Equivalence of ML decoding to a continuous optimization problem

Sundara Rajan Srinivasavaradhan, Suhas Diggavi and Christina Fragouli
University of California Los Angeles, CA, USA

***Abstract*—Maximum likelihood (ML) and symbolwise maximum aposteriori (MAP) estimation for discrete input sequences play a central role in a number of applications that arise in communications, information and coding theory. Many instances of these problems are proven to be intractable, for example through reduction to NP-complete integer optimization problems. In this work, we prove that the ML estimation of a discrete input sequence (with no assumptions on the encoder/channel used) is equivalent to the solution of a continuous non-convex optimization problem, and that this formulation is closely related to the computation of symbolwise MAP estimates. This equivalence is particularly useful in situations where a function we term the expected likelihood is efficiently computable. In such situations, we give a ML heuristic and show numerics for sequence estimation over the deletion channel.**

***Index Terms*—Maximum-Likelihood, Symbolwise MAP, Deletion Channels, Expected Likelihood.**

I. Introduction

The problem of estimating an unknown discrete input sequence from its noisy observation arises in many disciplines, including communications, information and coding theory. Fig. 1 represents a typical decoding problem – an input message sequence X must be estimated from the observation or output sequence Y. The input-output relation depends on a multitude of factors such as the exact choice and properties of the encoder, the model and parameters of the noisy channel, and initializations. We capture all these factors together by what we call the *system channel* $\mathcal{C}$. In this work, we are agnostic on what exactly happens inside $\mathcal{C}$, and instead, only assume the knowledge of the input-output relation $\Pr(Y|X,\mathcal{C})$.

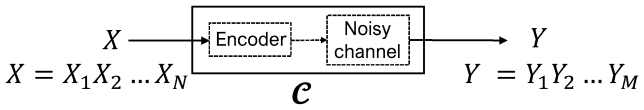


Fig. 1: A generic model of a probabilistic system channel where each $X_i \in \mathcal{A} = \{1, ..., A\}$. The goal is to estimate X given Y and we assume the knowledge of $\Pr(Y|X,\mathcal{C})$.

Central to the above discussion lie two algorithmic problems on optimal decoding – computation of the maximum-likelihood (ML) and the symbolwise maximum-aposteriori (MAP) estimates. The ML problem, in words, is the integer program that maximizes the likelihood $\Pr(Y|X,\mathcal{C})$ over all N-length input sequences X, and the symbolwise MAP problem involves computing the symbolwise posterior probabilities (SPs) $\Pr(X_i{=}a|Y,\mathcal{C})$ for a pre-defined prior distribution on X. Many of these problems are proven to be intractable, for example through reduction to NP-complete integer optimization problems (see [1], [2] for instance).

In this work, we provide continuous variable formulations for the ML and MAP decoding problems for an arbitrary system channel $\mathcal{C}$. In particular, we posit the ML problem as maximization over all product distributions for X. Rather surprisingly, this formulation closely relates to an expression for computing the MAP SPs. Our formulation is particularly useful for system channels where a function termed *expected likelihood function* (that we will define later) can be computed efficiently. Although in full generality this function would be hard to compute, it could still lead to new efficient optimal or heuristic algorithms over new classes of system channels. We believe that at the very least, our observations give a new theoretical perspective on ML and MAP decoding.

Contributions. The main result of our work formulates the ML estimate of an arbitrary system channel as a continuous optimization problem; in particular, we optimize the *expected likelihood function* over the space of product distributions for X, instead of optimizing the actual likelihood. This opens the door to the use of first-order heuristics like gradient ascent. Moreover, we propose an alternate heuristic called *coordinate refinement* for the ML estimate. For the SPs, we give an expression in terms of the expected likelihood and its gradient. As an application, we illustrate performance benefits of our formulations via numerics for the deletion channel.

Related work. Over the past few decades, there has been significant progress towards understanding the complexity of computing optimal ML and MAP estimates (see [1], [3], [2], [4]) as well as towards coming up with efficient algorithms/heuristics (such as Viterbi [5], forward-backward [6], message-passing [7], sphere decoding [8], [9]). These algorithms are tailored to specific classes of system channels; in contrast, our approach applies for all system channels (is agnostic to the encoder and system model).

Decoding a discrete sequence via continuous optimization methods has also been explored in [10] which formulates the ML decoding problem for a linear code over a discrete memoryless channel as a continuous optimization problem,

This work was supported in part by NSF grants 1705077, 1740047 and UC-NL grant LFR-18-548554. Correspondence: sundar@ucla.edu, suhas@ee.ucla.edu, christina.fragouli@ucla.edu.

and proposes a gradient ascent heuristic to solve it[1]. In contrast, our work applies to an arbitrary system channel, and we further we propose a decoding heuristic which empirically performs better than gradient ascent.

