General Introduction to Error-correcting Codes

Eric Filiol

ESIEA - Laval
Laboratoire de cryptologie et de virologie opérationnelles
$(C+V)^O$
filiol@esiea.fr

2013 - 2014

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Introduction

From the Information Theory course we have learnt that :

- Shannon's theorem for noisy coding proves that we always can have a maximal transinformation (or a minimal residual error probability after decoding).
- It is therefore possible to manage noise over communication reliably.
- Existence theorem only : Shannon's tells us only that such efficient codes exist.
- Shannon's theory gave birth to the error-correcting theory (how to build such codes practically) that will be exposed in the present lecture.
- Shannon's noisy coding theory assumes that the noise is neither adaptative nor malicious (safety issue).
- Error-correcting techniques applies to a wider class of problems (safety, reliability and even security for example when dealing with DDoS or cryptography).

Formalization of the Coding Problem

- Wlog, we assume that the channel has the same input and output alphabet Σ, of size q (binary when $p = 2$, ternary if $p = 3$).
- A *code* $\mathcal{C}$ over Σ is a collection of sequences of symbols from Σ. These sequences are called *codewords*.
- We assume that all codewords are of the same length (easier for decoding ; valid assumption from Shannon's second theorem). We then consider *block codes*.
- If codewords of $\mathcal{C}$ have length n, then the code is described as a *q-ary code of length n*.
- Let V_n denotes the set of all n-sequences of symbols from Σ and call the elements of $V_n(\Sigma) = V_n(q)$ *vectors* or *words*. We note V_n whenever $q = 2$.
- The numbers of codewords $A_q(n, d)$ play a central role in coding theory. The main coding problem lies in determining these numbers.

Discrete Topology & Coding Problem

- The ball of radius ρ centered on $c \in V_n$ is $\mathcal{B}(c, \rho) = \{y \in V_n | d(y, c) < \rho\}$. We call it a sphere when $d(y, c) \leq \rho$.
- The volume of a ball of radius ρ centered on $c \in V_n$ is the number of points that it contains :

$$|\mathcal{B}(c, \rho)| = \sum_{k=0}^{\rho} \binom{n}{k}$$

- This volume is clearly independent from the center c and we will therefore note it by $\mathcal{V}(n, \rho)$.

Packing and Covering Radius

The *packing radius* of a code $\mathcal{C}$ is the largest integer $\rho_{\mathcal{C}}$ for which the spheres centered at each codeword c are disjoint. The *covering radius* of $\mathcal{C}$ is the smallest integer $\rho'_{\mathcal{C}}$ for which the spheres centered at each codeword c cover $V_q(n)$. We note them $pr(\mathcal{C})$ and $cr(\mathcal{C})$ respectively.

The Coding Problem : Error-detection and Error-correction

Error-detection

A code $\mathcal{C}$ is e*-error detecting* if, whenever no more than e symbols in a codeword c are altered, the received word is not a different codeword c'. This means that we can tell whether there have been errors, provided that there are at most e of them.

Error-detection

A code $\mathcal{C}$ is e*-error correcting* if, whenever no more than e symbols in a codeword c are altered, the received word is still decoded using the minimum distance decoding rule.

- Give examples of error-detecting and error-correcting codes.

The Coding Problem : Hamming Distance

- Let V_n denotes the set of all binary n-sequences (it is thus a n-dimensional vector space over $\mathbb{F}_2$. If x and y are two vectors in V_n, we define the *Hamming distance* $d(x, y)$ between x and y as $|\{i \in [1, n] | x_i \neq y_i\}|$.
- *Minimum-distance* or *nearest-neighbour* decoding means decoding a received vector y into a codeword c that is a minimum distance from y.
- We decode any received vector y into a vector c that is a minimum Hamming distance from y. If there are more than one such codeword, we choose arbitrarily.
- If $\mathcal{C}$ is a code, the *minimum distance* of $\mathcal{C}$ is defined by

$$d(\mathcal{C}) = \min_{i,j}\{d(c_i, c_j)\}$$

- Good codes have their codewords scattered so that their minimum distance is large.

