

**This paper is a preprint (IEEE “accepted” status).**

**IEEE copyright notice.** © 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Linear and Geometric Mixtures – Analysis

Christopher Mattern

Technische Universität Ilmenau

Ilmenau, Germany

`christopher.mattern@tu-ilmenau.de`

## Abstract

Linear and geometric mixtures are two methods to combine arbitrary models in data compression. Geometric mixtures generalize the empirically well-performing PAQ7 mixture. Both mixture schemes rely on weight vectors, which heavily determine their performance. Typically weight vectors are identified via Online Gradient Descent. In this work we show that one can obtain strong code length bounds for such a weight estimation scheme. These bounds hold for arbitrary input sequences. For this purpose we introduce the class of *nice* mixtures and analyze how Online Gradient Descent with a fixed step size combined with a nice mixture performs. These results translate to linear and geometric mixtures, which are nice, as we show. The results hold for PAQ7 mixtures as well, thus we provide the first theoretical analysis of PAQ7.

## 1 Introduction

**Background.** The combination of multiple probability distributions plays a key role in modern statistical data compression algorithms, such as Prediction by Partial Matching (PPM), Context Tree Weighting (CTW) and “Pack” (PAQ) [6, 7, 8, 11]. Statistical compression algorithms split compression into *modeling* and *coding* and process an input sequence symbol-by-symbol. During modeling a model computes a *model distribution*  $p$  and during coding an encoder maps the next character  $x$ , given  $p$ , to a codeword of a length close to  $-\log p(x)$ . Decoding is the very reverse: Given  $p$  and the codeword the decoder restores  $x$ . Arithmetic Coding (AC) is the de facto standard en-/decoder, it closely approximates the ideal code length [3]. All of the aforementioned algorithms combine (or *mix*) multiple model distributions into a single model distribution in each step. PAQ is able to mix *arbitrary* distributions. As its superior empirical performance shows, mixing arbitrary models is a promising approach.

**Previous Work.** To our knowledge there exist few compression algorithms which combine arbitrary models. Volf’s Snake- and Switching-Algorithms [10] were the first approaches to combine just *two* arbitrary models. Kufleitner et al. [5] proposed Beta-Weighting, a CTW-spin-off, which mixes arbitrary models by weighting the model distributions linearly. The weights are posterior probabilities on the models (based on a given prior distribution). Another linear weighting scheme was introduced by Veness [9], who transferred techniques for tracking from the online learning literature to statistical data compression. His weighting scheme is based on a cleverly chosen prior distribution, which enjoys good theoretical guarantees. Starting in 2002 Mahoney introduced PAQ and its successors [7], which attracted great attention among practitioners. PAQ7 and its follow-ups combine models for a binary alphabet via a nonlinear ad-hoc neural network and adjust the network weights by Online Gradient Descent (OGD) with a fixed step size [7]. Up to 2012 there was no theoretical justification for PAQ7-mixing. In [6] we proposed geometric (a non-linear mixing scheme) and linear mixtures as solutions to two weighted divergence minimization problems. Geometric mixtures add a sound theoretical base to PAQ7-mixing and generalize it to non-binary alphabets. Both mixture schemes require weights, which we estimate via OGD with a fixed step size.

In machine learning online parameter estimation via OGD and its analysis is well understood [2] and has a variety of applications, which closely resemble mixture-based compression. Hence we can adopt machine learning analysis techniques for OGD in data compression to obtain theoretical guarantees. This work draws great inspiration from Zinkevich [12], who introduced projection-based OGD in online learning and from Bianchi [1] and Warmuth [4] who analyzed OGD (without projection) in various online regression settings.

**Our Contribution.** In this work we establish upper bounds on the code length for linear and geometric mixtures coupled with OGD using a fixed step size for weight estimation. The bounds show that the number of bits wasted w.r.t. a desirable competing scheme (such as a sequence of optimal weight vectors) is small. These results directly apply to PAQ7-mixing, since it is a geometric mixture for a binary alphabet and typically uses OGD with a fixed step size for weight estimation. Thus we provide the first theoretical guarantees for PAQ. To do so, in Section 3 we introduce the class of nice mixtures which we combine with OGD with a fixed step size and establish code length bounds. It turns out that the choice of the step size is of great importance. Next, in Section 4 we show that linear and geometric mixtures are nice mixtures and apply the results of Section 3. Finally in Section 5 we summarize our results.

## 2 Preliminaries

**Notation.** In general, calligraphic letters denote sets, lowercase boldface letters indicate column vectors and boldface uppercase letters name matrices. The expression  $(a_i)_{1 \leq i \leq m}$  expands to  $(a_1 \ a_2 \ \dots \ a_m)^\top$  where “ $\top$ ” is the transpose operator; the  $i$ -th component of a vector  $\mathbf{a}$  is labeled  $a_i$  and its squared euclidean norm is  $|\mathbf{a}|^2 = \mathbf{a}^\top \mathbf{a}$ . By  $\mathbf{e}_i$  we denote the  $i$ -th unit vector and  $\mathbf{1}$  is  $(1 \ 1 \ \dots \ 1)^\top \in \mathbb{R}^m$ . For any bounded set  $\mathcal{W} \subset \mathbb{R}^m$  let  $|\mathcal{W}| := \sup_{\mathbf{a}, \mathbf{b} \in \mathcal{W}} |\mathbf{a} - \mathbf{b}|$ . Further, let  $\mathcal{S} := \{\mathbf{a} \in \mathbb{R}^m \mid \mathbf{a} \geq 0 \text{ and } \mathbf{1}^\top \mathbf{a} = 1\}$  (unit  $m$ -simplex). Let  $\mathcal{X} := \{1, 2, \dots, N\}$  be an alphabet of cardinality  $1 < N < \infty$  and let  $x_a^b := x_a x_{a+1} \dots x_b$  be a sequence over  $\mathcal{X}$  where  $x^n$  abbreviates  $x_1^n$ . The set of all probability distributions over  $\mathcal{X}$  with non-zero probabilities on all letters is  $\mathcal{P}_+$  and with probability at least  $\varepsilon > 0$  on all letters is  $\mathcal{P}_\varepsilon$ . For  $p_1, p_2, \dots, p_m \in \mathcal{P} \subseteq \mathcal{P}_+$  let  $\mathbf{p}(x) = (p_i(x))_{1 \leq i \leq m}$  be the vector of probabilities of  $x$ , the matrix  $\mathbf{P} := (\mathbf{p}(1) \ \dots \ \mathbf{p}(N))$  is called a probability matrix over  $\mathcal{P}$ . Furthermore we set  $p_{\max}(x; \mathbf{P}) := \max_{1 \leq i \leq m} p_i(x)$  and  $p_{\max}(\mathbf{P}) := \max_{x \in \mathcal{X}} p_{\max}(x; \mathbf{P})$ ;  $p_{\min}(x; \mathbf{P})$  and  $p_{\min}(\mathbf{P})$  are defined analogously. We omit the dependence on  $\mathbf{P}$ , whenever clear from the context. The natural logarithm is “ $\ln$ ”, whereas “ $\log$ ” is the base-two logarithm. For a vector  $\mathbf{a}$  with positive entries we define  $\log \mathbf{a} := (\log a_i)_{1 \leq i \leq m}$ . For  $x \in \mathcal{X}$  and  $p \in \mathcal{P}_+$  we denote the (ideal) code length of  $x$  w.r.t.  $p$  as  $\ell(x, p) := -\log p(x)$ . The expression  $\nabla_{\mathbf{w}} f := (\partial f / \partial w_i)_{1 \leq i \leq m}$  denotes the gradient of a function  $f$ , when unambiguous we write  $\nabla f$  in place of  $\nabla_{\mathbf{w}} f$ .

