Coding in the Presence of Semantic Value of Information: Unequal Error Protection Using Poset Decoders

Marcelo Firer

Joint work with
Luciano Panek - UEM
Laura L. Ramos Rifo - UNICAMP

Workshop on Algebraic Coding Theory
CIB - EPFL
Basic question of Coding Theory

Consider an $\mathbb{F}_q^n$-linear code $C \subseteq \mathbb{F}_q^n$ defined by the conditional probabilities $P(y|x) = \prod_{i=1}^{n} P(x_i|y_i)$ satisfying $x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_n) \in \mathbb{F}_q^n$. $P(c)$ is the probability that a message $c \in C$ is sent.

A decoder for $C$ is a map $a : \mathbb{F}_q^n \rightarrow C$. 

Marcelo Firer (IMECC - UNICAMP)
September 2011
Consider

- A \([n, k]_q\) linear code \(C \subseteq \mathbb{F}_q^n\)
Consider

- A \([n, k]_q\) linear code \(C \subseteq \mathbb{F}_q^n\)
- A discrete channel defined by the conditional probabilities

\[
P(y|x) = \prod_{i=1}^{n} P(x_i|y_i)
\]

satisfying \(x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_n) \in \mathbb{F}_q^n\)
Consider

- A \([n, k]_q\) linear code \(C \subseteq \mathbb{F}_q^n\)
- A discrete channel defined by the conditional probabilities

\[
P(y|x) = \prod_{i=1}^{n} P(x_i|y_i)
\]

satisfying \(x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_n) \in \mathbb{F}_q^n\)

- \(P(c)\) is the probability that a message \(c \in C\) is sent
Consider

- A \([n, k]_q\) linear code \(C \subseteq \mathbb{F}_q^n\)
- A discrete channel defined by the conditional probabilities

\[
P(y|x) = \prod_{i=1}^{n} P(x_i|y_i)
\]

satisfying \(x = (x_1, \cdots, x_n), y = (y_1, \cdots, y_n) \in \mathbb{F}_q^n\)

- \(P(c)\) is the probability that a message \(c \in C\) is sent

A *decoder* for \(C\) is a map

\[
a : \mathbb{F}_q^n \longrightarrow C
\]
The error probability of the pair \((C, a)\) is

\[
P_e (C, a) = \sum_{c \in C} \sum_{y \in \mathbb{F}_q^n} \delta_{a(y), c} P(y|c) P(c)
\]

where

\[
\delta_{a(y), c} = \begin{cases} 
0 & \text{if } a(y) - c = 0 \\
1 & \text{if } a(y) - c \neq 0
\end{cases}
\]
The error probability of the pair \( (C, a) \) is

\[
P_e (C, a) = \sum_{c \in C} \sum_{y \in \mathbb{F}_q^n} \delta_{a(y),c} P (y|c) P (c)
\]

where

\[
\delta_{a(y),c} = \begin{cases} 
0 & \text{if } a(y) - c = 0 \\
1 & \text{if } a(y) - c \neq 0 
\end{cases}
\]

**Main question of Coding Theory:** Given \( n \) and \( k \), find a pair \( (C, a) \) that minimizes the error probability function \( P_e (C, a) \) where \( C \) is a \( [n, k]_q \) code and \( a \) a decoder for \( C \).
Basic question of Error Correcting Codes Theory

Important Remarks

Given $C$, if $P(c)$ is constant on $C$, a decoder $a$ that minimizes $P_e(C,a)$ is necessarily a Maximum Likelyhood (ML) Decoder or equivalently, a Nearest Neighbour (NN) Decoder.

The space of configurations of this optimization problem is $\{[n,k]_q\text{-codes}\} \times \{\text{ML decoders}\}$.

Assuming $C$ linear, $a(y) - c \in C$, so that errors in decoding are also code words.
Important Remarks

- Given $C$, if $P(c)$ is constant on $C$, a decoder $a$ that minimizes $P_e(C, a)$ is necessarily a **Maximum Likelyhood** (ML) Decoder or equivalently, a **Nearest Neighbour** (NN) Decoder.

- The space of configurations of this optimization problem is

$$\left\{ [n, k]_q \text{-codes} \right\} \times \{ \text{ML decoders} \}$$
Basic question of Error Correcting Codes Theory

Important Remarks

- Given $C$, if $P(c)$ is constant on $C$, a decoder $a$ that minimizes $P_e(C, a)$ is necessarily a **Maximum Likelyhood** (ML) Decoder or equivalently, a **Nearest Neighbour** (NN) Decoder.

