Iterative Processing of Information

Joseph J. Boutros

Ecole Nationale Supérieure des Télécommunications, Paris

AUST, Lebanon

April 19, 2006
Presentation outline

1. Historical notes and motivation
2. Codes on graphs for the erasure channel
3. Asymptotic analysis of iterative decoding
4. Probabilistic decoding on soft channels (e.g. the gaussian channel)
5. Iterative receivers for multiple access (CDMA) and multiple antennas (MIMO)
Introduction

Motivation:

1. Split a complex problem into two or more simpler sub-problems.

2. Under a given optimality criterion, aim at finding the optimal solution with a reasonable complexity.

3. Iterative methods are useful for both design (e.g. compound codes) and optimization (e.g. iterative decoding).

Where?

1. Iterative methods exist in numerical analysis and other fields in mathematics.

2. They appeared in the field of digital transmission at least since the mid 50's.
   - In coding and information theory
   - In communications theory
   - In signal processing
**Introduction**

**Motivation:**

1. Split a complex problem into two or more simpler sub-problems.
2. Under a given optimality criterion, aim at finding the optimal solution with a reasonable complexity.
3. Iterative methods are useful for both design (e.g. compound codes) and optimization (e.g. iterative decoding).

**Where?**

1. Iterative methods exist in numerical analysis and other fields in mathematics.
2. They appeared in the field of digital transmission at least since the mid 50's.
   - In coding and information theory
   - In communications theory
   - In signal processing
All ingredients are here for the birth of modern coding/communication theory.

- Blahut & Arimoto, Computation of channel capacity, 1972.
- Hartmann & Rudolph, APP decoding based on the dual code, 1976.
- M. Tanner, Graph codes and iterative algorithms, 1981.
The era of capacity achieving codes.

- Berrou & Glavieux, Parallel Turbo Codes, 1993.
- More recent papers (1998-2005) can be found in IEEE journals and other scientific publications.
Toy Example


Toy example: Find the minimum distance between two sets A and B.

Exhaustive search: $|A| \times |B| = O(n^2)$ metric computations. It is possible to find the minimum distance after $O(n)$ metric computations.
Toy Example

- Toy example: Find the minimum distance between two sets A and B.

![Toy Example Diagram](image)

- Exhaustive search: \(|A| \times |B| = O(n^2)\) metric computations. It is possible to find the minimum distance after \(O(n)\) metric computations.
Iteration 1 - Initialize

Set A

Set B
Iteration 1 - Minimize distance

Set A

Set B
Iteration 2 - Initialize

Set A

Set B
Iteration 2 - Minimize distance
Iteration 3 - Initialize

Set A

Set B
**Iteration 3 - Minimize distance**
We consider the ergodic binary erasure channel (BEC). The channel input is binary and its output is ternary. Only erasures occur on this channel, no errors.

- Shannon capacity of the BEC is $C = 1 - \epsilon$ bits per channel use.
- Rate 1/2 code $\rightarrow \epsilon_{max} = 1/2$. Rate 1/4 code $\rightarrow \epsilon_{max} = 3/4$
We consider the ergodic binary erasure channel (BEC). The channel input is binary and its output is ternary. Only erasures occur on this channel, no errors.

- Shannon capacity of the BEC is $C = 1 - \epsilon$ bits per channel use.
- Rate 1/2 code $\rightarrow \quad \epsilon_{max} = 1/2$. Rate 1/4 code $\rightarrow \quad \epsilon_{max} = 3/4$
Bipartite graph representation, the Tanner Graph (1/2)

- We build a code defined by a graph in order to fill erasures on the BEC.
- A low-density parity-check (LDPC) code of length $N$ bits and dimension $K$ is defined by a bipartite graph.
- Bitnodes are drawn as circles, checknodes are drawn as squares.