**The Setting.** Recall the process of statistical data compression for a sequence  $x^n$  over  $\mathcal{X}$  (see Section 1), which we now formally refine to our setting of interest. Fix an arbitrary step  $1 \leq k \leq n$ . First, we represent the  $m > 1$  model distributions  $p_1, \dots, p_m \in \mathcal{P}_+$  (which may depend on  $x^{k-1}$  and typically vary from step to step) in a probability matrix  $\mathbf{P}_k$ . One can think of  $x^n$  and the sequence  $\mathbf{P}^n := \mathbf{P}_1, \dots, \mathbf{P}_n$  of probability matrices over  $\mathcal{P}_+$  as fixed. On the basis of  $\mathbf{P}_k$  we determine a mixture distribution (for short *mixture*)  $\text{MIX}(\mathbf{w}, \mathbf{P}_k)$  for coding the  $k$ -th character  $x_k$  in  $\ell(x_k, \text{MIX}(\mathbf{w}, \mathbf{P}_k))$  bits. The mixture depends on a parameter vector or *weight vector*  $\mathbf{w} = \mathbf{w}_k$  which is typically constrained to a domain  $\mathcal{W}$  (a non-empty, compact, convex subset of  $\mathbb{R}^m$ ). Based on an initial weight vector  $\mathbf{w}_1$  (chosen by the user) we generate a sequence of weight vectors  $\mathbf{w}_2, \mathbf{w}_3, \dots$  via OGD: In step  $k$  we adjust  $\mathbf{w}_k$  by a step towards  $\mathbf{d} := -\alpha \nabla_{\mathbf{w}} \ell(x_k, \text{MIX}(\mathbf{w}, \mathbf{P}_k))$  where  $\alpha > 0$  is the step size. The resulting vector  $\mathbf{v} = \mathbf{w}_k + \mathbf{d}$  might not lie in  $\mathcal{W}$ , the operation  $\text{proj}(\mathbf{v}; \mathcal{W}) := \arg \min_{\mathbf{w} \in \mathcal{W}} |\mathbf{v} - \mathbf{w}|^2$  maps a vector  $\mathbf{v} \in \mathbb{R}^m$  back to the feasible set  $\mathcal{W}$  and we obtain  $\mathbf{w}_{k+1} = \text{proj}(\mathbf{v}; \mathcal{W})$ . Algorithm 1 summarizes this process. Next we define the general term mixture as well as linear and geometric mixtures.

---

**Algorithm 1:** MIX-OGD( $\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n$ )

---

**Input** : a weight estimation  $\mathbf{w}_1 \in \mathcal{W}$ , a step size  $\alpha > 0$ , a sequence  $x^n$  over  $\mathcal{X}$ ,  
and a sequence  $\mathbf{P}^n$  of probability matrices over  $\mathcal{P}_+$

**Output** : a codeword for  $x^n$  of length  $\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n))$

---

- 1 **for**  $k \leftarrow 1$  **to**  $n$  **do**
  - 2     compute  $p \leftarrow \text{MIX}(\mathbf{w}_k, \mathbf{P}_k)$  and emit a codeword for  $x_k$  sized  $\ell(x_k, p)$  bits;
  - 3      $\mathbf{w}_{k+1} \leftarrow \text{proj}(\mathbf{w}_k - \alpha \nabla_{\mathbf{w}} \ell(x_k, \text{MIX}(\mathbf{w}, \mathbf{P}_k))|_{\mathbf{w}=\mathbf{w}_k}; \mathcal{W})$ ;
- 

**Definition 2.1.** A mixture  $\text{MIX} : (\mathbf{w}, \mathbf{P}) \mapsto p$  maps a probability matrix  $\mathbf{P}$  over  $\mathcal{P}_+$ , given a parameter vector  $\mathbf{w}$  drawn from the parameter space  $\mathcal{W}$ , to a mixture distribution  $p \in \mathcal{P}_+$ . The shorthand  $\text{MIX}(x; \mathbf{w}, \mathbf{P})$  is for  $p(x)$  where  $p = \text{MIX}(\mathbf{w}, \mathbf{P})$ .

**Definition 2.2.** For weight (parameter) vector  $\mathbf{w} \in \mathcal{S}$  and probability matrix  $\mathbf{P}$  over  $\mathcal{P}_+$  the linear mixture LIN is defined by  $\text{LIN}(x; \mathbf{w}, \mathbf{P}) := \mathbf{w}^\top \mathbf{p}(x)$ .

**Definition 2.3.** For weight (parameter) vector  $\mathbf{w} \in \mathbb{R}^m$  and probability matrix  $\mathbf{P}$  over  $\mathcal{P}_+$  the geometric mixture GEO is defined by  $\text{GEO}(x; \mathbf{w}, \mathbf{P}) := \prod_{i=1}^m p_i(x)^{w_i} / \sum_{y \in \mathcal{X}} \prod_{i=1}^m p_i(y)^{w_i}$ .

**Observation 2.4.** If  $\mathbf{l}(x) := -\log \mathbf{p}(x)$ , then  $\text{GEO}(x; \mathbf{w}, \mathbf{P}) = 2^{-\mathbf{w}^\top \mathbf{l}(x)} / \sum_{y \in \mathcal{X}} 2^{-\mathbf{w}^\top \mathbf{l}(y)}$ .

In the following we will draw heavily on the alternate expression for  $\text{GEO}(x; \mathbf{w}, \mathbf{P})$  given in Observation 2.4. This expression simplifies some of the upcoming calculations. Furthermore, let

$$\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) := \sum_{k=1}^n \ell(x_k, \text{MIX}(\mathbf{w}_k, \mathbf{P}_k)) \text{ (for } \mathbf{w}_k \text{ see Algorithm 1),}$$

$$\ell(x^n, \mathbf{P}^n, \mathbf{w}, \text{MIX}) := \sum_{k=1}^n \ell(x_k, \text{MIX}(\mathbf{w}, \mathbf{P}_k)) \text{ and } \ell^*(x^n, \mathbf{P}^n, \text{MIX}) := \min_{\mathbf{w} \in \mathcal{W}} \ell(x^n, \mathbf{P}^n, \mathbf{w}, \text{MIX}).$$

### 3 Nice Mixtures and Code Length Bounds

**Nice mixtures.** We now introduce a class of especially interesting mixtures. We call such mixtures *nice*. A nice mixture satisfies a couple of properties that allow us to derive bounds on the code length of combining such a mixture with OGD for parameter estimation (e.g. weight estimation). These properties have been chosen carefully, s.t. linear and geometric mixtures fall into the class of nice mixtures (see Section 4).