Reconstruction over deletion channels without the use of a codebook is closely related to the problem of *trace reconstruction* (see for example [11], [12], [13], [14]). The symbolwise MAP estimate for this case has been solved (see [15]). However little is known about ML estimate in this case (see [16], [17]).

Paper Organization. Section II describes the notation used throughout this paper, Section III contains our main theoretical results, Section IV discusses some algorithmic aspects of our formulation and also gives a ML heuristic, and finally Section V outlines numerics when $\mathcal{C}$ is a deletion channel. We defer some proofs to the longer version of the paper [18].

II. Notation and Tools

Basic notation: Calligraphic letters refer to sets, capitalized letters correspond to either random variables or integer constants (usage will be clear with context), bold letters are used for matrices and greek letters are used to denote functions.

Notation	Definition
$[i:j]$	$\{i, i+1, ..., j\}$ if $j \geq i$ and $[i:j] \triangleq \varnothing$ otherwise
$[i]$	$[1:i]$
$x_{[i:j]}$	$x_i x_{i+1} ... x_j$
$\mathbf{P} \in [0,1]^{N \times A}$	matrix that parametrizes the distribution of an N-length random vector
$\mathbf{P}_i$ for matrix $\mathbf{P}$	i^{th} row of the matrix
$\mathbf{P}_{ij}$ for matrix $\mathbf{P}$	$(i,j)^{th}$ entry of the matrix
$X \sim (\mathbf{P})$	$X = X_{[1:N]}$ and X_i is *independently* distributed and $\Pr(X_i{=}a) = \mathbf{P}_{ia}$
$\mathbf{P} \odot \mathbf{Q}$ for matrices $\mathbf{P}$ and $\mathbf{Q}$	Hadamard product (element wise product)
$\mathbf{cat}(x)$ for sequence x	$\mathbf{cat}(x)$ is an $N \times A$ matrix where $\mathbf{cat}(x)_{ia}{=}1$ if $x_i{=}a$ and $\mathbf{cat}(x)_{ia}{=}0$ otherwise, i.e., $\mathbf{cat}(x)$ is the categorical representation of x.
Matrix $\mathbf{P}$ is a lattice point	$\exists\ x$ such that $\mathbf{cat}(x) = \mathbf{P}$
$\mathbf{cat}^{-1}(\mathbf{P})$	$\mathbf{cat}^{-1}(\mathbf{P}) = x$ if $\mathbf{cat}(x) = \mathbf{P}$.
$\mathbf{P}^{(j \to b)}$	$\mathbf{P}^{(j\to b)}_{ia} = \begin{cases} \mathbf{P}_{ia} & i \neq j \\ \mathbb{1}_{\{a=b\}} & i = j. \end{cases}$ $\mathbf{P}^{(j\to b)}$ modifies only the j^{th} row of $\mathbf{P}$ to be a unit vector where $\mathbf{P}_{jb} = 1$.

TABLE I: Table of common notation.

ML estimate. For a system channel $\mathcal{C}$ as in Fig. 1 where $X_i \in \mathcal{A} = \{1, 2, ..., A\}$, the ML estimate of X given observation Y is the integer program:

$$x^*_{ml} \triangleq \underset{x \in \mathcal{A}^N}{\arg\max}\ \Pr(Y|X = x, \mathcal{C}). \tag{1}$$

Note that there could be multiple optimal solutions to (1) and in such cases, it suffices to obtain just one such solution.

[1]We were not aware of [10] when we did the work, but was pointed out to us during the review process.

SPs and symbolwise MAP. For the system channel $\mathcal{C}$ in Fig. 1 where $X_i \in \mathcal{A} = \{1, 2, ..., A\}$, let the prior input distribution be $X \sim (\mathbf{P})$. The SPs can be collected in the matrix $\mathbf{P}^{\text{post}}$:

$$\mathbf{P}^{\text{post}}_{ia} = \Pr(X_i = a | Y, \mathcal{C}).$$

Note that $\mathbf{P}^{\text{post}}$ varies with both $\mathbf{P}$ as well as Y. The SPs give a convenient way of estimating X by picking the most likely symbol at each position (the symbolwise MAP estimate).