For a regular $(d_b, d_c)$ LDPC, bitnodes have degree $d_b$ and checknodes have degree $d_c$. The coding rate is

$$R_c = \frac{K}{N} \geq 1 - \frac{d_b}{d_c}$$

**SPC** \((3, 2)_2\)

PCE: $c_1 + c_2 + c_3 = 0$
Introduction  Erasure Channel  Asymptotic Analysis  Soft Channels  Iterative Receivers  Conclusions

Bipartite graph representation, the Tanner Graph (2/2)

Tanner graph of an LDPC code, length $N = 1000$, dimension $K = 500$, coding rate $R_c = 1/2$. The 3000 edges are chosen at random.
Decoding algorithm

The number of checknodes is denoted by $L = N - K$.

In the presence of a uniform source encoded by an LDPC(N,K) code whose codewords are transmitted on an iid BEC channel, the iterative non probabilistic decoding algorithm is given by the following steps:

- **Step 0**: Initialize $Iter = 0$ and $j = 1$.

- **Step 1**: Count the number $\mu$ of erased bits connected to checknode $j$.

- **Step 2**: If $\mu = 1$ then fill the erased bit by summing other bits modulo 2.

- **Step 3**: Increment $j$. If $j > L$ then increment $Iter$ and set $j = 1$.

- **Step 4**: If $Iter > MaxIter$ then Stop else Goto Step 1.
Decoding algorithm

The number of checknodes is denoted by $L = N - K$.

In the presence of a uniform source encoded by an LDPC(N,K) code whose codewords are transmitted on an iid BEC channel, the iterative non probabilistic decoding algorithm is given by the following steps:

- **Step 0:** Initialize $Iter = 0$ and $j = 1$.

- **Step 1:** Count the number $\mu$ of erased bits connected to checknode $j$.

- **Step 2:** If $\mu = 1$ then fill the erased bit by summing other bits modulo 2.

- **Step 3:** Increment $j$. If $j > L$ then increment $Iter$ and set $j = 1$.

- **Step 4:** If $Iter > MaxIter$ then Stop else Goto Step 1.
Decoding algorithm

The number of checknodes is denoted by $L = N - K$.

In the presence of a uniform source encoded by an LDPC(N,K) code whose codewords are transmitted on an iid BEC channel, the iterative non probabilistic decoding algorithm is given by the following steps:

- **Step 0**: Initialize $Iter = 0$ and $j = 1$.

- **Step 1**: Count the number $\mu$ of erased bits connected to checknode $j$.

- **Step 2**: If $\mu = 1$ then fill the erased bit by summing other bits modulo 2.

- **Step 3**: Increment $j$. If $j > L$ then increment $Iter$ and set $j = 1$.

- **Step 4**: If $Iter > MaxIter$ then Stop else Goto Step 1.
Decoding algorithm

The number of checknodes is denoted by $L = N - K$.
In the presence of a uniform source encoded by an LDPC(N,K) code whose codewords are transmitted on an iid BEC channel, the iterative non probabilistic decoding algorithm is given by the following steps:

- **Step 0:** Initialize $Iter = 0$ and $j = 1$.

- **Step 1:** Count the number $\mu$ of erased bits connected to checknode $j$.

- **Step 2:** If $\mu = 1$ then fill the erased bit by summing other bits modulo 2.

- **Step 3:** Increment $j$. If $j > L$ then increment $Iter$ and set $j = 1$.

- **Step 4:** If $Iter > MaxIter$ then Stop else Goto Step 1.
Regular (3,6) binary LDPC, \( N = 1000, K = 500 \), iid BEC
Assume an infinite length code with a graph representation without cycles (tree representation). The local neighborhood of a bitnode \( v \) in a regular (3,6) LDPC code is shown below. Let \( p_i \) denote the erasure probability at iteration \( i \).

It is trivial to prove that

\[
p_{i+1} = p_0 \times (1 - (1 - p_i)^5)^2
\]
Analysis of iterative decoding (1/3)

Assume an infinite length code with a graph representation without cycles (tree representation). The local neighborhood of a bitnode $v$ in a regular (3,6) LDPC code is shown below. Let $p_i$ denote the erasure probability at iteration $i$.