Error-detecting & Error-correcting

Theorem

Let $\mathcal{C}$ be a code with minimum distance $d > 0$.

- $\mathcal{C}$ is $(d-1)$-error detecting and not d-error detecting.
- $\mathcal{C}$ is $\lfloor\frac{1}{2}(d-1)\rfloor$-error correcting but not $\lfloor\frac{1}{2}(d+1)\rfloor$-error correcting.

- Proof left as an exercice.
- If a code has M codewords of length n and has minimum distance d, then it is called a (n, M, d) code.
- Increasing M tends to decrease d and conversely.
- We must also observe that a code $\mathcal{C}$ is e-error correcting precisely when the balls $\mathcal{B}(c, e+1)$ for $c \in \mathcal{C}$ are disjoint.

Error-detecting & Error-correcting (2)

Theorem

The packing radius of an (n, M, d)-code is $pr(\mathcal{C}) = \lfloor \frac{d-1}{2} \rfloor$. Assuming that ties are always reported as errors, a code $\mathcal{C}$ is exactly e-error correcting if and only $pr(\mathcal{C}) = e$.

Theorem

A code $\mathcal{C}$ is simultaneously e-error correcting and e'-error detecting if and only if $d \geq e + \frac{1}{2}.e' + 1$.

- Proof left as exercices.

Error-detecting & Error-correcting (2)

- It is intuitively clear that given a code $\mathcal{C}$ we may add new codewords at no cost to its minimum distance.

Definition

An (n, M, d)-code is said to be *maximal* if it is not contained in any larger code with the same minimum distance, that is, if it is not contained in any $(n, M+1, d)$-code.

Theorem (decoding error)

For the BSC with error probability p using the minimum distance decoding rule, the probability of decoding error for maximal (n, M, d)-code satisfies

$$\sum_{k=d}^{n} \binom{n}{k} p^k (1-p)^{n-k} \leq P(error) \leq 1 - \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{k} p^k (1-p)^{n-k}$$

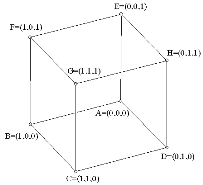
For a non maximal code, the uppr bound still holds but the lower bound may not.

Enumerating Codewords

- Let $A_q(n, d)$ denote the maximum M such that there exists a q-ary (n, M, d) code.
- Clearly by taking the code all all n vectors, we have $A_q(n, 1) = q^n$.
- For even relatively small values of q, n and d, the quantity $A_q(n, d)$ is unknown and we can only provide bounds.

$$72 \leq A_2(10, 3) \leq 79 \qquad 144 \leq A_2(11, 3) \leq 158$$

- Determining $A_2(n, d)$ is a finite geometry (intractable) problem.



Give $A_2(3, 2)$ and the corresponding $(3, 4, 2)$ code

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Introduction : Equivalent Codes

- Most of the mathematical problems underlying the construction of good codes are generally intractable. The notion of equivalence of codes is useful to reduce the amount of work involved in this research.
- Suppose that we have a (n, M, d)-code $\mathcal{C}$. We can represent it as a $M \times n$ matrix whose rows are the distinct codewords.
- Let us consider a permutation π of $(1, 2, , \ldots, n)$ and that for any $c \in \mathcal{C}$ we apply $\pi : c \mapsto c'$ (*positional permutation*) defined by $c'_i = c_{\pi(i)}$.
- If π is a permutation of the symbols of Σ, we say that π induces a *symbol permutation* of $\mathcal{C}$ if, for some $i, (1 \leq i \leq n)$ and for each codeword $c \in \mathcal{C}$ we transform c into c' where c' is identified by

$$c'_j = c_j \quad (1 \leq j \leq n, j \neq i), \qquad c'_i = \pi(c_i)$$

- If a code $\mathcal{C}'$ can be obtained from a code $\mathcal{C}$ by a sequence of positional or symbol permutations, then $\mathcal{C}'$ is called an equivalent code.