Definition 1. Expected likelihood function. For the system channel model in Fig. 1, given an observation Y and a distribution matrix $\mathbf{P}$, we define the expected likelihood function as the expectation of the likelihood of observing Y w.r.t the distribution $X \sim (\mathbf{P})$, i.e.,

$$\lambda(\mathbf{P}, Y; \mathcal{C}) \triangleq \underset{X \sim (\mathbf{P})}{\mathbb{E}} \Pr(Y|X, \mathcal{C}). \tag{2}$$

Some properties of the expected likelihood function are:

- $0 \leq \lambda(\mathbf{P}, Y; \mathcal{C}) \leq 1$, since it is an expectation of the likelihood.
- For a lattice point $\mathbf{P}$, $\lambda(\mathbf{P}, Y; \mathcal{C}) = \Pr(Y|X{=}\mathbf{cat}^{-1}(\mathbf{P}), \mathcal{C})$.

III. ML and SPs via expected likelihood function

In this section, we discuss our ML and MAP SPs formulations through the lens of the expected likelihood function.

A. ML via expected likelihood

Theorem 1. *Consider a system channel $\mathcal{C}$ as in Fig. 1. Assume that $X_i \in \mathcal{A} = \{1, 2, ..., A\}$. The ML estimate in (1) is equivalent to solving the following continuous optimization:*

$$\begin{aligned} \underset{\mathbf{P} \in \mathbb{R}^{N \times A}}{\arg\max} \quad & \lambda(\mathbf{P}, Y; \mathcal{C}) \\ \text{s.t.} \quad & \mathbf{P} \cdot \mathbf{1} = \mathbf{1} \\ & \mathbf{0} \leq \mathbf{P}. \end{aligned} \tag{3}$$

$\mathbf{P} \cdot \mathbf{1}$ represents the matrix product of $\mathbf{P}$ with the all ones vector $\mathbf{1}$, and " $\leq$ " represents component-wise inequality.

Proof. The idea behind the proof is that instead of optimizing over all possible choices for X, we optimize over all possible product distributions for X. Recall that $\lambda(\mathbf{P}, Y; \mathcal{C}) \triangleq \mathbb{E}_{X \sim (\mathbf{P})} \Pr(Y|X, \mathcal{C})$. We prove the theorem by proving the following three claims:

1) For every feasible $\mathbf{P}$,
$$\lambda(\mathbf{P}, Y; \mathcal{C}) \leq \max_{x \in \mathcal{A}^N} \Pr(Y|X = x, \mathcal{C}).$$
2) Given a solution x^*_{ml} of (1), there exists $\mathbf{P}^*$ such that
$$\begin{aligned} \lambda(\mathbf{P}^*, Y; \mathcal{C}) &= \max_{x \in \mathcal{A}^N} \Pr(Y|X = x, \mathcal{C}) \\ &= \max_{\mathbf{P} \in \mathbb{R}^{N \times A}} \lambda(\mathbf{P}, Y; \mathcal{C}). \end{aligned}$$
3) Consider a $\mathbf{P}^*$ which maximizes $\lambda(\mathbf{P}, Y; \mathcal{C})$. Sample an X from $X \sim (\mathbf{P}^*)$, then X is a solution of (1).

Claims 1) and 2) together prove that the maximum objective values of (1) and (3) are equal, claim 2) also gives a way of obtaining a solution of (3) from a solution of (1), and claim 3) gives a way of obtaining a solution of (1) from (3).

Claim 1) is easily seen by observing that $\lambda(\mathbf{P}, Y; \mathcal{C})$ is the expectation of $\Pr(Y|X, \mathcal{C})$ w.r.t to a distribution on X defined over the set $\mathcal{A}^N$. Clearly $\lambda(\mathbf{P}, Y; \mathcal{C})$ must not exceed the maximum value taken by $\Pr(Y|X, \mathcal{C})$ over $\mathcal{A}^N$.

Claim 2) can be seen by taking $\mathbf{P}^* = \mathbf{cat}(x^*_{ml})$ i.e.,

$$\lambda(\mathbf{cat}(x^*_{ml}), Y; \mathcal{C}) = \Pr(Y|X = x^*_{ml}, \mathcal{C}).$$

To prove claim 3) we first note that $\lambda(\mathbf{P}^*, Y; \mathcal{C}) = \Pr(Y|X{=}x^*_{ml})$ from claims 1) and 2). But $\lambda(\mathbf{P}^*, Y; \mathcal{C})$ is also the expectation of $\Pr(Y|X, \mathcal{C})$ over $X \sim (\mathbf{P}^*)$. Since $\Pr(Y|X{=}x, \mathcal{C}) \leq \Pr(Y|X{=}x^*_{ml}, \mathcal{C})\ \forall x$, we have that for every x such that $\Pr(Y|X{=}x, \mathcal{C}) > 0$ w.r.t to $X \sim (\mathbf{P}^*)$, $\Pr(Y|X{=}x, \mathcal{C}){=}\Pr(Y|X{=}x^*_{ml}, \mathcal{C})$. □

We remark that the formulation in (3) falls under the umbrella of signomial optimization problems (see [19], [20] and references therein) which are, in general, hard to solve. Typical heuristic approaches to such problems involve convexification strategies that instead solve a series of related convex programs. However, such strategies would in general fail for (3) since, with a change of variables, (3) can be written as minimization of a concave function over a convex set.