It is trivial to prove that

$$p_{i+1} = p_0 \times (1 - (1 - p_i)^5)^2$$
Analysis of iterative decoding (2/3)

Plot of the transfer function $f(x) = p_0(1 - (1 - x)^5)^2$ (in blue) versus $y = x$ (in green).
Analysis of iterative decoding (3/3)

Plot of the transfer function $f(x) = p_0(1 - (1 - x)^5)^2$ versus $y = x$. 

![Plot of the transfer function](image-url)
An LDPC code is said to be 'irregular' if graph nodes of same kind do not have equal degree.

**Left irregularity:** Bitnodes have different degrees. The fraction of edges connected to bitnodes of degree $i$ is $\lambda_i$. The degree distribution is defined by

$$\lambda(x) = \sum_{i=1}^{d_b} \lambda_i x^{i-1}$$

where $\lambda(1) = 1$ and $0 \leq \lambda_i \leq 1 \ \forall i$.

**Right irregularity:** Checknodes have different degrees. The fraction of edges connected to checknodes of degree $j$ is $\rho_j$. The degree distribution is defined by

$$\rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1}$$

where $\rho(1) = 1$ and $0 \leq \rho_j \leq 1 \ \forall j$. 
Evolution for irregular codes (1/2)

An LDPC code is said to be 'irregular' if graph nodes of same kind do not have equal degree.

Left irregularity: Bitnodes have different degrees. The fraction of edges connected to bitnodes of degree $i$ is $\lambda_i$. The degree distribution is defined by

$$\lambda(x) = \sum_{i=1}^{d_b} \lambda_i x^{i-1}$$

where $\lambda(1) = 1$ and $0 \leq \lambda_i \leq 1 \ \forall i$.

Right irregularity: Checknodes have different degrees. The fraction of edges connected to checknodes of degree $j$ is $\rho_j$. The degree distribution is defined by

$$\rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1}$$

where $\rho(1) = 1$ and $0 \leq \rho_j \leq 1 \ \forall j$. 
Evolution for irregular codes (1/2)

- An LDPC code is said to be 'irregular' if graph nodes of same kind do not have equal degree.

- Left irregularity: Bitnodes have different degrees. The fraction of edges connected to bitnodes of degree $i$ is $\lambda_i$. The degree distribution is defined by

$$\lambda(x) = \sum_{i=1}^{d_b} \lambda_i x^{i-1}$$

where $\lambda(1) = 1$ and $0 \leq \lambda_i \leq 1 \ \forall i$.

- Right irregularity: Checknodes have different degrees. The fraction of edges connected to checknodes of degree $j$ is $\rho_j$. The degree distribution is defined by

$$\rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1}$$

where $\rho(1) = 1$ and $0 \leq \rho_j \leq 1 \ \forall j$. 
Evolution for irregular codes (2/2)

- By looking to the tree neighborhood of a graph node, it is easy to show that erasure probability $p_i$ after LDPC decoding at iteration $i$ on the binary erasure channel satisfies

$$p_{i+1} = p_0 \times \lambda(1 - \rho(1 - p_i))$$

where $p_0 = \epsilon$ is the channel erasure probability.

- The stability condition of the fixed point at the origin is obtained by writing that $f'(0) < 1$. Thus, the necessary and sufficient condition for stability of 0 is

$$\lambda'(0) \times \rho'(1) \times p_0 < 1$$

- For a general type of channels, assume that all-zero codeword is transmitted (geometrical uniformity is satisfied). If log-ratios are used, $LR = \log\left(\frac{P(0)}{P(1)}\right)$, then the fixed point is at $+\infty$. The stability condition becomes

$$\lambda'(0)\rho'(1) \int_{-\infty}^{+\infty} p_0(x)e^{-x/2}dx < 1$$
Evolution for irregular codes (2/2)

- By looking to the tree neighborhood of a graph node, it is easy to show that erasure probability $p_i$ after LDPC decoding at iteration $i$ on the binary erasure channel satisfies

$$p_{i+1} = p_0 \times \lambda(1 - \rho(1 - p_i))$$

where $p_0 = \epsilon$ is the channel erasure probability.