Equivalent Codes : Examples

- Take a code $\mathcal{C}$ of length 5 over $\Sigma = \{a, b, c\}$ having codewords c_1, c_2, c_3, c_4 (left).
- We apply a shift (positional) permutation $\{1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 4, 4 \mapsto 5, 5 \mapsto 1\}$ to obtain the code $\mathcal{C}'$ (center).
- We apply a (symbol) permutation $\{a \mapsto b, b \mapsto c, c \mapsto a\}$ to the first symbol of each word of $\mathcal{C}'$ to obtain the code $\mathcal{C}''$ (right).

$$\begin{bmatrix} a & b & c & a & c \\ b & a & b & a & b \\ b & c & c & b & a \\ c & b & a & c & a \end{bmatrix} \quad \begin{bmatrix} c & a & b & c & a \\ b & b & a & b & a \\ a & b & c & c & b \\ a & c & b & a & c \end{bmatrix} \quad \begin{bmatrix} a & a & b & c & a \\ c & b & a & b & a \\ b & b & c & c & b \\ b & c & b & a & c \end{bmatrix}$$

Equivalent Codes : Results

Theorem

If $\mathcal{C}$ and $\mathcal{C}'$ are equivalent codes, then the set of distances between the codewords of $\mathcal{C}$ is identical with that between the codewords of $\mathcal{C}'$.

Theorem

If $\mathcal{C}$ is any code of length n and u is any n-vector over the same alphabet, then there exists a code $\mathcal{C}'$ that contains u and is equivalent to $\mathcal{C}$.

- Proofs left as exercices (hint : you can transform any codeword c_i into u by at most n symbol permutations).
- The knowledge of at least one code enables to build other (equivalent) codes by means of simple permutations (remind that there are $n!$ permutations over a set of n objects).

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Sphere Packing

- Since estimating $A_q(n,d)$ is generally a difficult problem, we are going to look for bounds. We are going to consider sphere packing techniques (*sphere packing* deals with arrangement of non-overlapping spheres within a containing space in such a way that the spheres fill as large a proportion of the space as possible).

Theorem : Sphere Packing Upper Bound (Hamming's bound)

Let $\mathcal{C}$ be a code. When d is odd, say $d = 2e + 1$ then

$$A_q(n,d) \sum_{k=0}^{e} \binom{n}{k} (q-1)^k \leq q^n$$

- Proof left as an exercice.

Gilbert-Varshamov Bound

Theorem : Gilbert-Varshamov Bound

Let $\mathcal{C}$ be a (n, M, d)-code. Then

$$A_q(n,d) \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i \geq q^n$$

- Proof left as an exercice.
- So by applying both bounds we have

$$\frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i} \leq A_q(n,d) \leq \frac{q^n}{\sum_{k=0}^{e} \binom{n}{k} (q-1)^k}$$

Discrete Topology Vision

- We can describe the previous results in terms of ball volumes $V(n, \rho)$.

Theorem (Binary Case)

If $\mathcal{C}$ is an e-error correcting code, the number M of codewords must satisfy

$$M \leq \frac{2^n}{V(n, e+1)}$$

- Proof left as an exercice.

Sphere-packing Bound Equality

A code $\mathcal{C}$ attains the (Hamming) sphere-packing bound if and only if every word in $V_q(n)$ lies at distance e or smaller from exactly one word in $\mathcal{C}$.

Plotkin Bound

Theorem : Plotkin Bound

Let $\theta = 1 - \frac{1}{q}$ and suppose that $d > \theta n$. Then a q-ary code (n, M, d) is such that $A_q(n, d) \leq \frac{d}{d - \theta n}$. This code attains the Plotkin bound if and only if

- any two codewords have distance d, and,
- each symbol occurs in a given position in the same number $\frac{M}{q}$ codewords.