- The space of configurations of this optimization problem is
  
  $$\left\{ [n, k]_q \text{-codes} \right\} \times \{ \text{ML decoders} \}$$

- Assuming $C$ linear, $a(y) - c \in C$, so that errors in decoding are also code words
We are concerned with existence of errors and not the type of errors: if we send a message \( c = 11111111110000000000000001111111111 \) it will be the same if the decoded code word will be either \( c_1 = 00000000000111111111100000000000 \), \( c_2 = 1111111111000000000001111111110 \), or \( c_3 = 0111111111000000000001111111111 \).
Basic question of Coding Theory

Important Remarks (cont.)

We are concerned with existence of errors and not the type of errors: if we send a message

\[ c = 111111111100000000001111111111 \]

it will be the same if the decoded code word will be either
Important Remarks (cont.)

We are concerned with existence of errors and not the type of errors: if we send a message

\[ c = 11111111100000000001111111111 \]

it will be the same if the decoded code word will be either

\[ c_1 = 0000000000011111111111000000000000 \]
Important Remarks (cont.)

We are concerned with existence of errors and not the type of errors: if we send a message

\[ c = 111111111100000000001111111111 \]

it will be the same if the decoded code word will be either

\[ c_1 = 00000000000111111111100000000000, \]

\[ c_2 = 1111111111000000000011111111110, \]
Important Remarks (cont.)

We are concerned with existence of errors and not the type of errors: if we send a message

$$c = \text{11111111110000000001111111111}$$

it will be the same if the decoded code word will be either

$$c_1 = 00000000000111111111100000000000,$$

$$c_2 = 111111111100000000001111111110,$$

or

$$c_3 = 011111111100000000001111111111,$$
Important Remarks (cont.)

We are concerned with existence of errors and not the type of errors: if we send a message

\[ c = 111111111100000000001111111111 \]

it will be the same if the decoded code word will be either

\[ c_1 = 0000000000011111111111000000000000, \]

\[ c_2 = 11111111111000000000001111111110, \]

or

\[ c_3 = 01111111111000000000001111111111, \]

\[ \delta_{c,c_1} = \delta_{c,c_2} = \delta_{c,c_3} = 1 \]
Our aim: what can be done when we can evaluate the semantic value of the errors?

Example

Concerning my salary, errors in the cents are less important than errors in the thousands.

Slight differences in colors in a picture: exchanging...
Our aim: what can be done when we can evaluate the semantic value of the errors?
Our aim: what can be done when we can evaluate the semantic value of the errors?

Example

- Concerning my salary, errors in the cents are less important than errors in the thousands
Our aim: what can be done when we can evaluate the semantic value of the errors?

Example

- Concerning my salary, errors in the cents are less important than errors in the thousands
- Slight differences in colors in a picture: exchanging
Semantic Value of information

Our aim: what can be done when we can evaluate the semantic value of the errors?

Example

- Concerning my salary, errors in the cents are less important than errors in the thousands.
- Slight differences in colors in a picture: exchanging

Figure: Original colors in the middle column
Semantic Value of information

Definition

A value function defined on a code $C$ is just a function $\mu : C \rightarrow \mathbb{R}^+$. 

Remark

$\mu$ depends on the way information is placed inside a code. Considering the set of informations $F_{kq}$, an embedding $i : F_{kq} \rightarrow C$ and $\tilde{\mu} : F_{kq} \rightarrow \mathbb{R}^+$, we have that $\mu \circ i = \tilde{\mu}$. 

$F_{kq} i \downarrow \downarrow \tilde{\mu} \rightarrow \rightarrow R^+$

$C \subseteq F_{nk}$

$\tilde{\mu} = \mu \circ i \uparrow \uparrow$
A value function defined on a code $C$ is just a function $\mu : C \rightarrow \mathbb{R}^+$. 

Remark

$\mu$ depends on the way information is placed inside a code. Considering the set of informations $\mathbb{F}_q^k$, an embending $i : \mathbb{F}_q^k \rightarrow C$ and $\tilde{\mu} : \mathbb{F}_q^k \rightarrow \mathbb{R}^+$, we have that $\mu \circ i = \tilde{\mu}$. 

\[
\begin{tikzcd}
\mathbb{F}_q^k \ar[rr, \tilde{\mu}] \ar[d, i] & & \mathbb{R}^+ \\
C \ar[u, \mu \circ i] & & \\
\mathbb{F}_q^n & & 
\end{tikzcd}
\]
Semantic Value of information

**Definition**

A value function defined on a code $C$ is just a function $\mu : C \rightarrow \mathbb{R}^+$. 