B. SPs via expected likelihood

Recall that the SPs for the model in Fig. 1 can be collected in the matrix $\mathbf{P}^{\text{post}}$ where $\mathbf{P}^{\text{post}}_{ia} = \Pr(X_i = a|Y, \mathcal{C})$. We first state some results that will be used to prove Theorem 2 and for the heuristic in Section IV.

Lemma 1. *Consider Fig. 1 and let the prior input distribution be $X \sim (\mathbf{P})$. Then,*

$$\sum_{x:x_i=a} \Pr(x) \Pr(Y|X = x, \mathcal{C}) = \mathbf{P}_{ia} \lambda(\mathbf{P}^{(i\to a)}, Y; \mathcal{C}).$$

The proof follows from the definition of expected likelihood and can be found in [18]. The following two corollaries are easily seen from the definition of $\lambda(\cdot)$ and Lemma 1.

Corollary 1. $$\lambda(\mathbf{P}, Y; \mathcal{C}) = \sum_{a=1}^{A} \mathbf{P}_{ia} \lambda(\mathbf{P}^{(i\to a)}, Y; \mathcal{C}). \quad (4)$$

Corollary 2. $$\frac{\partial}{\partial \mathbf{P}_{ia}} \lambda(\mathbf{P}, Y; \mathcal{C}) = \lambda(\mathbf{P}^{(i\to a)}, Y; \mathcal{C}). \quad (5)$$

(4) indicates an important property about the geometry of the expected likelihood function – it is linear in each $\mathbf{P}_i$ (however it is not linear in $\mathbf{P}$) and (5) relates $\lambda(\cdot)$ and its gradient.

Theorem 2. *In Fig. 1, let the prior distribution be $X \sim (\mathbf{P})$. Then the SPs $\mathbf{P}^{post}$ can be written as:*

$$\mathbf{P}^{post}_{ia} = \mathbf{P}_{ia} \frac{\lambda(\mathbf{P}^{(i\to a)}, Y; \mathcal{C})}{\lambda(\mathbf{P}, Y; \mathcal{C})} = \mathbf{P}_{ia} \frac{\frac{\partial}{\partial \mathbf{P}_{ia}} \lambda(\mathbf{P}, Y; \mathcal{C})}{\lambda(\mathbf{P}, Y; \mathcal{C})}. \quad (6)$$

Alternatively a matrix formulation for the SPs is,

$$\mathbf{P}^{post} = \frac{\mathbf{P} \odot \nabla_{\mathbf{P}} \lambda(\mathbf{P}, Y; \mathcal{C})}{\lambda(\mathbf{P}, Y; \mathcal{C})}. \quad (7)$$

Proof. First we note that the SPs can be written as,

$$\mathbf{P}^{\text{post}}_{ia} = \frac{\Pr(X_i = a, Y|\mathcal{C})}{\Pr(Y|\mathcal{C})}$$

$$= \frac{1}{\lambda(\mathbf{P}, Y; \mathcal{C})} \sum_{x:x_i=a} \Pr(x) \Pr(Y|X = x, \mathcal{C}). \quad (8)$$

Using Lemma 1 and (5) with (8) concludes the proof. □

IV. Coordinate refinement: A global ML heuristic based on expected likelihood

In this section, we propose a heuristic for ML based on our theoretical observations in Section III. But first, we discuss some classes of system channels where our expected-likelihood formulation of ML and MAP SPs can be useful.

A. Algorithmic aspects of expected likelihood

Clearly, Theorem 1 and Theorem 2 are directly applicable for system channels where the expected likelihood can be computed efficiently. For such cases, we observe the following:

- First we observe that (5) implies that computing the gradient of $\lambda(\cdot)$ amounts to NA computations of $\lambda(\cdot)$. However, in many cases it might be possible to compute the gradients directly and faster (we do this for deletion channels). Moreover, Theorem 2 signifies that in such cases, the MAP SPs can be computed in polynomial time.
- Existence of a polynomial time algorithm to compute $\lambda(\cdot)$ does not necessarily imply that the ML problem in (3) is solvable in polynomial time. However, what it indicates is that heuristics for continuous optimization can be employed for (3).
- A natural first-order heuristic for (3) is *projected gradient ascent*, which is a variant of gradient ascent for maximization with constraints. In our case the constraint is that $\mathbf{P}$ must be a valid distribution matrix, i.e., $\mathbf{P}$ lies in the polytope

$$\mathcal{D} \triangleq \left\{\mathbf{Q} : \mathbf{Q} \in [0, 1]^{N\times A} \text{ and } \mathbf{Q} \cdot \mathbf{1} = \mathbf{1}\right\}.$$