**Definition 3.1.** A mixture MIX is called *nice* if

1. the parameter space  $\mathcal{W}$  is a non-empty, compact and convex subset of  $\mathbb{R}^m$ ,
2.  $\ell(x, \text{MIX}(\mathbf{w}, \mathbf{P}))$  is convex in  $\mathbf{w} \in \mathcal{W}$  for all  $\mathbf{P}$  over  $\mathcal{P}^+$  and all  $x \in \mathcal{X}$ ,
3.  $\ell(x, \text{MIX}(\mathbf{w}, \mathbf{P}))$  is differentiable by  $\mathbf{w}$  for all  $\mathbf{P}$  over  $\mathcal{P}^+$  and all  $x \in \mathcal{X}$  and
4. there exists a constant  $a > 0$  s.t.  $|\nabla_{\mathbf{w}} \ell(x, \text{MIX}(\mathbf{w}, \mathbf{P}))|^2 \leq a \cdot \ell(x, \text{MIX}(\mathbf{w}, \mathbf{P}))$  for all  $\mathbf{w} \in \mathcal{W}$ ,  $\mathbf{P}$  over  $\mathcal{P}^+$  and  $x \in \mathcal{X}$ .

*Remark 3.2.* Properties 1 to 3 are similar to the assumptions made in [12], Property 4 differs. This allows us to obtain meaningful bounds on  $\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n))$  when  $\alpha$  is independent of  $n$ , as [1, 4] show.

**Bounds on the Code Length for OGD.** Algorithm 1 illustrates an online algorithm for mixture-based statistical data compression which employs a mixture MIX. We want to analyze the algorithm in terms of the number of bits required to encode a sequence when MIX is nice. We strive to show that in some sense the code length produced by Algorithm 1 is not much worse than a desirable competing scheme. At first we choose the code length produced by the best static weight vector  $\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathcal{W}} \ell(x^n, \mathbf{P}^n, \mathbf{w}, \text{MIX})$  as the competing scheme.

**Proposition 3.3.** *Algorithm 1 run with a nice mixture MIX, initial weight vector  $\mathbf{w}_1 \in \mathcal{W}$  and step size  $\alpha = 2(1 - b^{-1})/a$  for  $b > 1$  (the constant  $a$  is due to Definition 3.1, Property 4) satisfies*

$$\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \leq b \cdot \ell^*(x^n, \mathbf{P}^n, \text{MIX}) + \frac{a}{4} \frac{b^2}{b-1} \cdot \|\mathbf{w}_1 - \mathbf{w}^*\|^2, \quad (1)$$

where  $\mathbf{w}^*$  minimizes  $\ell(x^n, \mathbf{P}^n, \mathbf{w}, \text{MIX})$ , for all  $x^n$  over  $\mathcal{X}$  and all  $\mathbf{P}^n$  over  $\mathcal{P}_+$ .

*Proof.* For brevity we set  $\ell_k(\mathbf{w}) := \ell(x_k, \text{MIX}(\mathbf{w}, \mathbf{P}_k))$ . As in [4], for arbitrary  $\mathbf{w} \in \mathcal{W}$ , we first establish a lower bound on

$$\|\mathbf{w}_k - \mathbf{w}\|^2 - \|\mathbf{w}_{k+1} - \mathbf{w}\|^2 = \|\mathbf{w}_k - \mathbf{w}\|^2 - \|\text{proj}(\mathbf{w}_k - \alpha \nabla \ell_k(\mathbf{w}_k); \mathcal{W}) - \mathbf{w}\|^2.$$

For  $\mathbf{v} \in \mathbb{R}^m$  and  $\mathbf{w} \in \mathcal{W}$  it is well-known [12], that  $\|\text{proj}(\mathbf{v}; \mathcal{W}) - \mathbf{w}\| \leq \|\mathbf{v} - \mathbf{w}\|$ , i.e.

$$\begin{aligned} \|\mathbf{w}_k - \mathbf{w}\|^2 - \|\mathbf{w}_{k+1} - \mathbf{w}\|^2 &\geq \|\mathbf{w}_k - \mathbf{w}\|^2 - \|(\mathbf{w}_k - \mathbf{w}) - \alpha \nabla \ell_k(\mathbf{w}_k)\|^2 \\ &= 2\alpha \nabla \ell_k(\mathbf{w}_k)^\top (\mathbf{w}_k - \mathbf{w}) - \alpha^2 \|\nabla \ell_k(\mathbf{w}_k)\|^2. \end{aligned}$$

Since MIX is nice,  $\ell_k(\mathbf{w})$  is convex (due to Definition 3.1, Property 2) and we have  $\ell_k(\mathbf{v}) - \ell_k(\mathbf{w}) \leq \nabla \ell_k(\mathbf{v})^\top (\mathbf{v} - \mathbf{w})$  for any  $\mathbf{w}, \mathbf{v} \in \mathcal{W}$ . We deduce

$$\begin{aligned} \|\mathbf{w}_k - \mathbf{w}\|^2 - \|\mathbf{w}_{k+1} - \mathbf{w}\|^2 &\geq 2\alpha(\ell_k(\mathbf{w}_k) - \ell_k(\mathbf{w})) - \alpha^2 \|\nabla \ell_k(\mathbf{w}_k)\|^2 \\ &\geq 2\alpha(\ell_k(\mathbf{w}_k) - \ell_k(\mathbf{w})) - a\alpha^2 \ell_k(\mathbf{w}_k), \end{aligned} \quad (2)$$

the last inequality follows from Definition 3.1, Property 4. Next, we sum the previous inequality over  $k$  to obtain (the sum telescopes)

$$\alpha(2 - a\alpha) \sum_{k=1}^n (\ell_k(\mathbf{w}_k)) - 2\alpha \sum_{k=1}^n \ell_k(\mathbf{w}) \leq \sum_{k=1}^n \|\mathbf{w}_k - \mathbf{w}\|^2 - \|\mathbf{w}_{k+1} - \mathbf{w}\|^2 \leq \|\mathbf{w}_1 - \mathbf{w}\|^2,$$

which we solve for the first sum:

$$\sum_{k=1}^n \ell_k(\mathbf{w}_k) \leq \frac{2}{2 - a\alpha} \sum_{k=1}^n (\ell_k(\mathbf{w})) + \frac{\|\mathbf{w}_1 - \mathbf{w}\|^2}{\alpha(2 - a\alpha)}.$$

Since this holds for any  $\mathbf{w}$ , it must hold for  $\mathbf{w} = \mathbf{w}^*$ , too. By the definition of  $\ell_k(\mathbf{w})$  we have  $\sum_{k=1}^n \ell_k(\mathbf{w}_k) = \ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n))$  and  $\sum_{k=1}^n \ell_k(\mathbf{w}) = \ell(x^n, \mathbf{P}^n, \mathbf{w}, \text{MIX})$ . Our choice of  $\alpha$  gives (1).  $\square$

*Remark 3.4.* The technique of using a progress invariant (c.f. (2)) in the previous proof is adopted from the machine learning community, see [1, 4]. These two papers assume that the domain of the parameter (weight) vector  $\mathbf{w}$  is unbounded. Techniques of [12] allow us to overcome this limitation. Proposition 3.3 generalizes the analysis of online regression of [1] to prediction functions  $f(\mathbf{w}, \mathbf{z})$  ( $\mathbf{z}$  is the input vector for a prediction) instead of  $f(\mathbf{w}^\top \mathbf{z})$  when the domain of  $\mathbf{w}$  is restricted.

