processed forgeries. Even though further work is needed to improve convergence, we have shown that our proposal based on MM provides a promising practical way for undertaking counterforensics of multiple significant digits.

5. REFERENCES

[1] T. Hill, "The significant-digit phenomenon," *The American Mathematical Monthly*, vol. 102, pp. 322–327, 1995.

[2] S. Milani, M. Tagliasacchi, and S. Tubaro, "Antiforensics attacks to Benford's law for the detection of double compressed images," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, May 2013, pp. 3053–3057.

[3] C. Pasquini and G. Boato, "JPEG compression anti-forensics based on first significant digit distribution," in *IEEE 15th Int. Workshop on Multimedia Signal Processing (MMSP)*, Pula, Italy, September 2013, pp. 500–505.

[4] C. Pasquini, P. Comesaña-Alfaro, F. Pérez-González, and G. Boato, "Transportation-theoretic image counterforensics to first significant digit histogram forensics," in *39th IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6240–6244.

[5] P. Comesaña and F. Pérez-González, "The optimal attack to histogram-based forensic detectors is simple(x)," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Atlanta, USA, December 2014, pp. 137–142.

[6] P. Comesaña and F. Pérez-González, "Optimal counterforensics for histogram-based forensics," in *38th IEEE Int. Conference on Audio, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 3048–3052.

[7] F. Balado, "The role of permutation coding in minimum-distortion perfect counterforensics," in *39th IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6240–6244.

[8] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge, at the University Press, 1934.

[9] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., pp. 85–103. Plenum Press, 1972.

[10] V. Klee and G.J. Minty, "How good is the simplex algorithm?," in *Inequalities III*, pp. 159–175. Academic Press, New York, USA, 1972.

[11] M. Kirchner and S. Chakraborty, "A second look at first significant digit histogram restoration," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov 2015, pp. 1–6.

[12] A. De Rosa, M. Fontani, M. Massai, A. Piva, and M. Barni, "Second-order statistics analysis to cope with contrast enhancement counter-forensics," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1132–1136, Aug 2015.

[13] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[14] D. E. Knuth, *The Art of Computer Programming: Sorting and Searching*, vol. 3, Addison Wesley, 2nd edition, 1998.