  In projected gradient ascent, at each update step, the updated point is projected back onto $\mathcal{D}$ by finding a point in $\mathcal{D}$ closest to $\mathbf{P}$, i.e, the update step is

$$\mathbf{P} \leftarrow \arg\min_{\mathbf{Q}\in\mathcal{D}} \left|\left|\mathbf{Q} - (\mathbf{P} + \epsilon.\nabla_{\mathbf{P}} \lambda(\mathbf{P}, Y; \mathcal{C}))\right|\right|^2. \quad (9)$$

We now comment on the complexity of computing the expected likelihood for a few examples of system channel $\mathcal{C}$ in Fig. 1. We refer the reader to [18] for a more detailed discussion on each of the following situations.

Discrete memoryless channel (DMC): When $\mathcal{C}$ is a DMC, $\lambda(\mathbf{P}, Y; \mathcal{C})$ breaks down into a product of N terms and can be computed in $O(NA)$. We clarify here that this case does not subsume the situation of having an encoder before a DMC. Note that the ML formulation in Theorem 1 also breaks down into N smaller problems each of which can be solved efficiently. This can also be proved to be equivalent to the symbolwise MAP estimate.

Probabilistic finite state machine (FSM): Say $\mathcal{C}$ is an FSM with states in $\mathcal{S}$ which outputs exactly K symbols corresponding to each input symbol. Then the expected likelihood can be computed using a dynamic programming approach in $O(A|\mathcal{S}|^2N)$. We note that this complexity is of the same order

as the complexity of computing the ML via Viterbi algorithm or the symbolwise MAP via Forward-Backward algorithm. The class of probabilistic FSM system channels encompasses a variety of situations. For example, a convolutional encoder followed by a DMC can be represented by such a model.

Deletion channel: Assume that $\mathcal{C}$ deletes each input symbol with a constant probability of deletion δ (the encoder is an identity encoder). We prove that a dynamic program can be used to compute $\lambda(\cdot)$ in $O(NM)$ (see [18]).

Sticky channel: A sticky channel with parameter p repeats each input symbol $K \in \{1, 2, ...\}$ times with a probability $(1-p)^{K-1}p$. Recent works have looked at the capacity of sticky channels [21]. We can prove that a dynamic program can compute $\lambda(\cdot)$ in $O(NM^2)$ time complexity (see [18]).

B. Coordinate refinement

In situations where the expected likelihood function and its gradient are efficiently computable, we give a heuristic for the ML formulation in Theorem 1. This algorithm exploits the linearity of $\lambda(\mathbf{P}, Y; \mathcal{C})$ when projected on to the coordinates $\mathbf{P}_i$ (4). The basic underlying idea is as follows (the exact algorithm is detailed in Alg. 1):

- Say we start with a distribution matrix $\mathbf{P} \in [0,1]^{N \times A}$.
- We iterate over the indices $[1{:}N]$ (the rows of $\mathbf{P}$) in a specified order (here we do so greedily). In the iteration corresponding to index i, we update row $\mathbf{P}_i$ such that the value of $\lambda(\cdot)$ never decreases.
- This update is done by comparing $\lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C})\ \forall\ a$ and picking the a which maximizes $\lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C})$, i.e.,

$$\mathbf{P} \leftarrow \mathbf{P}^{(i \to \arg\max_a \lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C}))}. \tag{10}$$

Note that $\forall\ i,\ \exists\ a$ such that $\lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C}) \geq \lambda(\mathbf{P}, Y; \mathcal{C})$ due to (4), thus ensuring that the update step never decreases $\lambda(\cdot)$. Further, (5) signifies that $\nabla_{\mathbf{P}} \lambda(\mathbf{P}, Y; \mathcal{C})$ computes $\lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C})\ \forall\ i, a$.

Iterating over the indices $[1{:}N]$ once amounts to one *refinement iteration*. At the end of a refinement iteration, the final $\mathbf{P}$ is a lattice point (since every row has been updated to a unit vector). A new refinement iteration can now be started using current distribution $\mathbf{P}$ to further improve $\lambda(\cdot)$. Note that once we reach a lattice point, every update step results in a distribution which is also lattice point. Since the number of lattice points are finite, there will arise a situation where the update stagnates (does not strictly increase $\lambda(\cdot)$). In that case, we have arrived at a *fixed point* of this algorithm and we stop.