The previous proposition is good news. The number of bits required to code any sequence will be within a multiplicative constant  $b$  of the code length generated by weighting with an optimal fixed weight vector,  $\ell^*(x^n, \mathbf{P}^n, \text{MIX})$ , plus an  $O(1)$  term. At the expense of increasing the  $O(1)$  term we can set the multiplicative constant  $b$  arbitrarily close to 1. Note that the  $O(1)$ -term originates in the inaccuracy of the initial weight estimation  $\|\mathbf{w}_1 - \mathbf{w}^*\|$  (see (1)) and as  $b$  approaches 1, the step size  $\alpha$  approaches zero. Hence the  $O(1)$  term in (1) penalizes a slow movement away from  $\mathbf{w}_1$ . A high proximity of  $\mathbf{w}_1$  to the optimal weight vector  $\mathbf{w}^*$

damps this penalization. We now make two key observations, which allow us to greatly strengthen the result of Proposition 3.3.

**Observation 3.5.** From the previous discussion we know that the significance of the  $O(1)$  term vanishes as  $\ell^*(x^n, \mathbf{P}^n, \text{MIX})$  grows. We can allow small values of  $b$  for large values of  $n$ , i.e.,  $b$  may depend on  $n$ . Thus we choose  $b = 1 + f(n)$  where,  $f(n)$  decreases, and obtain

$$\begin{aligned} & \ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \\ & \leq \ell^*(x^n, \mathbf{P}^n, \text{MIX}) + \ell^*(x^n, \mathbf{P}^n, \text{MIX}) \cdot f(n) + \frac{a(1 + f(1))^2 |\mathbf{w}_1 - \mathbf{w}^*|^2}{4} \cdot \frac{1}{f(n)}. \end{aligned}$$

If  $\ell^*(x^n, \mathbf{P}^n, \text{MIX})$  is  $O(n)$  (i.e.,  $\text{MIX}(x; \mathbf{w}, \mathbf{P})$  is bounded below by a constant, which is a natural assumption) then the rightmost two terms on the previous line are  $O(n \cdot f(n) + f(n)^{-1})$  (since by Definition 3.1, Property 1,  $|\mathbf{w}_1 - \mathbf{w}^*|$  is  $O(1)$ ) and represent the number of bits wasted by MIX-OGD w.r.t.  $\ell^*(x^n, \mathbf{P}^n, \text{MIX})$ . Clearly the rate of growth is minimized in the  $O$ -sense if we choose  $f(n) = n^{-1/2}$ , i.e.  $\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \leq \ell^*(x^n, \mathbf{P}^n, \text{MIX}) + O(n^{1/2})$ . The average code length excess of MIX-OGD over  $\ell^*(x^n, \mathbf{P}^n, \text{MIX})$  vanishes asymptotically.

**Observation 3.6.** The state of MIX-OGD right after step  $k$  is captured completely by the single weight vector  $\mathbf{w}_{k+1}$ . Hence we can view running  $\text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)$  as first executing  $\text{MIX-OGD}(\mathbf{w}_1, \alpha, x^k, \mathbf{P}^k)$  and running  $\text{MIX-OGD}(\mathbf{w}_{k+1}, \alpha, x_{k+1}^n, \mathbf{P}_{k+1}^n)$  afterwards. The code lengths for these procedures match for all  $1 \leq k < n$ :

$$\begin{aligned} & \ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \\ & = \ell(x^k, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^k, \mathbf{P}^k)) + \ell(x_{k+1}^n, \text{MIX-OGD}(\mathbf{w}_{k+1}, \alpha, x_{k+1}^n, \mathbf{P}_{k+1}^n)). \end{aligned}$$

Given the previous observations as tools of trade we now enhance Proposition 3.3.

**Theorem 3.7.** We consider sequences  $t_1 = 1 < t_2 < \dots < t_s < t_{s+1} = n + 1$  of integers for  $1 \leq s \leq n$ . Let  $\ell^*(i, j, \text{MIX}) := \ell^*(x_i^j, \mathbf{P}_i^j, \text{MIX})$ . For all  $x^n \in \mathcal{X}^n$ , all  $\mathbf{P}^n$  over  $\mathcal{P}_+$ , any nice mixture MIX and any  $\mathbf{w}_1 \in \mathcal{W}$  Algorithm 1 satisfies:

1. If  $\alpha = 2(1 - b^{-1})/a$ , where  $b > 1$ , then

$$\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \leq \min_{s, t_2, \dots, t_s} \left[ \frac{ab^2 |\mathcal{W}|^2}{4(b-1)} s + b \sum_{i=1}^s \ell^*(t_i, t_{i+1}-1, \text{MIX}) \right]. \quad (3)$$

2. If  $\alpha = 2/a \cdot (1 + n^{1/2})^{-1}$  (i.e.,  $b = 1 + n^{-1/2}$ ) and  $\ell^*(x^n, \mathbf{P}^n, \text{MIX}) \leq c \cdot n$  holds for a constant  $c > 0$ , all  $x^n$  over  $\mathcal{X}$  and all  $\mathbf{P}^n$  over  $\mathcal{P}_+$  then

$$\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \leq \min_{s, t_2, \dots, t_s} \left[ (as|\mathcal{W}|^2 + c) \sqrt{n} + \sum_{i=1}^s \ell^*(t_i, t_{i+1}-1, \text{MIX}) \right] \quad (4)$$

*Proof.* We start proving (3). First, we define  $\ell_k(\mathbf{w}) := \ell(x_k, \text{MIX}(\mathbf{w}, \mathbf{P}_k))$ . By Observation 3.6 for any  $1 \leq s \leq n$  and  $t_1 = 1 < t_2 < \dots < t_s < t_{s+1} = n + 1$  we may write

$$\begin{aligned} \ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) &= \sum_{i=1}^s \ell(x_{t_i}^{t_{i+1}-1}, \text{MIX-OGD}(\mathbf{w}_{t_i}, \alpha, x_{t_i}^{t_{i+1}-1}, \mathbf{P}_{t_i}^{t_{i+1}-1})) \\ &\leq \frac{a|\mathcal{W}|^2}{4} \frac{b^2}{b-1} \cdot s + b \sum_{i=1}^s \ell^*(t_i, t_{i+1}-1, \text{MIX}). \end{aligned} \quad (5)$$

For the last step we used Proposition 3.3, the definition of  $\ell^*(t_i, t_{i+1}-1, \text{MIX})$  and Definition 3.1, Property 1 which implies that  $|\mathbf{v} - \mathbf{w}| \leq |\mathcal{W}|$  for any  $\mathbf{v}, \mathbf{w} \in \mathcal{W}$ . Since this holds for arbitrary  $s$  and  $t_2, \dots, t_s$  we can take the minimum over the corresponding entities, which gives (3).