- The stability condition of the fixed point at the origin is obtained by writing that $f'(0) < 1$. Thus, the necessary and sufficient condition for stability of 0 is

$$\lambda'(0) \times \rho'(1) \times p_0 < 1$$

- For a general type of channels, assume that all-zero codeword is transmitted (geometrical uniformity is satisfied). If log-ratios are used, $LR = \log \left( \frac{P(0)}{P(1)} \right)$, then the fixed point is at $+\infty$. The stability condition becomes

$$\lambda'(0)\rho'(1) \int_{-\infty}^{+\infty} p_0(x) e^{-x/2} dx < 1$$
Evolution for irregular codes (2/2)

- By looking to the tree neighborhood of a graph node, it is easy to show that erasure probability $p_i$ after LDPC decoding at iteration $i$ on the binary erasure channel satisfies

$$p_{i+1} = p_0 \times \lambda (1 - \rho (1 - p_i))$$

where $p_0 = \epsilon$ is the channel erasure probability.

- The stability condition of the fixed point at the origin is obtained by writing that $f'(0) < 1$. Thus, the necessary and sufficient condition for stability of 0 is

$$\lambda'(0) \times \rho'(1) \times p_0 < 1$$

- For a general type of channels, assume that all-zero codeword is transmitted (geometrical uniformity is satisfied). If log-ratios are used, $LR = \log \left( \frac{P(0)}{P(1)} \right)$, then the fixed point is at $+\infty$. The stability condition becomes

$$\lambda'(0) \rho'(1) \int_{-\infty}^{+\infty} p_0(x) e^{-x/2} dx < 1$$
APP Decoding of Binary SPC Codes (1/2)

Let us describe the soft-input soft-output (SISO) decoder which is capable of determining the a posteriori probability $APP(c_i) = P(c_i|r, PCE)$ for each bit $c_i$.

- **The observation** of $c_i$ is a probability proportional to the channel likelihood

  $$obs(c_i) \propto p(r_i|c_i) \propto \exp(-\frac{(r_i - I(c_i))^2}{2\sigma_n^2})$$

- The **a priori information** $\pi(c_i)$ is produced by a genie. Any other code connected to our code is a potential genie. Write $\pi(c_i) = 1/2$ when no a priori information is available.

- The **extrinsic information** is generated by the SISO decoder based on the PCE constraint. It can be considered as a new a priori information produced by our decoder. For example, since $c_1 = c_2 + c_3 + \ldots + c_n$, the SISO-APP decoder computes $Extr(c_1)$ as follows:

  $$Extr(c_1 = 1) = \frac{1}{2} \times \left(1 - \prod_{i=2}^{n}(1 - 2p_i)\right)$$

  where $p_i \propto \pi_i(c_i = 1) \times obs(c_i = 1)$. 
Let us describe the soft-input soft-output (SISO) decoder which is capable of determining the a posteriori probability $APP(c_i) = P(c_i|r, PCE)$ for each bit $c_i$.

- **The observation** of $c_i$ is a probability proportional to the channel likelihood

$$\text{obs}(c_i) \propto p(r_i|c_i) \propto \exp\left(-\frac{(r_i - I(c_i))^2}{2\sigma_n^2}\right)$$

- **The a priori information** $\pi(c_i)$ is produced by a genie. Any other code connected to our code is a potential genie. Write $\pi(c_i) = 1/2$ when no a priori information is available.