**Remark**

$\mu$ depends on the way information is placed inside a code. Considering the set of informations $\mathbb{F}_q^k$, an embedding $i : \mathbb{F}_q^k \rightarrow C$ and $\tilde{\mu} : \mathbb{F}_q^k \rightarrow \mathbb{R}^+$, we have that $\mu \circ i = \tilde{\mu}$.
We replace
\[
P(e(C, a)) = \sum_{c \in C} \sum_{y \in F} n_{Y} \delta(a(y)) P(y | c) P(c)
\]
by
\[
E\mu(C, a, \mu, i) = \sum_{c \in C} \sum_{y \in F} n_{Y} \mu(a(y) - c) P(y | c) P(c).
\]
We replace

\[ P_e (C, a) = \sum_{c \in C} \sum_{y \in \mathbb{F}_q^n} \delta_{a(y),c} P (y|c) P (c) \]
Overall Expected Loss Function

We replace

$$P_e (C, a) = \sum_{c \in C} \sum_{y \in \mathbb{F}_q^n} \delta_{a(y), c} P(y|c) \, P(c)$$

by

$$E_\mu (C, a, \mu, i) = \sum_{c \in C} \sum_{y \in \mathbb{F}_q^n} \mu(a(y) - c) P(y|c) \, P(c).$$
Overall Expected Loss Function

Usual error problem
Usual error problem

Given $n$ and $k$, find a pair $(C, a)$ that minimizes the error probability function $P_e(C, a)$ where $C$ is a $[n, k]_q$ code and $a$ a decoder for $C$. 
Usual error problem

Given \( n \) and \( k \), find a pair \((C, a)\) that minimizes the error probability function \( P_e(C, a) \) where \( C \) is a \([n, k]_q\) code and \( a \) a decoder for \( C \).

The valued problem

Given \( n, k \) and a measure function \( \mu \) find \((C, a, i)\) that minimizes the overall expected loss function \( \mathbb{E}(C, a, \mu, i) \) where \( C \) is a \([n, k]_q\) code and \( a \) a decoder for \( C \).
"Size" of the error problem

\[
\left\{ [n, k]_q -\text{codes} \right\} \times \{ \text{ML decoders} \}
\]
Overall Expected Loss Function

"Size" of the error problem

\[ \left\{ [n, k]_q \text{-codes} \right\} \times \{ \text{ML decoders} \} \]

"Size" of the new problem

\[ \left\{ [n, k]_q \text{-codes} \right\} \times \{ \text{general decoders} \} \times \{ \text{embeddings of } \mathbb{F}_q^k \text{ onto } C \} \]
"Size" of the error problem

\[ \left\{ [n, k]_q \text{-codes}\right\} \times \{\text{ML decoders}\} \]

"Size" of the new problem

\[ \left\{ [n, k]_q \text{-codes}\right\} \times \{\text{general decoders}\} \times \left\{\text{embendings of } \mathbb{F}_q^k \text{ onto } C\right\} \]

\[ \prod_{k=1}^{n} \frac{q^n - q^{i-1}}{q^k - q^{i-1}} \times (q^n)^{q^k} \times \left(q^k\right)! \]
We consider a value function $\tilde{\mu} : \mathbb{F}_q^k \rightarrow \mathbb{R}^+$ as determined by the semantic value of the information.
We consider a value function $\tilde{\mu} : \mathbb{F}_q^k \rightarrow \mathbb{R}^+$ as determined by the semantic value of the information. Given a code $C$, we need to determine both an embending $i : \mathbb{F}_q^k \rightarrow C$ and a decoder $a : \mathbb{F}_q^n \rightarrow C$. 

Intuitively, how we shall look for a "good" $i : \mathbb{F}_q^k \rightarrow C$?
We consider a value function $\tilde{\mu} : \mathbb{F}_q^k \rightarrow \mathbb{R}^+$ as determined by the semantic value of the information. Given a code $\mathcal{C}$, we need to determine both an embedding $i : \mathbb{F}_q^k \rightarrow \mathcal{C}$ and a decoder $a : \mathbb{F}_q^n \rightarrow \mathcal{C}$.

Intuitively, how we shall look for a "good" $i : \mathbb{F}_q^k \rightarrow \mathcal{C}$?
Embendings of messages inside a code

We consider a value function $\tilde{\mu} : \mathbb{F}_q^k \rightarrow \mathbb{R}^+$ as determined by the semantic value of the information. Given a code $C$, we need to determine both an embedding $i : \mathbb{F}_q^k \rightarrow C$ and a decoder $a : \mathbb{F}_q^n \rightarrow C$.

Intuitively, how we shall look for a "good" $i : \mathbb{F}_q^k \rightarrow C$? We should place similar information close one to the other:
We consider a value function \( \tilde{\mu} : \mathbb{F}_q^k \to \mathbb{R}^+ \) as determined by the semantic value of the information. Given a code \( \mathcal{C} \), we need to determine both an embending \( i : \mathbb{F}_q^k \to \mathcal{C} \) and a decoder \( a : \mathbb{F}_q^n \to \mathcal{C} \).