Before moving further we first define a *fixed point* of an update algorithm. An update algorithm takes as input a distribution $\mathbf{P}$ and updates it iteratively. The projected gradient ascent and coordinate refinement are both update algorithms.

Definition 2. $\mathbf{P}^{\text{fixed}}$ is a *fixed point* of an update algorithm if the update step applied on $\mathbf{P}^{\text{fixed}}$ does not change $\mathbf{P}^{\text{fixed}}$.

$\mathbf{P}^{\text{fixed}}$ is a fixed point of the projected gradient ascent if the right-hand side of (9) is equal to $\mathbf{P}^{\text{fixed}}$ itself and it is a fixed point of coordinate refinement if the right-hand side of (10) is $\mathbf{P}^{\text{fixed}}$ itself. More precisely,

Algorithm 1 Greedy coordinate refinement

Inputs: Distribution $\mathbf{P}^{\text{init}}$, Observation Y, Algorithms to query $\lambda(\mathbf{P}, Y; \mathcal{C})$ and $\nabla_{\mathbf{P}} \lambda(\mathbf{P}, Y; \mathcal{C})$, Max refinement iterations RF_{max}
Outputs: Estimate $\hat{X}$
Initialize $\mathbf{P} = \mathbf{P}^{\text{init}}$
for *iter* in $[1 : RF_{max}]$ **do**
  Initialize visited indices $\mathcal{I} = \varnothing$
  while $|\mathcal{I}| < N$ **do**
    Compute gradient matrix $\mathbf{G} = \nabla_{\mathbf{P}} \lambda(\mathbf{P}, Y; \mathcal{C})$
    if $\mathbf{P}$ is a lattice point and satisfies (12) **then**
      return $\mathbf{cat}^{-1}(\mathbf{P})$ and **exit**
    $(i^*, a^*) = \underset{i \in [N] \setminus \mathcal{I},\ a \in \mathcal{A}}{\arg\max} \mathbf{G}_{ia}$
    Update $\mathbf{P} \leftarrow \mathbf{P}^{(i^* \to a^*)}$
    Update $\mathcal{I} \leftarrow \mathcal{I} \cup \{i^*\}$
return $\mathbf{cat}^{-1}(\mathbf{P})$

- $\mathbf{P}$ is a fixed point for project gradient descent iff

$$\mathbf{P} = \underset{\mathbf{Q} \in \mathcal{D}}{\arg\min} \left|\left| \mathbf{Q} - (\mathbf{P} + \epsilon\ \nabla_{\mathbf{P}} \lambda(\mathbf{P}, Y; \mathcal{C})) \right|\right|^2 \forall\ \epsilon > 0, \tag{11}$$

where $\mathcal{D} = \left\{ \mathbf{Q} : \mathbf{Q} \in [0,1]^{N \times A} \text{ and } \mathbf{Q} \cdot \mathbf{1} = \mathbf{1} \right\}$.

- $\mathbf{P}$ is a fixed point for coordinate refinement iff

$$\mathbf{cat}^{-1}(\mathbf{P})_i = \underset{a \in \mathcal{A}}{\arg\max}\ \lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C}) \quad \forall\ i. \tag{12}$$

We do note that there could be multiple solutions to $\arg\max_{a \in \mathcal{A}}\ \lambda(\mathbf{P}^{(i \to a)}, Y; \mathcal{C})$, but we choose to stop coordinate refinement at $\mathbf{P}^{\text{fixed}}$ instead. Although coordinate refinement reaches a fixed point after a finite number of refinement iterations, this number could potentially be exponential in N. However in practice, for the deletion channel, the coordinate refinement reached a fixed point mostly within 3 refinement iterations even for N=100. Further, we give an interesting result about such fixed points.