Now we turn to (4). The choice  $b = 1 + n^{-1/2}$  follows from Observation 3.5. We combine  $b^2/(b-1) \leq 4n^{1/2}$  (by the choice of  $b$ ) with  $\ell^*(x^n, \mathbf{P}^n, \text{MIX}) \leq c \cdot n$ , i.e.  $\ell^*(i, j, x^n) \leq c \cdot (j-i+1)$  for  $j \geq i$  in the r.h.s. of (5) to yield

$$\begin{aligned} \ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) &\leq a|\mathcal{W}|^2 s \cdot n^{1/2} + (1 + n^{-1/2}) \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{MIX}) \\ &\leq (a|\mathcal{W}|^2 s + c) \cdot n^{1/2} + \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{MIX}). \end{aligned}$$

As in the proof of (3) we take the minimum over  $s$  and  $t_2, \dots, t_s$ , which gives (4).  $\square$

The previous theorem gives much stronger bounds than Proposition 3.3, since the competing scheme is a sequence of weight vectors with a total code length of  $\ell^*(t_1, t_2 - 1, \text{MIX}) + \dots + \ell^*(t_s, t_{s+1} - 1, \text{MIX})$ , where the  $i$ -th weight vector minimizes the code length of the  $i$ -th subsequence  $x_{t_i} \dots x_{t_{i+1}-1}$  of  $x^n$ . By (3) the performance of Algorithm 1 is within a multiplicative constant  $b > 1$  of the performance of any competing scheme (since in (3) we take the minimum over all competing schemes) plus an  $O(s)$ -term, when  $\alpha$  is independent of  $n$ . The  $O(s)$  term penalizes the complexity of a competing predictor (the number  $s$  of subsequences). When  $\alpha$  depends on  $n$  (c.f. (4)) we can reduce the multiplicative constant to 1 at the expense of increasing the penalty term to  $O(s\sqrt{n})$ , i.e. Algorithm 1 will asymptotically perform not much worse than any such competing scheme with  $s = o(\sqrt{n})$  subsequences.

#### 4 Bounds for Geometric and Linear Mixtures

**Geometric and Linear Mixtures are Nice.** We can only apply the machinery of the previous section to geometric and linear mixtures if they fall into the class of nice mixtures. Since the necessary conditions have been chosen carefully, this is the case:

**Lemma 4.1.** *The geometric mixture  $\text{GEO}(\mathbf{w}, \mathbf{P})$  is nice for  $\mathbf{w} \in \mathcal{W}$ , if  $\mathcal{W}$  is a compact and convex subset of  $\mathbb{R}^m$ . Property 4 of Definition 3.1 is satisfied for  $a \geq \frac{m}{\log(e)} \log^2(p_{\max}/p_{\min})$ .*

**Lemma 4.2.** *The linear mixture  $\text{LIN}(\mathbf{w}, \mathbf{P})$  is nice. Property 4 of Definition 3.1 is satisfied for  $a \geq m \log^2(e) \frac{p_{\max}^2}{p_{\min}^2 \log(1/p_{\min})}$ .*

Before we prove these two lemmas we give two technical results. The proofs of the lemmas below use standard calculus, we omit them for reasons of space.

**Lemma 4.3.** *For  $0 < z < 1$  the function  $f(z) := -\frac{\ln z}{1-z}$  satisfies  $f(z) \geq 1$ .*

**Lemma 4.4.** *For  $0 < a \leq z \leq 1 - a$  the function  $f(z) := -z^2 \ln z$  satisfies  $f(z) \geq f(a)$ .*

Now we are ready to prove Lemma 4.1 and Lemma 4.2.

*Proof of Lemma 4.1.* Let  $p(x; \mathbf{w}) := \text{GEO}(x; \mathbf{w}, \mathbf{P})$  and  $\ell(\mathbf{w}) := \ell(x, \text{GEO}(\mathbf{w}, \mathbf{P}))$ . To show the claim we must make sure that properties 1-4 of Definition 3.1 are met. By the constraint on  $\mathcal{W}$  Property 1 is satisfied. Property 2 was shown in [6, Section 3.2]. To see that Property 3 holds, we set  $c := \sum_{y \in \mathcal{X}} 2^{-\mathbf{w}^\top \mathbf{l}(y)}$  and compute

$$\nabla \ell(\mathbf{w}) = \nabla_{\mathbf{w}} (\mathbf{w}^\top \mathbf{l}(x) + \log c) = \mathbf{l}(x) - \sum_{y \in \mathcal{X}} \frac{2^{-\mathbf{w}^\top \mathbf{l}(y)}}{c} \cdot \mathbf{l}(y),$$

which is (by the definition of GEO)

$$\nabla \ell(\mathbf{w}) = \nabla_{\mathbf{w}} \ell(x, \text{GEO}(\mathbf{w}, \mathbf{P})) = \sum_{y \neq x} \text{GEO}(y; \mathbf{w}, \mathbf{P}) \cdot (\mathbf{l}(x) - \mathbf{l}(y)). \quad (6)$$

Clearly (6) is well-defined for the given range of  $\mathbf{w}$  and  $\mathbf{P}$ . For Property 4 we bound  $|\nabla \ell(\mathbf{w})|^2 / \ell(\mathbf{w})$  from above by a constant;  $a$  takes at least the value of this constant. We obtain

$$\begin{aligned} |\nabla \ell(\mathbf{w})|^2 &\leq \sum_{y \neq x} p(y; \mathbf{w}) |\mathbf{l}(x) - \mathbf{l}(y)|^2 = \sum_{y \neq x} p(y; \mathbf{w}) \sum_{i=1}^m \log^2 \frac{p_i(y)}{p_i(x)} \\ &\leq \sum_{y \neq x} p(y; \mathbf{w}) m \log^2 \frac{p_{\max}}{p_{\min}} = (1 - p(x; \mathbf{w})) m \log^2 \frac{p_{\max}}{p_{\min}} \text{ and} \\ \frac{|\nabla \ell(\mathbf{w})|^2}{\ell(\mathbf{w})} &\leq \frac{(1 - p(x; \mathbf{w})) m \log^2 \left( \frac{p_{\max}}{p_{\min}} \right)}{-\log p(x; \mathbf{w})} \leq \frac{m \log^2 \left( \frac{p_{\max}}{p_{\min}} \right)}{\log(e)} \cdot \left[ \inf_{0 < z < 1} -\frac{\ln z}{1 - z} \right]^{-1}. \end{aligned}$$

By Lemma 4.3 the infimum is at least 1. This yields the claimed lower bound on  $a$ .  $\square$