Intuitively, how we shall look for a ”good” \( i : \mathbb{F}_q^k \to \mathcal{C} \)? We should place similar information close one to the other:

Figure: On the left, message-wise UEP as proposed by Borade, Nakiboglu and Zheng, 2009
Poset metrics

We consider a family of decoders that gives us a hope to have efficient decoding algorithms.
Poset metrics

We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

- \([n] = \{1, 2, \ldots, n\}\)
We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

- $[n] = \{1, 2, \ldots, n\}$
- $P = ([n], \preceq)$ a partial order (poset) on $[n]$

The $P$-weight $\omega_P(x) = |\langle \text{supp}(x) \rangle|$.

The $P$-distance $d_P(x, y) = \omega_P(x - y)$. 

Marcelo Firer (IMECC - UNICAMP)
September 2011
Poset metrics

We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

- \([n] = \{1, 2, ..., n\}\)
- \(P = ([n], \preceq)\) a partial order (poset) on \([n]\)
- \(I \subseteq [n]\) is an ideal if whenever \(i \in I\) and \(j \preceq i\) then \(j \in I\)
We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

- $[n] = \{1, 2, \ldots, n\}$
- $P = ([n], \preceq)$ a partial order (poset) on $[n]$
- $I \subset [n]$ is an **ideal** if whenever $i \in I$ and $j \preceq i$ then $j \in I$
- $\langle A \rangle$ is the ideal generated by $A$. 

\[ \text{supp}(x) = \{i | x_i \neq 0\} \]

\[ \omega_P(x) = |\langle \text{supp}(x) \rangle| \]

\[ d_P(x, y) = \omega_P(x - y) \]
Poset metrics

We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

- $[n] = \{1, 2, ..., n\}$
- $P = ([n], \preceq)$ a partial order (poset) on $[n]$
- $I \subset [n]$ is an ideal if whenever $i \in I$ and $j \preceq i$ then $j \in I$
- $\langle A \rangle$ is the ideal generated by $A$.
- Given $x = (x_1, ..., x_n) \in \mathbb{F}_q^n$ we define the support

$$\text{supp}(x) = \{i | x_i \neq 0\}$$
We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

\[ [n] = \{1, 2, \ldots, n\} \]

\[ P = ([n], \preceq) \] a partial order (poset) on \([n] \]

\( I \subset [n] \) is an \textbf{ideal} if whenever \( i \in I \) and \( j \preceq i \) then \( j \in I \)

\( \langle A \rangle \) is the ideal generated by \( A \).

Given \( x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n \) we define the \textbf{support}

\[ \text{supp} (x) = \{i | x_i \neq 0\} \]

\textbf{The \( P \)-weight}

\[ \omega_P (x) = |\langle \text{supp} (x) \rangle| \]
Poset metrics

We consider a family of decoders that gives us a hope to have efficient decoding algorithms.

- $[n] = \{1, 2, ..., n\}$
- $P = ([n], \preceq)$ a partial order (poset) on $[n]$
- $I \subset [n]$ is an ideal if whenever $i \in I$ and $j \preceq i$ then $j \in I$
- $\langle A \rangle$ is the ideal generated by $A$.
- Given $x = (x_1, ..., x_n) \in \mathbb{F}_q^n$ we define the support
  \[ \text{supp} (x) = \{ i | x_i \neq 0 \} \]
- The $P$-weight
  \[ \omega_P (x) = |\langle \text{supp} (x) \rangle| \]
- The $P$-distance
  \[ d_P (x, y) = \omega_P (x - y) \]
Figure: Hasse diagrams of posets
A nearest-neighbour $P$-decoder (NN-$P$) is a decoder $a$ such that
\[ d_P(y, a(y)) = \min \{ d_P(y, c) \mid c \in C \} \]
for every $y \in F_n^q$.

Remark: nearest-neighbour $P$-decoders may be somehow surprising:

If $d_P, 1(C)$ is the minimal distance of a code $C$ and $R_P(C)$ the packing radius, it is possible (depending on $P$) to have $R_P(C) = d_P, 1(C) - 1$. 

Marcelo Firer (IMECC - UNICAMP)
September 2011
A **nearest-neighbour $P$-decoder** (NN-$P$) is a decoder $a_P$ such that

$$d_P(y, a_P(y)) = \min \{d_P(y, c) | c \in C\}$$

for every $y \in \mathbb{F}_q^n$. 