- The proof can be omitted.
- Orthogonal arrays of strength t and index λ are combinatorial objects to build code attaining the Plotkin bound $(t = 1)$.

Singleton Bound

Theorem

$A_q(n,d) \leq q^{n-d+1}$ codewords. Equality holds if and only if, given any $n-d+1$ coordinate positions and any $n-d+1$ symbols from Σ, there is a unique codeword having those symbols in those positions.

- Proofs left as exercices.
- Codes satisfying the equality are called *Maximum Distance Separable* (MDS).
- Orthogonal arrays of strength t and index λ are combinatorial objects to build code attaining the Singleton bound ($t = n - d + 1$ and $\lambda = 1$).

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Perfect Codes

- The ideal situation (for decoding in an economic and efficient way) is to have a code $\mathcal{C}$ over $V_n(q)$ such that for some $\rho > 0$, the ρ-spheres around the codewords of $\mathcal{C}$ are a partition of $V_n(q)$. Such a code is called a *perfect code*.
- From the definition, it is clear that such a code
  - can correct up to ρ errors,
  - cannot correct $\rho + 1$ errors by minimum distance decoding.
- A trivial perfect code is that with exactly one codeword.

Perfect Codes

Theorem : Perfect Codes (sphere-packing condition)

A necessary condition for an (n, M, d)-code to be perfect is that d is odd. Moreover, an (n, M, d)-code is perfect if and only if d is odd and attains the sphere-packing bound. In other words

$$M. \sum_{i=0}^{\frac{1}{2}(d-1)} \binom{n}{i} (q-1)^i = q^n$$

Theorem : Perfect Codes (2)

A q-ary (n, M, d)-code $\mathcal{C}$ is perfect if $pr(\mathcal{C}) = cr(\mathcal{C})$.

$$M. \sum_{i=0}^{\frac{1}{2}(d-1)} \binom{n}{i} (q-1)^i = q^n$$

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Introduction

- Constructing codes may be really hard and the issue of determining $A_q(n, d)$ is at the center of this difficulty.
- There are several useful techniques that can be used to build new codes from old codes, and therefore we can lessen this difficulty.
- Wlog we assume that our working alphabet is $\Sigma = \mathbb{Z}/q\mathbb{Z}$.
- We already have presented the concept of code equivalence.
- We are going to explore the techniques of *code extension*, *code puncturing*, *code shortening*, *code augmenting*, *direct sum construction*.

Extending a Code

- We add one or more positions to all codewords, thus increasing n.
- The most common way consists in adding an *overall parity check*.
- If $\mathcal{C}$ is a q-ary (n, M, d)-code, we define the extended code $\overline{\mathcal{C}}$ code by

$$\overline{\mathcal{C}} = \{x_1 x_2 \ldots x_n x_{n+1} | x_1 x_2 \ldots x_n \in \mathcal{C} \text{ and } \sum_{k=1}^{n+1} c_k \equiv 0 \mod q\}$$

- The extended code $\overline{\mathcal{C}}$ is then a $(n+1, M, d)$- or a $(n+1, M, d+1)$-code.
- Exercice : prove that the minimum distance of $\overline{\mathcal{C}}$ depends on the parity of d only.

Puncturing a Code

- We remove one or more positions to all codewords, thus decreasing n.
- This is often considered for bandwith optimization.
- If $\mathcal{C}$ is a q-ary (n, M, d)-code and if $d \geq 2$, we obtain $\mathcal{C}^*$ by puncturing $\mathcal{C}$ once. Then $\mathcal{C}^*$ has parameters $(n-1, N, d)$ or $(n-1, N, d-1)$

Lemma

A binary $(n, M, 2e+1)$-code exists if and only if a binary $(n+1, M, 2e+2)$-code exists.