- **The extrinsic information** is generated by the SISO decoder based on the PCE constraint. It can be considered as a new a priori information produced by our decoder. For example, since $c_1 = c_2 + c_3 + \ldots + c_n$, the SISO-APP decoder computes $\text{Extr}(c_1)$ as follows:

$$\text{Extr}(c_1 = 1) = \frac{1}{2} \times \left(1 - \prod_{i=2}^{n}(1 - 2p_i)\right)$$

where $p_i \propto \pi_i(c_i = 1) \times \text{obs}(c_i = 1)$. 
APP Decoding of Binary SPC Codes (1/2)

Let us describe the soft-input soft-output (SISO) decoder which is capable of determining the a posteriori probability $APP(c_i) = P(c_i|r, PCE)$ for each bit $c_i$.

- The observation of $c_i$ is a probability proportional to the channel likelihood

$$obs(c_i) \propto p(r_i|c_i) \propto \exp\left(-\frac{(r_i - I(c_i))^2}{2\sigma_n^2}\right)$$

- The a priori information $\pi(c_i)$ is produced by a genie. Any other code connected to our code is a potential genie. Write $\pi(c_i) = 1/2$ when no a priori information is available.

- The extrinsic information is generated by the SISO decoder based on the PCE constraint. It can be considered as a new a priori information produced by our decoder. For example, since $c_1 = c_2 + c_3 + \ldots + c_n$, the SISO-APP decoder computes $Extr(c_1)$ as follows:

$$Extr(c_1 = 1) = \frac{1}{2} \times \left(1 - \prod_{i=2}^{n}(1 - 2p_i)\right)$$

where $p_i \propto \pi_i(c_i = 1) \times obs(c_i = 1)$. 
APP Decoding of Binary SPC Codes (2/2)

- The a posteriori probability, given the total observation $r$, some apriori information $\pi_i$ and the SPC code constraint, is proportional to the product of the two opposite streams on a graph edge,

$$APP(c_i) \propto \pi_i \times obs(c_i) \times Extr(c_i)$$

- The proportionality factors are determined by forcing the following sum:

$$APP(c_i = 0) + APP(c_i = 1) = 1.$$
(3,6) LDPC, \( N = 10000, \ K = 5000, \) BPSK + AWGN

Regular(6,3) LDPC code, length=10000 bits, dimension=5000, Gaussian channel, BPSK

**Bit Error Probability versus Eb/N0**

- Shannon Limit
- BPSK Input 0.5 bits/dimension
- Random Regular Graph
- Probabilistic Decoding (exact sum-product)

**Results:**
- 1 iteration
- 10 iterations
- 32 iterations
- 64 iterations
- 128 iterations
Consider a frequency non-selective multiple-input multiple-output (MIMO) channel with $n_t$ transmit antennas and $n_r$ receive antennas.

- Channel coefficients are given by the entries of a $n_t \times n_r$ matrix $H = [h_{ij}]$, where $h_{ij}$ is the complex fading of the channel path linking transmit antenna $i$ to receive antenna $j$.

- The channel output is
  \[ y = xH + \eta \]
  where $x \in (M - QAM)^{n_t} \subseteq \mathbb{C}^{n_t}$, $y \in \mathbb{C}^{n_r}$, and $\eta$ is an additive white complex gaussian noise vector.

- A multidimensional alphabet $\Omega = (M - QAM)^{n_t}$ of size $M^{n_t} = 2^{mn_t}$. 
Iterative APP detection for MIMO channels (1/2)

Consider a frequency non-selective multiple-input multiple-output (MIMO) channel with $n_t$ transmit antennas and $n_r$ receive antennas.

Channel coefficients are given by the entries of a $n_t \times n_r$ matrix $H = [h_{ij}]$, where $h_{ij}$ is the complex fading of the channel path linking transmit antenna $i$ to receive antenna $j$.

The channel output is

$$y = xH + \eta$$

where $x \in (M - QAM)^{n_t} \subset \mathbb{C}^{n_t}$, $y \in \mathbb{C}^{n_r}$, and $\eta$ is an additive white complex gaussian noise vector.