**Remark**

Nearest-neighbour $P$-decoders may be somehow surprising: If $d_P(C)$ is the minimal distance of a code $C$ and $R_P(C)$ the packing radius, it is possible (depending on $P$) to have $R_P(C) = d_P(C) - 1$. 
Definition

A nearest-neighbour $P$-decoder (NN-$P$) is a decoder $a_P$ such that

$$d_P (y, a_P (y)) = \min \{d_P (y, c) | c \in C\}$$

for every $y \in \mathbb{F}_q^n$.

Remark

nearest-neighbour $P$-decoders may be somehow surprising:
**Definition**

A **nearest-neighbour P-decoder** (NN-P) is a decoder $a_P$ such that

$$d_P(y, a_P(y)) = \min \{d_P(y, c) | c \in C\}$$

for every $y \in \mathbb{F}_q^n$.

**Remark**

nearest-neighbour P-decoders may be somehow surprising: If $d_{P,1}(C)$ is the minimal distance of a code $C$ and $R_P(C)$ the packing radius, it is possible (depending on $P$) to have

$$R_P(C) = d_{P,1}(C) - 1$$
Hierarchical poset decoders

A poset \( P = (\mathbb{N}, \preceq) \) is called hierarchical if \( \mathbb{N} \) can be partitioned as \( \mathbb{N} = \bigcup_{l=1}^{h} H_l \) such that given \( i \in H_l \) and \( j \in H_{l'} \) (\( i \neq j \)) then \( i \preceq j \) iff \( l < l' \).

Figure: Hasse diagrams of hierarchical posets
A poset \( P = ([n], \preceq) \) is called **hierarchical** if \([n]\) can be partitioned as

\[
[n] = \bigcup_{l=1,\ldots,h} H_l
\]

such that given \( i \in H_{l_i} \) and \( j \in H_{l_j} \) (\( i \neq j \)) then \( i \preceq j \) iff \( l_i < l_j \).
A poset $P = ([n], \preceq)$ is called **hierarchical** if $[n]$ can be partitioned as

$$[n] = \bigcup_{l=1,\ldots,h} H_l$$

such that given $i \in H_{l_i}$ and $j \in H_{l_j}$ ($i \neq j$) then $i \preceq j$ iff $l_i < l_j$. 

![Hasse diagrams of hierarchical posets](image-url)
Hierarchical poset decoders

Hierarchical posets have the following interesting features:

Hierarchical posets have the following interesting features:

To determine important invariants of a code (such as weight hierarchy, packing and covering radius) is simpler than in the usual Hamming setting.

If we write

\begin{align*}
\mathbf{n}^i &= |H^i| = \\
F_{n^q} &= F_{n^1} \oplus F_{n^2} \oplus \cdots \oplus F_{n^h}
\end{align*}

then syndrome decoding of a \([n^q, k^q]\)-code relatively to a \(P\) is “essentially” equivalent to decoding a code \(C = C_1 \oplus C_2 \oplus \cdots \oplus C_h\) with \(C_i \subseteq F_{n^i}^q\).

If \(k^i = \dim(C_i)\) then the complexity of syndrome decoding is about

\begin{align*}
n - k \leq h \sum_{i=1}^{i=h} q^{n^i - k^i} \leq q^{n - k} = q^{\sum (n^i - k^i)}
\end{align*}
Hierarchical poset decoders

Hierarchical posets have the following interesting features:

- To determine important invariants of a code (such as weight hierarchy, packing and covering radius) is simpler than in the usual Hamming setting.
Hierarchical poset decoders

Hierarchical posets have the following interesting features:

- To determine important invariants of a code (such as weight hierarchy, packing and covering radius) is simpler than in the usual Hamming setting.
- If we write \( n_i = |H_i| \) and \( \mathbb{F}_q^n = \mathbb{F}_q^{n_1} \oplus \mathbb{F}_q^{n_2} \oplus \cdots \oplus \mathbb{F}_q^{n_h} \) then syndrome decoding of a \([n, k]_q\)-code relatively to \( a_P \) is “essentially” equivalent to decoding a code

\[
C = C_1 \oplus C_2 \oplus \cdots \oplus C_h
\]

with \( C_i \subseteq \mathbb{F}_q^{n_i} \).
Hierarchical poset decoders

Hierarchical posets have the following interesting features:

- To determine important invariants of a code (such as weight hierarchy, packing and covering radius) is simpler than in the usual Hamming setting.
- If we write $n_i = |H_i| = |H|$ and $F_q^n = F_q^{n_1} \oplus F_q^{n_2} \oplus \cdots \oplus F_q^{n_h}$ then syndrome decoding of a $[n, k]_q$-code relatively to $a_P$ is "essentially" equivalent to decoding a code

$$C = C_1 \oplus C_2 \oplus \cdots \oplus C_h$$

with $C_i \subseteq F_q^{n_i}$. If $k_i = \dim (C_i)$ then the complexity of syndrome decoding is about

$$n - k \leq \sum_{i=1}^{h} q^{n_i-k_i} \leq q^{n-k} = q \sum (n_i-k_i)$$
Comparing decoders