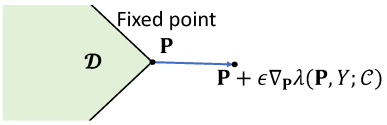
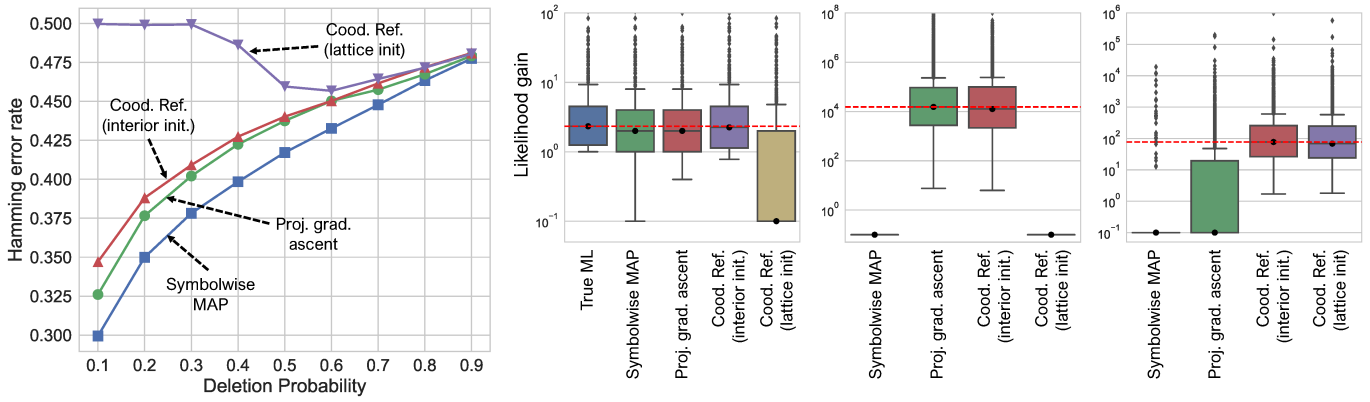


Fig. 3: Figure illustrating the idea behind Theorem 3.

Theorem 3. *If the distribution $\mathbf{P}$ is a fixed point for coordinate refinement (given Y), then $\mathbf{P}$ is also a fixed point for projected gradient ascent.*

The proof of the theorem can be found in [18]. The idea behind the proof is that the gradient at a fixed point $\mathbf{P}$ extends outwardly from $\mathbf{P}$ such that any point lying outside $\mathcal{D}$ in the direction of the gradient is closer to $\mathbf{P}$ than every other point in $\mathcal{D}$ (see Fig. 3). Thus the result of the projection onto $\mathcal{D}$ is again $\mathbf{P}$.

Note on initializations for coordinate refinement: A natural question is if it makes a difference initializing $\mathbf{P}$ as an interior



(a) Hamming error rates for blocklength $N = 100$.

(b) Likelihood gains for $N = 10$, $\delta = 0.2$.

(c) Likelihood gains for $N = 100$, $\delta = 0.2$.

(d) Likelihood gains for $N = 100$, $\delta = 0.8$.

Fig. 2: Hamming error rates and likelihood gains for coordinate refinement (with both vertex and interior point initializations), symbolwise MAP, and projected gradient ascent. We compare for various blocklengths and deletion probabilities. We use box plots to visualize the sample distribution of the likelihood gains. The ends of the boxes indicate the upper and lower quartiles, the dot in each box is the median of the samples, the whiskers indicate the extrema and the diamonds are deemed as outlier samples. We note here that we enforce a lower cap for the likelihood gain at 0.1 to aid log domain visualization.

point ($\mathbf{P}_{ia} \in (0,1)$) or a lattice point. For an interior point, the first refinement iteration updates $\mathbf{P}$ to a lattice point and subsequent refinement iterations deal with $\mathbf{P}$ in the set of lattice points thereon, in which case we are optimizing $\Pr(Y|X,\mathcal{C})$ directly. One could have initialized $\mathbf{P}$ to be a lattice point to begin with and optimize $\Pr(Y|X,\mathcal{C})$, circumventing the use of expected likelihood: numerical evaluation in the next section indicates that such an initialization can significantly deteriorate the performance of coordinate refinement.

V. Example Application: Numerical Results for the Deletion Channel

As an application, we focus on the deletion channel and show numerical results for the various algorithms which exploit our ML and SPs formulation. We restrict ourselves to the binary alphabet $\mathcal{A} = \{0,1\}$ for simplicity. As mentioned in Section IV, and detailed in [18], the expected likelihood for the deletion channel can be computed in $O(NM)$. A similar dynamic programming approach can be employed to compute its gradient in $O(NM)$ as well (we omit the details in lieu of space). Our comparisons are based on two metrics:

- The likelihood gain $\gamma(x,\hat{X}) = \frac{\Pr(Y|X=\hat{X},\mathcal{C})}{\Pr(Y|X=x,\mathcal{C})}$ where x is the actual input and $\hat{X}$ is the estimate. The true ML sequence gives the optimal (largest) likelihood gain.
- The hamming error rate $\psi(x,\hat{X})$ which is defined to be number of bit errors between the actual input x and estimate $\hat{X}$ divided by its blocklength. The symbolwise MAP is an optimal estimator for the hamming error rate. Note that, in general, optimizing for hamming error rate is not equivalent to optimizing for the likelihood gain and vice-versa.