- Proof : use the extension or puncturing technique.

Shortening a Code

- We keep only those codewords in a code that have a given symbol in a given position, and then deleting that position.
- If $\mathcal{C}$ is a (n, M, d)-code, then a shortened code has length $n-1$ and minimum distance at least d.
- Shortening a code can result in a substantial increase in the minimum distance but does result in a code with smaller size.
- Shortening a code by taking codewords with an 's' in the i-th position is referred to as the *cross-section* $x_i = s$.

Augmenting a Code

- *Augmenting* a code simply means adding additional strings (codewords) to the code.
- A common way consists in including the complements of each codeword in $\mathcal{C}$, where the complement of a binary codeword c is the string obtained by $c \oplus 0x1111\ldots1111$.
- Let us denote the complement of c by c^c and denote the set of all complements of the codewords in $\mathcal{C}$ by $\mathcal{C}^c$. It is easy to check that if $(x, y) \in V(n)$, then $d(x, y^c) = n - d(x, y)$.

Proposition

Let $\mathcal{C}$ be a (n, M, d)-code. Suppose that d' is the maximum distance between codewords in $\mathcal{C}$. Then $d(\mathcal{C} \cup \mathcal{C}^c) = \min\{d, n - d'\}$.

Direct Sum and $u(u+v)$ Constructions

- If $\mathcal{C}_1$ if a q-ary (n_1, M_1, d_1)-code and $\mathcal{C}_2$ if a q-ary (n_2, M_2, d_2)-code, the *direct sum* $\mathcal{C}_1 \odot \mathcal{C}_2$ is the code

$$\mathcal{C}_1 \odot \mathcal{C}_2 = \{xy | x \in \mathcal{C}_1, y \in \mathcal{C}_2\}$$

- Clearly the resulting code $\mathcal{C}_1 \odot \mathcal{C}_2$ has parameters $(n_1 + n_2, M_1.M_2, \min\{d_1, d_2\})$.
- A much more useful construction than is the following. From previous codes $\mathcal{C}_1$ and $\mathcal{C}_2$, we define a code $\mathcal{C}_1 \oplus \mathcal{C}_2$ by

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{x(x+y) | x \in \mathcal{C}_1, y \in \mathcal{C}_2\}$$

Lemma

The code $\mathcal{C}_1 \oplus \mathcal{C}_2$ has parameters $(2n, M_1.M_2, d')$ where $d' = \min\{2d_1, d_2\}$.

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Conclusion

- We have defined what a code is more precisely (a subset of the Hamming space).
- We have sketched the principle of decoding (using the minimum distance decoding rule).
- We have defined bounds on the number of codewords. Still many open problems exist regarding this issue.
- We have now all required tools to explore different families of codes.
- Go now to the computer room to practice with exercices.

Agenda

1. Introduction : The Coding Problem
2. Equivalent Codes
3. Sphere Packing, Gilbert-Varshamov and Other Bounds
4. Perfect Codes
5. Making New Codes from Old Ones
6. Conclusion
7. Bibliography

Essential Bibliography

A few papers are available on the Moodle repository for this lecture.

- Cameron, P. J. (1996). *Combinatorics : Topics, Techniques and Algorithms*. Cambridge University Press.
- Hamming, R. W. (1950). Error Detecting and Error correcting Codes. *Bell. Syst. tech. J.*, 29, pp. 147–160.
- MacWilliams, F. J. & Sloane, N. J. A. (1977). *The Theory of Error-Correcting Codes*. North-Holland.
- Plotkin, M. (1960). Binary Codes with Specified Minimum Distance. *IEEE Trans. Info. Theory*, 6, pp. 445–450.
- Shannon, C.E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656.
- Singleton, R.C. (1964). Maximum Distance q-ary Codes. *IEEE Trans. Info. Theory*, 10, pp.116-118.
- Welsh, D. (1988). *Codes and Cryptography*, Oxford Science Publishing.