Consider the difference between overall expected loss functions:

\[ T(C, i, a_P, a_Q)(\mu) = \mathbb{E}_\mu(C, a_P, \mu, i) - \mathbb{E}_\mu(C, a_Q, \mu, i) \]

We say that, for the measure function \( \mu \), it is better to decode with \( a_P \) if

\[ T(C, i, a_P, a_Q)(\mu) < 0 \]

it is better to decode with \( a_Q \) if

\[ T(C, i, a_P, a_Q)(\mu) > 0 \]
Comparing decoders

Consider the difference between overall expected loss functions $T(C, i, a_P, a_Q)(\mu) = E_{\mu}(C, a_P, \mu, i) - E_{\mu}(C, a_Q, \mu, i)$.

We say that, for the measure function $\mu$, it is better to decode with $a_P$ if $T(C, i, a_P, a_Q)(\mu) < 0$ and it is better to decode with $a_Q$ if $T(C, i, a_P, a_Q)(\mu) > 0$. 

---

Marcelo Firer (IMECC - UNICAMP)

September 2011
Comparing decoders

Consider the difference between overall expected loss functions

$$\mathcal{T}(C, i, a_p, a_Q)(\mu) = \mathbb{E}_\mu(C, a_P, \mu, i) - \mathbb{E}_\mu(C, a_Q, \mu, i)$$
Comparing decoders

Consider the difference between overall expected loss functions

\[ T_{(C,i,a_P,a_Q)}(\mu) = \mathbb{E}_\mu(C, a_P, \mu, i) - \mathbb{E}_\mu(C, a_Q, \mu, i) \]

We say that, for the measure function \( \mu \),
Comparing decoders

Consider the difference between overall expected loss functions

\[ T_{(C,i,a_P,a_Q)}(\mu) = \mathbb{E}_\mu (C, a_P, \mu, i) - \mathbb{E}_\mu (C, a_Q, \mu, i) \]

We say that, for the measure function \( \mu \),

it is better to decode with \( a_P \) if \( T_{(C,i,a_P,a_Q)}(\mu) < 0 \)
Comparing decoders

Consider the difference between overall expected loss functions

\[ T_{(C,i,a_p,a_Q)} (\mu) = \mathbb{E}_\mu (C, a_P, \mu, i) - \mathbb{E}_\mu (C, a_Q, \mu, i) \]

We say that, for the measure function \( \mu \),

it is better to decode with \( a_P \) if \( T_{(C,i,a_p,a_Q)} (\mu) < 0 \)

it is better to decode with \( a_Q \) if \( T_{(C,i,a_p,a_Q)} (\mu) > 0 \)
No poset decoder is dispensable

Theorem

Under quite “suitable conditions”, given a code $C$, there are posets $P$ and $Q$ such that we can split \{value functions\} $\equiv (R + |C|)$ as a disjoint union $A \cup B$ of non-empty open-sets such that it is better to decode with a $P$ if $\mu \in A$ it is better to decode with a $Q$ if $\mu \in B$ in other words, given $C$, there are $P$ and $Q$ such that for some value functions $a_P$ is better than $a_Q$ and vice-versa.
Theorem

Under quite "suitable conditions", given a code $C$, there are posets $P$ and $Q$ such that we can split \{value functions\} $\equiv (\mathbb{R}^+)^{|C|}$ as a disjoint union $A \cup B$ of non-empty open-sets such that

- it is better to decode with $a_P$ if $\mu \in A$
- it is better to decode with $a_Q$ if $\mu \in B$

in other words, given $C$, there are $P$ and $Q$ such that for some value functions $a_P$ is better than $a_Q$ and vice-versa.
No code is dispensable
Theorem

Under quite ”suitable conditions”, given hierarchical posets $P$ and $Q$ there is a code $C$ such that we can split $\{\text{value functions}\} \equiv (\mathbb{R}^+)^{|C|}$ as a disjoint union $A \cup B$ of non-empty open-sets such that