We compare the performance of the following algorithms:

- *Symbolwise MAP*: we first compute the SPs via Theorem 2 and then pick the most likely symbol for each position.
- *Projected gradient ascent:* as defined by (9). At distribution $\mathbf{P}$, we use an adaptive step size $\epsilon = \frac{0.1}{\lambda(\mathbf{P},Y;\mathcal{C})}$ and allow a maximum of 200 update steps.
- *Coordinate refinement with interior point initialization:* We use Alg. 1 with $\mathbf{P}^{\text{init}}$ whose entries are all 0.5, i.e., they correspond to the uniform distribution.
- *Coordinate refinement with lattice point initialization:* We use Alg. 1 and initialize $\mathbf{P}^{\text{init}}$ as a random lattice point.

Observations in Fig. 2.

- Symbolwise MAP has the least hamming error rate as it is an optimal estimator for this error metric. However, it has poor likelihood gains. The reasoning is very specific to the nature of deletion channels – changing just a few bits could vastly affect the likelihoods of the corresponding sequences. For instance, consider N=5 and an observation Y=001. It is easily seen that input sequence X=00001 corresponds to a high likelihood while X=00000 corresponds to 0 likelihood although it differs by only one bit.
- Coordinate refinement with lattice point initialization is seen to perform much worse than coordinate refinement with interior point initialization in all cases, which supports the usefulness of the relaxation provided by Theorem 1.
- Coordinate refinement with interior point initialization has consistently good likelihood gain performance across deletion probabilities unlike the other algorithms. One intuitive explanation is that it can be envisioned as a two step process: 1) in the first refinement iteration, the algorithm performs a coarse search (via the gradient values) and finds a "good" initial lattice point distribution for subsequent refinement iterations 2) subsequent refinement iterations finely "refine" the symbols to further improve the quality of the solution. The projected gradient ascent is lacking of step 2) while coordinate refinement with lattice point initialization lacks step 1).

REFERENCES

[1] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Trans. Inf. Theor.*, vol. 24, no. 3, p. 384–386, Sep. 2006.

[2] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of reed–solomon codes is np-hard," *Information Theory, IEEE Transactions on*, vol. 51, pp. 2249– 2256, 08 2005.

[3] A. Vardy, "Algorithmic complexity in coding theory and the minimum distance problem," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, ser. STOC '97. New York, NY, USA: Association for Computing Machinery, 1997, p. 92–109.

[4] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to gps," *IEEE Transactions on signal processing*, vol. 46, no. 11, pp. 2938–2952, 1998.

[5] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on information theory*, vol. 20, no. 2, pp. 284–287, 1974.

[7] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[8] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of computation*, vol. 44, no. 170, pp. 463–471, 1985.

[9] B. Hassibi and H. Vikalo, "On the expected complexity of integer least-squares problems," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 2002, pp. II–1497.

[10] K. Farrell, L. Rudolph, C. Hartmann, and L. Nielsen, "Decoding by local optimization (corresp.)," *IEEE transactions on information theory*, vol. 29, no. 5, pp. 740–743, 1983.

[11] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *SODA '04*, 2004, pp. 910–918.

[12] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results," in *ACM-SIAM SODA '08*, 2008, pp. 389–398.

[13] A. De, R. O'Donnell, and R. A. Servedio, "Optimal mean-based algorithms for trace reconstruction," in *STOC 2017*.

[14] N. Holden, R. Pemantle, and Y. Peres, "Subpolynomial trace reconstruction for random strings and arbitrary deletion probability," *Proceedings of Machine Learning Research*, 2018.

[15] S. R. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "Symbol-wise map for multiple deletion channels," in *2019 IEEE International Symposium on Information Theory (ISIT)*.

[16] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, vol. 6, pp. 1–33, 2009.

[17] S. R. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "On maximum likelihood reconstruction over multiple deletion channels," in *2018 IEEE International Symposium on Information Theory (ISIT)*.

[18] S. R. Srinivasavaradhan, S. Diggavi, and C. Fragouli, "Equivalence of ml decoding to a continuous optimization problem." [Online]. Available: http://arni.ee.ucla.edu/_media/group/isit2020_sundar.pdf

[19] G. Xu, "Global optimization of signomial geometric programming problems," *European Journal of Operational Research*, vol. 233, no. 3, pp. 500 – 510, 2014.

[20] V. Chandrasekaran and P. Shah, "Relative entropy relaxations for signomial optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1147–1173, 2016.

[21] M. Cheraghchi and J. Ribeiro, "Sharp analytical capacity upper bounds for sticky and related channels," *IEEE Transactions on Information Theory*, 2019.