A multidimensional alphabet $\Omega = (M - QAM)^{n_t}$ of size $M^{n_t} = 2^{mn_t}$.
Consider a frequency non-selective multiple-input multiple-output (MIMO) channel with \( n_t \) transmit antennas and \( n_r \) receive antennas.

Channel coefficients are given by the entries of a \( n_t \times n_r \) matrix \( H = [h_{ij}] \), where \( h_{ij} \) is the complex fading of the channel path linking transmit antenna \( i \) to receive antenna \( j \).

The channel output is

\[
y = xH + \eta
\]

where \( x \in (M - QAM)^{n_t} \subset \mathbb{C}^{n_t} \), \( y \in \mathbb{C}^{n_r} \), and \( \eta \) is an additive white complex gaussian noise vector.

A multidimensional alphabet \( \Omega = (M - QAM)^{n_t} \) of size \( M^{n_t} = 2^{mn_t} \).
The observation of a multidimensional symbol is

\[
\text{obs}(x) \propto p(y|x) \propto \exp \left( -\frac{\|y - xH\|^2}{2\sigma^2} \right)
\]

Independent a priori information \( \pi(x) = \prod_{j=1}^{mn} \pi(b_j) \).

The a posteriori probability for a complex QAM symbol is

\[
APP(x_i) = \sum_{x \in \Omega|x_i} APP(x) \propto \sum_{x \in \Omega|x_i} \text{obs}(x) \prod_{\ell=1}^{nt} \pi(x_\ell) \propto \pi(x_i) \text{Extr}(x_i)
\]

The a posteriori information for binary elements is

\[
APP(b_j) \propto \pi(b_j) \sum_{x \in \Omega|b_j} \text{obs}(x) \prod_{\ell \neq j} \pi(b_\ell) \text{Extr}(b_j)
\]
Iterative APP detection for MIMO channels (2/2)

The observation of a multidimensional symbol is

\[ \text{obs}(x) \propto p(y|x) \propto \exp \left( -\frac{||y - xH||^2}{2\sigma^2} \right) \]

- Independent a priori information \( \pi(x) = \prod_{j=1}^{mn} \pi(b_j) \).

- The a posteriori probability for a complex QAM symbol is

\[
\text{APP}(x_i) = \sum_{x \in \Omega|x_i} \text{APP}(x) \propto \sum_{x \in \Omega|x_i} \text{obs}(x) \prod_{\ell=1}^{n_t} \pi(x_{\ell}) \propto \pi(x_i) \text{Extr}(x_i)
\]

- The a posteriori information for binary elements is

\[
\text{APP}(b_j) \propto \pi(b_j) \sum_{x \in \Omega|b_j} \text{obs}(x) \prod_{\ell \neq j} \pi(b_{\ell}) \underbrace{\text{Extr}(b_j)}
\]
Consider a chip synchronous code division multiple access channel for $K$ users.

Symbols $x_i$ belong to a linear modulation, e.g. $x_i = \pm 1$.

Channel gain coefficient for user $i$ is $\omega_i$.

Equations for APP multiuser (CDMA) detection are similar to those encountered in MIMO detection.
Iterative APP joint detection in CDMA with $K = 4$ users on a gaussian channel. The NRNSC code is a rate 1/4 16-states (25,27,33,37) for all users. Same SNR per bit for all users. Each user pseudo-randomly interleaves its $N = 8192$ bits before transmitting on the multiple access channel. No PN spreading. System load is 100%.
Conclusions

- Concatenating simple elementary codes leads to powerful compound codes.
- Iterative decoding/detection is an efficient tool in compound coding systems.
- Iterative processing of information opens a new era in coding and communications.
- Future telecommunications products will benefit from iterative probabilistic processing in order to boost performance and to gain in flexibility and compatibility.

Supplementary references have been cited by the speaker during the talk.