- it is better to decode with $a_P$ if $\mu \in A$
- it is better to decode with $a_Q$ if $\mu \in B$

in other words, given $C$, there are $P$ and $Q$ such that for some value functions $a_P$ is better than $a_Q$ and vice-versa.
For the BSMC, equation
\[ T(C,i,a_p,a_Q)(\mu) = E_{\mu}(C,a_P,\mu,i) - E_{\mu}(C,a_Q,\mu,i) = 0 \]
is linear on the variables \( \{\mu(c) | c \in C\} \).
In this case there is very simple condition to ensure that
\[ T(C,i,a_p,a_Q)(\mu) \] is not identicaly 0.
For the BSMC, equation

\[ T(c,i,a_p,a_Q)(\mu) = \mathbb{E}_\mu(C,a_P,\mu,i) - \mathbb{E}_\mu(C,a_Q,\mu,i) = 0 \]

is linear on the variables \( \{\mu(c) | c \in C\} \).
For the BSMC, equation

\[ T_{(\mathcal{C}, i, a_p, a_Q)} (\mu) = \mathbb{E}_\mu (\mathcal{C}, a_p, \mu, i) - \mathbb{E}_\mu (\mathcal{C}, a_Q, \mu, i) = 0 \]

is linear on the variables \( \{\mu (c) | c \in \mathcal{C}\} \).

In this case there is very simple condition to ensure that \( T_{(\mathcal{C}, i, a_p, a_Q)} (\mu) \) is not identically 0.
The space \((\mathbb{R}^+)^{|\mathcal{C}|}\) of value functions can be decomposed as a union \((\mathbb{R}^+)^{|\mathcal{C}|} = \bigcup_{i=1}^{\text{r}} A_i\) where each \(A_i\) has the following properties:

1. To each \(A_i\) there is a decoder \(a\) such that for every \(\mu \in A_i\) decoder \(a\) is the optimal decoder of the given code.
2. Each \(A_i\) is a cone over a polyhedron with non-empty interior.
3. If \(i \neq j\) then \(\text{int}(A_i) \cap \text{int}(A_j) = \emptyset\).
The space $(\mathbb{R}^+)^{|C|}$ of value functions can be decomposed as a union

$$(\mathbb{R}^+)^{|C|} = \bigcup_{i=1}^r A_i$$

where each $A_i$ has the following properties:

1. To each $A_i$ there is a decoder $a$ such that for every $\mu \in A_i$ decoder $a$ is the optimal decoder of the given code.
2. Each $A_i$ is a cone over a polyhedron with non-empty interior.
3. If $i \neq j$ then $\text{int}(A_i) \cap \text{int}(A_j) = \emptyset$. 
The space $(\mathbb{R}^+)^{|C|}$ of value functions can be decomposed as a union

$$(\mathbb{R}^+)^{|C|} = \bigcup_{i=1}^{r} A_i$$

where each $A_i$ has the following properties:

1. To each $A_i$ there is a decoder $a$ such that for every $\mu \in A_i$ decoder $a$ is the optimal decoder of the given code.
The space \((\mathbb{R}^+)^{|C|}\) of value functions can be decomposed as a union
\[
(\mathbb{R}^+)^{|C|} = \bigcup_{i=1}^{r} A_i
\]
where each \(A_i\) has the following properties:

1. To each \(A_i\) there is a decoder \(a\) such that for every \(\mu \in A_i\) decoder \(a\) is the optimal decoder of the given code.
2. Each \(A_i\) is a cone over a polyedron with non-empty interior.
The space \((\mathbb{R}^+)^{|C|}\) of value functions can be decomposed as a union

\[
(\mathbb{R}^+)^{|C|} = \bigcup_{i=1}^{r} A_i
\]

where each \(A_i\) has the following properties:

1. To each \(A_i\) there is a decoder \(a\) such that for every \(\mu \in A_i\) decoder \(a\) is the optimal decoder of the given code.
2. Each \(A_i\) is a cone over a polyedron with non-empty interior.
3. If \(i \neq j\) then \(\text{int} \ (A_i) \cap \text{int} \ (A_j) = \emptyset\)
Toy example: What we can actually achieve

We consider the information set to contain $16 = 2^4$ different tones in gray-scale palette. We encode this information set as a $[7, 4]$ Hamming code and identify each pixel with a message to produce the picture.
We consider as our information set to contain $16 = 2^4$ different tones in gray-scale pallette. We encode this information set as a $[7, 4]_2$ Hamming code and identify each pixel with a message to produce the picture.
We consider as our information set to contain $16 = 2^4$ different tones in gray-scale palette. We encode this information set as a $[7, 4]_2$ Hamming code and identify each pixel with a message to produce the picture.

HELLO

WORLD
Toy example: What we can actually achieve

We use a pseudo-random number generator to produce noise over our channel and then decode the "received image" using two different decoders: ML decoder and a poset decoder determined by the poset $P$ such that $1 \preceq 2 \preceq 3 \preceq 4 \preceq 5 \preceq 6 \preceq 7$.

For comparison, pixels that are correctly decoded are painted in purple.