*Remark 4.5.* It is interesting to note that we can express  $\nabla_{\mathbf{w}} \ell(x, \text{GEO}(\mathbf{w}, \mathbf{P}))$  (see (6)) in terms of information theoretic quantities (for the basic notation see, e.g. [3]). The  $i$ -th component is

$$\begin{aligned} &-\log p_i(x) - \sum_{y \in \mathcal{X}} \text{GEO}(y; \mathbf{w}, \mathbf{P}) (-\log p_i(y)) \\ &= -\log p_i(x) - \sum_{y \in \mathcal{X}} \text{GEO}(y; \mathbf{w}, \mathbf{P}) \left[ \log \left( \frac{1}{\text{GEO}(y; \mathbf{w}, \mathbf{P})} \right) + \log \frac{\text{GEO}(y; \mathbf{w}, \mathbf{P})}{p_i(y)} \right] \\ &= -\log p_i(x) - (H(\text{GEO}(\mathbf{w}, \mathbf{P})) + D(\text{GEO}(\mathbf{w}, \mathbf{P}) \parallel p_i)). \end{aligned}$$

If we now ignore possible constraints on the weight vector  $\mathbf{w}$  then for some character  $x$  a minimizer of  $\min_{\mathbf{w}} \ell(x, \text{GEO}(\mathbf{w}, \mathbf{P}))$  satisfies  $H(\text{GEO}(\mathbf{w}, \mathbf{P})) + D(\text{GEO}(\mathbf{w}, \mathbf{P}) \parallel p_i) = -\log p_i(x)$  for all  $1 \leq i \leq m$ . In effect the weight vector  $\mathbf{w}$  is chosen s.t. there is an equilibrium: The code length  $-\log p_i(x)$  matches the average code length of coding a symbol drawn from the source distribution  $\text{GEO}(\mathbf{w}, \mathbf{P})$  with the model distribution  $p_i$ .

*Proof of Lemma 4.2.* Again we set  $p(x; \mathbf{w}) := \text{LIN}(x; \mathbf{w}, \mathbf{P})$  and  $\ell(\mathbf{w}) := \ell(x, \text{LIN}(\mathbf{w}, \mathbf{P}))$  and proceed analogously to the proof of Lemma 4.1. By Definition 2.2 we have  $\mathbf{w} \in \mathcal{S}$ , Property 1 is met, and in [6, Section 4.2] we showed that Property 2 is met, as well. The gradient

$$\nabla \ell(\mathbf{w}) = \nabla_{\mathbf{w}} \ell(x, \text{LIN}(\mathbf{w}, \mathbf{P})) = -\nabla_{\mathbf{w}} \log \mathbf{w}^T \mathbf{p}(x) = -\log(e) \frac{\mathbf{p}(x)}{p(x; \mathbf{w})}$$

is well-defined for the given range of  $\mathbf{w}$  and  $\mathbf{P}$ , so Property 3 is fulfilled. We observe that

$$\frac{|\nabla \ell(\mathbf{w})|^2}{\ell(\mathbf{w})} \leq \frac{m \log^2(e) p_{\max}^2}{p(x; \mathbf{w})^2 (-\log p(x; \mathbf{w}))} \leq m \log(e) p_{\max}^2 \cdot \left[ \inf_{c \leq z \leq d} -z^2 \ln z \right]^{-1} \quad (7)$$

where  $c = p_{\min} \leq p(x; \mathbf{w}) \leq p_{\max} \leq d = 1 - p_{\min}$ . We used  $p_{\max} \leq 1 - p_{\min}$ , since

$$p_{\max} = \max_{1 \leq i \leq m} \max_{x \in \mathcal{X}} p_i(x) \leq \max_{1 \leq i \leq m} \left( 1 - \min_{x \in \mathcal{X}} p_i(x) \right) = 1 - \min_{1 \leq i \leq m} \min_{x \in \mathcal{X}} p_i(x) = 1 - p_{\min},$$

to apply Lemma 4.4 to bound the rightmost factor in (7) from above by  $[-p_{\min}^2 \ln p_{\min}]^{-1}$ . The resulting constant on the r.h.s. of (7) is a lower bound on  $a$ . The proof is done.  $\square$

**Upper bounds on the Code Length.** At this point we can combine Theorem 3.7 with Lemmas 4.1 and 4.2 to obtain code length bounds on Algorithm 1 for LIN and GEO. The discussion in Section 3 on nice mixtures coupled with Algorithm 1 applies to LIN and GEO as well.

**Theorem 4.6.** *Let  $x^n \in \mathcal{X}^n$ , let  $\mathbf{P}^n$  be a sequence of probability matrices over  $\mathcal{P}_{\varepsilon}$  where  $\varepsilon = 2^{-B}$  for  $1 \leq B < \infty$  and let  $\ell^*(k, l, \text{BEST}) := \min_{1 \leq i \leq m} \ell(x_k^l, \text{LIN}(\mathbf{e}_i, \mathbf{P}_k^l))$  be the code length of the best single model for  $x_k^l$ . We consider sequences  $t_1 = 1 < t_2 < \dots < t_s <$*



$t_{s+1} = n + 1$  of integers where  $1 \leq s \leq n$ . For MIX = LIN and MIX = GEO where  $\mathcal{W} = \mathcal{S}$  Algorithm 1 satisfies the bounds in Table 1 for the given step sizes for all  $\mathbf{w}_1 \in \mathcal{S}$ .

*Proof.* For the sake of simplicity we set  $\text{LIN-OGD}(\alpha) := \text{LIN-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)$ . We start by proving row 1 in Table 1. By Lemma 4.2 we can use Theorem 3.7, Equation (3) with MIX = LIN,  $b = 2$  and  $\mathcal{W} = \mathcal{S}$  where  $|\mathcal{S}|^2 \leq 2$  which gives

$$\alpha = \frac{1}{a} \quad \text{and} \quad \ell(x^n, \text{LIN-OGD}(\alpha)) \leq 2as + 2 \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{LIN}) \quad (8)$$

for any  $s, t_2, \dots, t_s$ . Observe that

$$\ell^*(k, l, \text{LIN}) = \min_{\mathbf{w} \in \mathcal{S}} \ell(x_k^l, \text{LIN}(\mathbf{w}, \mathbf{P}_k^l)) \leq \min_{1 \leq i \leq m} \ell(x_k^l, \text{LIN}(\mathbf{e}_i, \mathbf{P}_k^l)) = \ell^*(k, l, \text{BEST}) \quad (9)$$

and by Lemma 4.2 we can choose

$$a = \frac{17m4^B}{8B} \cdot f(n) \geq \frac{17m}{8} \frac{1}{\varepsilon^2 \log(1/\varepsilon)} \geq m \log^2(e) \frac{p_{\max}^2}{p_{\min}^2 \log(1/p_{\min})}. \quad (10)$$

for some  $f(n) \geq 1$ . We set  $f(n) = 1$  and combine (9) and (10) with (8) to yield

$$\alpha = \frac{8B}{17m4^B} \quad \text{and} \quad \ell(x^n, \text{LIN-OGD}(\alpha)) \leq \frac{17ms4^B}{4B} + 2 \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{BEST}).$$

Finally we can take the minimum over  $s, t_2, \dots, t_s$ , since these were arbitrary, which gives the claim. Now we advance to Table 1, row 2. Again, by Lemma 4.2 we use Theorem 3.7, Equation (4) with MIX = LIN,  $c = -\log \varepsilon = B$  and  $\mathcal{W} = \mathcal{S}$  which gives

$$\alpha = \frac{2/a}{1 + \sqrt{n}} \quad \text{and} \quad \ell(x^n, \text{LIN-OGD}(\alpha)) \leq (2as + B)\sqrt{n} + \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{LIN}) \quad (11)$$

for any  $s, t_2, \dots, t_s$ . We now choose  $a$  as in (10) with  $1 \leq f(n) = \frac{2\sqrt{n}}{1+\sqrt{n}} \leq 2$ , to get

$$(2as + B) = \frac{17ms4^B}{4B} f(n) + B \leq (17ms + 1) \frac{4^B}{2B} \leq \frac{35ms4^B}{4B} \quad (12)$$

for the constant on the r.h.s. of (11). We combine (9) and (12) with (11) to yield

$$\alpha = \frac{8B/\sqrt{n}}{17m4^B} \quad \text{and} \quad \ell(x^n, \text{LIN-OGD}(\alpha)) \leq \frac{35ms4^B}{4B} \sqrt{n} + \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{BEST}).$$

Again, taking the minimum over  $s, t_2, \dots, t_s$  finishes the proof. The bounds of Table 1 rows 3 and 4 follow analogously by the choice of  $f(n) = 1$  (row 3) and  $f(n) = \frac{2\sqrt{n}}{1+\sqrt{n}}$  (row 4) and

$$a = \frac{7mB^2}{10} \cdot f(n) \geq \frac{7m}{10} \log^2 \frac{1}{\varepsilon} \geq \frac{m \log^2(p_{\max}/p_{\min})}{\log e} \quad \text{and by}$$

$$\ell^*(k, l, \text{GEO}) = \min_{\mathbf{w} \in \mathcal{S}} \ell(x_k^l, \text{GEO}(\mathbf{w}, \mathbf{P}_k^l)) \leq \min_{1 \leq i \leq m} \ell(x_k^l, \text{GEO}(\mathbf{e}_i, \mathbf{P}_k^l)) = \ell^*(k, l, \text{BEST})$$

and using  $\ell^*(x^n, \text{GEO}(\mathbf{P}^n)) \leq B \cdot n$  (a premise of Theorem 3.7, Item 2), since for all  $\mathbf{w} \in \mathcal{S}$

$$\text{GEO}(x; \mathbf{w}, \mathbf{P}) = \frac{\prod_{i=1}^m p_i(x)^{w_i}}{\sum_{y \in \mathcal{X}} \prod_{i=1}^m p_i(y)^{w_i}} \geq \prod_{i=1}^m p_i(x)^{w_i} \geq p_{\min}(x; \mathbf{P}) = \varepsilon = 2^{-B} \quad (13)$$

and consequently  $\ell(x, \text{GEO}(\mathbf{w}, \mathbf{P})) \leq B$ .  $\square$

Table 1: Code length bounds of Algorithm 1 for MIX = LIN and MIX = GEO where  $\mathcal{W} = \mathcal{S}$ .

MIX	$\alpha$	$\ell(x^n, \text{MIX-OGD}(\mathbf{w}_1, \alpha, x^n, \mathbf{P}^n)) \leq \min_{\mathcal{S}} t_2, \dots, t_s \text{ of } \dots$
1 LIN	$\frac{8B}{17m4^B}$	$2 \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{BEST}) + \frac{17ms4^B}{4B}$
2	$\frac{8B/\sqrt{n}}{17m4^B}$	$\sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{BEST}) + \frac{35ms4^B}{4B} \sqrt{n}$
3 GEO	$\frac{10}{7mB^2}$	$2 \sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{BEST}) + \frac{7msB^2}{5}$
4	$\frac{10/\sqrt{n}}{7mB^2}$	$\sum_{i=1}^s \ell^*(t_i, t_{i+1} - 1, \text{BEST}) + \frac{19msB^2}{10} \sqrt{n}$

*Remark 4.7.* In the previous proof (13) shows that  $\text{GEO}(x; \mathbf{w}, \mathbf{P}) \geq p_{\min}(x; \mathbf{P})$  when  $\mathbf{w} \in \mathcal{S}$ , just as LIN. Subsequently GEO cannot use more bits than LIN to encode a single symbol in the worst case. In the best case  $\text{GEO}(x; \mathbf{w}, \mathbf{P})$  uses *at most* as much bits as LIN, since

$$\max_{\mathbf{w} \in \mathcal{S}} \text{GEO}(x; \mathbf{w}, \mathbf{P}) \geq \max_{1 \leq i \leq m} \text{GEO}(x; \mathbf{e}_i, \mathbf{P}) = p_{\max}(x; \mathbf{P}) = \max_{\mathbf{w} \in \mathcal{S}} \text{LIN}(x; \mathbf{w}, \mathbf{P}).$$

There exist situations where  $\max_{\mathbf{w} \in \mathcal{S}} \text{GEO}(x; \mathbf{w}, \mathbf{P}) > \max_{\mathbf{w} \in \mathcal{S}} \text{LIN}(x; \mathbf{w}, \mathbf{P})$ , see Example 4.8.

**Example 4.8.** For an alphabet  $\mathcal{X} = \{1, 2, \dots, N\}$ ,  $N > 2$ , we consider  $\text{GEO}(\mathbf{w}, \mathbf{P})$  where  $\mathbf{w} = (1/2 \ 1/2)^\top$  and  $\mathbf{P}^\top = (p_1(x) \ p_2(x))_{x \in \mathcal{X}}$  s.t. for  $0 < \varepsilon, q < 1$  we have

$$p_1(x) := \begin{cases} q & , x = 1 \\ (1-q) \cdot (1-\varepsilon) & , x = 2 \\ \frac{(1-q) \cdot \varepsilon}{N-2} & , \text{otherwise} \end{cases}, \quad p_2(x) := \begin{cases} q & , x = 1 \\ (1-q) \cdot (1-\varepsilon) & , x = 3 \\ \frac{(1-q) \cdot \varepsilon}{N-2} & , \text{otherwise} \end{cases},$$

The mixture probability  $\text{GEO}(1; \mathbf{w}, \mathbf{P})$  of the letter 1 is

$$\frac{p_1(1)^{1/2} \cdot p_2(1)^{1/2}}{\sum_{y \in \mathcal{X}} p_1(y)^{1/2} \cdot p_2(y)^{1/2}} = q / \underbrace{\left[ q + (1-q) \left( 2\sqrt{\frac{\varepsilon(1-\varepsilon)}{N-2}} + \frac{N-3}{N-2}\varepsilon \right) \right]}_{=: f(\varepsilon, N)},$$

We now show, that for any  $q$  there exists an  $\varepsilon$ , such that  $\text{GEO}(1; \mathbf{w}, \mathbf{P}) > p_{\max}(1; \mathbf{P}) = q$ . Clearly, if  $\text{GEO}(1; \mathbf{w}, \mathbf{P}) > q$  we must have  $f(\varepsilon, N) < 1$ . To observe this we bound  $f(\varepsilon, N)$  from above and give a possible choice for  $\varepsilon$ .

$$f(\varepsilon, N) = 2\sqrt{\frac{\varepsilon(1-\varepsilon)}{N-2}} + \frac{N-3}{N-2}\varepsilon \leq 2\sqrt{\frac{\varepsilon}{N-2}} + (N-3)\sqrt{\frac{\varepsilon}{N-2}} = \frac{N-1}{\sqrt{N-2}} \cdot \sqrt{\varepsilon}$$

If we choose  $0 < \varepsilon < (N-2)/(N-1)^2$  it follows that  $f(\varepsilon, N) < 1$  and  $\text{GEO}(1; \mathbf{w}, \mathbf{P}) > q$ .