Figure: Pixels in purple means correct decoding. Decoder $a_H$ on the left and $a_P$ on the right. Error probability = 0.
We use a pseudo-random number generator to produce noise over our channel and then decode the ”received image” using two different decoders: ML decoder and a poset decoder determined by the poset $P$ such that $1 \preceq 2 \preceq 3 \preceq 4 \preceq 5 \preceq 6 \preceq 7$. For comparison, pixels that are correctly decoded are painted in purple.
We use a pseudo-random number generator to produce noise over our channel and then decode the ”received image” using two different decoders: ML decoder and a poset decoder determined by the poset $P$ such that $1 \preceq 2 \preceq 3 \preceq 4 \preceq 5 \preceq 6 \preceq 7$. For comparison, pixels that are correctly decoded are painted in purple.
We use a pseudo-random number generator to produce noise over our channel and then decode the "received image" using two different decoders: ML decoder and a poset decoder determined by the poset $P$ such that $1 \preceq 2 \preceq 3 \preceq 4 \preceq 5 \preceq 6 \preceq 7$.

For comparison, pixels that are correctly decoded are painted in purple.

**Figure:** Pixels in purple means correct decoding. Decoder $a_H$ on the left and $a_P$ on the right. Error probability $= 0.3$
Toy example: What we can actually achieve
Toy example: What we can actually achieve

Figure: A close zoom at the picture
Toy example: What we can actually achieve

Looking now at the pictures how they were actually decoded we have: Figure: Decoder a H on the left and a P on the right. Error probability of the channel = 0.
Looking now at the pictures how the were actually decoded we have:
Toy example: What we can actually achieve

Looking now at the pictures how the were actually decoded we have:

Figure: Decoder $a_H$ on the left and $a_P$ on the right. Error probability of the channel $= 0.3$
Open problems

1. To get rid of the "suitable conditions" in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".
3. To estimate how much we lose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.
4. Consider a fixed family of value functions and search for upper bounds for $E(a, \mu)$.
5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.
6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is "easier" to manage).
7. Find encoding-decoding schemes for actual problems.
8. And more...
1. To get rid of the "‘suitable conditions’’ in the stated theorems.
Open problems

1. To get rid of the "suitable conditions" in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".

Find encoding-decoding schemes for actual problems. And more...
Open problems

1. To get rid of the "‘suitable conditions’’ in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".
3. To estimate how much we lose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.

4. Consider a fixed family of value functions and search for upper bounds for $E(a, \mu)$.
5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.
6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is "easier" to manage).
7. Find encoding-decoding schemes for actual problems.
8. And more...
Open problems

1. To get rid of the "‘suitable conditions’’ in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".
3. To estimate how much we lose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.
4. Consider a fixed family of value functions and search for upper bounds for $\mathbb{E}(a, \mu)$.
5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.
6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is "easier" to manage).
7. Find encoding-decoding schemes for actual problems.
8. And more...
Open problems

1. To get rid of the "'suitable conditions'" in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".
3. To estimate how much we loose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.
4. Consider a fixed family of value functions and search for upper bounds for $E(a, \mu)$.
5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.

And more...
Open problems

1. To get rid of the "‘suitable conditions’” in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the "toy example”.
3. To estimate how much we loose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.
4. Consider a fixed family of value functions and search for upper bounds for $\mathbb{E}(a, \mu)$.
5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.
6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is ”easier” to manage).
1. To get rid of the "‘suitable conditions’" in the stated theorems.

2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".

3. To estimate how much we loose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.

4. Consider a fixed family of value functions and search for upper bounds for $E(a, \mu)$.

5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.

6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is "easier" to manage).
Open problems

1. To get rid of the "'suitable conditions'" in the stated theorems.

2. Understand (and prove) the heuristic used to produce the pictures in the "toy example".

3. To estimate how much we lose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.

4. Consider a fixed family of value functions and search for upper bounds for $E(a, \mu)$.

5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.

6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is "easier" to manage).

7. Find encoding-decoding schemes for actual problems.
Open problems

1. To get rid of the ”‘suitable conditions’” in the stated theorems.
2. Understand (and prove) the heuristic used to produce the pictures in the ”toy example”.
3. To estimate how much we loose (asymptotically) restricting ourselves to poset decoders and hierarchical-poset decoders.
4. Consider a fixed family of value functions and search for upper bounds for $E(a, \mu)$.
5. Find bounds for the expected loss when considering a particular family of NN poset decoders, specially those determined by hierarchical posets.
6. Find bounds for the expected loss function fixing a family of value functions and a type of NN decoder (combines both the previous one into a problem that is ”easier” to manage).
7. Find encoding-decoding schemes for actual problems.
8. And more...