Note that the bounds in Table 1, rows 3 and 4 only translate to PAQ7 if  $\mathcal{W} = \mathcal{S}$ . To obtain bounds for other weight spaces  $\mathcal{W}$  we only need to substitute the appropriate values for  $|\mathcal{W}|$  and/or  $c > 0$  where  $\ell(x, \text{GEO}(\mathbf{w}, \mathbf{P})) \leq c$  in the previous proof. E.g., if we have  $-r \cdot \mathbf{1} \leq \mathbf{w} \leq r \cdot \mathbf{1}$  for  $r > 0$  then the penalization term of the bound in row 3 increases by a factor of  $|\mathcal{W}|^2/|\mathcal{S}|^2 = mr^2$ .

Veness [9] gave a bound for linear mixtures using a non-OGD weight estimation scheme which is identical to Table 1 row 2 except the penalty term, which is  $O(s \log n)$  in place of  $O(s\sqrt{n})$ . However our analysis is based on Theorem 3.7 which applies to the strictly larger class of nice mixtures with a generic scheme for weight estimation. Clearly, more restrictions can pay off in tighter bounds, consequently we might obtain better bounds by taking advantage of the peculiarities of LIN and GEO.

## 5 Conclusion

In this work we obtained code length guarantees for a particular mixture-based adaptive statistical data compression algorithm. The algorithm of interest combines multiple model distributions via a mixture and employs OGD to adjust the mixture parameters (typically model weights). As a cornerstone we introduced the class of nice mixtures and gave bounds on their code length in the aforementioned algorithm. Since, as we showed, linear and geometric mixtures are nice mixtures we were able to deduce code length guarantees for these two mixtures in the above data compression algorithm. Our results on geometric mixtures directly apply to PAQ7, a special case of geometric mixtures, and provide the first analysis of PAQ7.

We defer an exhaustive experimental study on linear and geometric mixtures to future research. A straightforward extension to Theorem 3.7, Item 2 is to remove the dependence of the step size on the sequence length (which is typically not known in advance). This can be accomplished by using the “doubling-trick” [2] or a decreasing step size [12]. Another interesting topic is whether geometric and/or linear mixtures have disjoint properties, which we can use to yield stronger bounds. This opposes our current approach, which we built on the (common) properties of a nice mixture.

**Acknowledgement.** The author would like to thank Martin Dietzfelbinger, Michael Rink, Sascha Grau and the anonymous reviewers for valuable improvements to this work.

## References

- [1] Nicolò Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. *Journal of Computer and System Sciences*, 59:392–411, 1999.
- [2] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. 1st edition.
- [3] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.
- [4] David P. Helmbold, Jyrki Kivinen, and Manfred K. Warmuth. Relative loss bounds for single neurons. *Proc. IEEE Transactions on Neural Networks*, 10:1291–1304, 1999.
- [5] Manfred Kufleitner, Edgar Binder, and Alexander Fries. Combining Models in Data Compression. In *Proc. Symposium on Information Theory in the Benelux*, volume 30, pages 135–142, 2009.
- [6] Christopher Mattern. Mixing Strategies in Data Compression. In *Proc. Data Compression Conference*, volume 22, pages 337–346, 2012.
- [7] David Salomon and Giovanni Motta. *Handbook of Data Compression*. Springer, 1st edition, 2010.
- [8] Dimitry Shkarin. PPM: one step to practicality. In *Proc. Data Compression Conference*, volume 12, pages 202–211, 2002.

- [9] Joel Veness, Kee Siong Ng, Marcus Hutter, and Michael H. Bowling. Context Tree Switching. In *Proc. Data Compression Conference*, pages 327–336, 2012.
- [10] Paulus Adrianus Jozef Volf. *Weighting Techniques in Data Compression: Theory and Algorithms*. PhD thesis, University of Eindhoven, 2002.
- [11] F. Willems, Yuri M. Shtarkov, and T. J. Tjalkens. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41:653–664, 1995.
- [12] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

## A Proof of Lemma 4.3 and Lemma 4.4

**Lemma 4.3.** *For  $0 < z < 1$  the function  $f(z) := -\frac{\ln z}{1-z}$  satisfies  $f(z) \geq 1$ .*

*Proof.* By the basic inequality  $-\ln(z) \geq 1 - z$  the claim follows.  $\square$

**Lemma 4.4.** *For  $0 < a \leq z \leq 1 - a$  the function  $f(z) := -z^2 \ln z$  satisfies  $f(z) \geq f(a)$ .*

*Proof.* First, we examine the derivative  $f'(z) = -z(1 + 2 \ln z)$  of  $f$ . Clearly,  $f'(z) \geq 0$  for  $0 < z < z_0 := 1/\sqrt{e}$  and  $f'(z) \leq 0$  for  $z_0 \leq z \leq 1$ . From  $a \leq 1 - a$  we conclude that  $a \leq \frac{1}{2}$ . We have  $f(z) \geq \min\{f(a), f(1 - a)\}$  (by monotonicity) and it remains to show that  $f(a) \leq f(1 - a)$ . Let  $g(a) := f(a)/f(1 - a)$  and observe that  $g(a)$  increases monotonically for  $0 < a \leq \frac{1}{2}$ , i.e.  $\frac{f(a)}{f(1-a)} = g(a) \leq g(\frac{1}{2}) = 1$ . Finally we argue that  $g'(a) \geq 0$  where

$$g'(a) = \frac{a \ln(a) \ln(1 - a)}{(a - 1)^3 \ln^2(1 - a)} \cdot \left[ -\frac{a}{\ln(1 - a)} - \frac{1 - a}{\ln a} - 2 \right].$$

Clearly, the left factor is negative for  $0 < a \leq \frac{1}{2}$ . The rightmost factor is at most 0, since by Lemma 4.3 we have  $-\frac{a}{\ln(1-a)} \leq 1/\inf_{0 < z < 1} -\frac{\ln z}{1-z} \leq 1$  (we substituted  $z = 1 - a$ ) and  $-\frac{1-a}{\ln a} \leq 1/\inf_{0 < z < 1} -\frac{\ln z}{1-z} \leq 1$ , which concludes the proof.  $\square$