PHY432, Complément de poly associé à l'Amphi7
Théorie de l'Information et Physique Statistique

Table des matières

1 Introduction

Longtemps cantonnée dans des cercles académiques, la théorie de l'information a fait son entrée dans notre quotidien : nul n'ignore plus le nombre d'octets utilisés pour mettre en mémoire sa photo ou son morceau de musique préférés, ni même l'existence de différents formats de compression.

C'est dans l'immédiat après-guerre que Shannon énonce les notions fondamentales de la théorie de l'information. Dès le début, le lien étroit de ces notions avec la physique statistique a été évident, puisque l'un des ingrédients essentiels, la notion de quantité d'information, est directement lié à l'entropie. Les échanges entre ces deux disciplines ont été très fructueux dans les deux sens, et ont acquis une nouvelle vitalité dans les dix dernières années, avec du côté théorie de l'information le développement de nouveaux codes de correction d'erreurs, comme les turbocodes et codes LDPC, et du côté physique les progrès dans l'étude des systèmes désordonnés comme les verres de spin. Ils ouvrent

par ailleurs de nombreuses perspectives dans d'autres branches de la science, en particulier en biologie.

Ce chapitre va explorer quelques unes de ces relations dans ses paragraphes 2 et 3. Il est complété par l'appendice I (officiellement 'hors programme') qui explique plus en détail comment les méthodes de champ moyen permettent de décoder des erreurs de transmission, et l'appendice II sur la méthode des multiplicateurs de Lagrange. Le lecteur curieux d'en savoir plus pourra consulter les ouvrages suivants :

- 'Elements of Information Theory' par Thomas M. Cover et Joy A. Thomas, Wiley 1991
- 'Information Theory, Inference & Learning Algorithms' par David J. C. MacKay, Cambridge University Press 2002
- 'Information, Physics, and Computation' par Marc Mézard et Andrea Montanari, Oxford University Press 2009
- Maxwell's Demon 2, H.S. Leff and A.F. Rex Eds, IOP Publising 2003
- 'Physique Statistique' par Roger Balian, Cours de l'Ecole Polytechnique , Ellipses 1982.

2 Information et entropie

2.1 Information de Shannon

Pour quantifier la notion d'information, Shannon utilise l'idée suivante : supposons une source d'information, qui peut engendrer un nombre fini N de différents mots, le mot numéro n apparaissant avec probabilité p_n. Comment quantifier notre degré d'incertitude sur le prochain mot qui va sortir de cette source ? Si tous les évenements sont équiprobables (donc si $p_n = 1/N$), l'incertitude doit évidemment augmenter avec N. Par ailleurs si les mots sont engendrés en deux temps, commençant par un choix d'une certaine famille de mots, suivi par le choix d'un mot au sein de la famille, notre incertitude devrait être la somme de l'incertitude liée au choix de la famille, plus celle liée au choix du mot dans la famille. Remarquablement, ces deux hypothèses de bon sens, jointes à une condition naturelle de continuité, imposent le résultat. On peut en effet en déduire, après quelques calculs, que l'incertitude de la source est donnée par

$$H = -\sum_n p_n \log_2 p_n \ . \tag{1}$$

Le choix de la notation H, habituel en théorie de l'information, interfère malheureusement avec l'hamiltonien en physique statistique, mais le lecteur attentif ne devrait pas faire la confusion entre ces deux objets très différents. L'unique arbitraire est un choix d'unité de mesure, correspondant au choix de la base du logarithme. Le choix habituel du logarithme en base 2 correspond à une mesure de l'incertitude en 'binary digits', ou 'bits'. Cette mesure d'incertitude est aussi une mesure de l'information qui nous manque pour connaître le mot envoyé, et induit donc une définition de la quantité d'information d'une source aléatoire. Si par exemple on reçoit un mot, ce mot nous apporte une certaine quantité d'information. En moyenne, cette quantité d'information n'est autre que H.

Cette définition est directement reliée à l'entropie, puisque nous avons vu que, dans tous les ensembles, canonique, microcanonique, grand-canonique... l'entropie est toujours donnée par $S = -k \sum_n p_n \log p_n$. L'unique différence est le choix de l'unité de mesure : le choix de la physique consiste à prendre la constante de Boltzmann $k = 1.3\ 10^{-23} J/K$, ce qui permet de retrouver exactement l'entropie définie par les thermodynamiciens. Mais il est clair qu'il s'agit bien de la même quantité, mesurée dans des échelles différentes.

La définition de Shannon de l'information possède des propriétés sympathiques, dont les démonstrations, simples, sont laissées au lecteur :

1. $H \geq 0$; $H = 0$ si et seulement si $p_n = \delta_{n,n_0}$: si la source émet toujours le même mot n_0, le fait de recevoir le mot n_0 ne nous apporte aucune information.
2. Si le nombre de mots possibles est N, la loi qui maximise H est la loi uniforme : $p_n = 1/N$. Son entropie est $\log_2 N$.
3. Supposons que chaque mot soit en fait un mot composé formé d'une paire de mots n, a qui sont indépendants : $P(n, a) = p_n q_a$ (par exemple n pourrait être le numéro gagnant au loto, et a le numéro du cheval gagnant dans la troisième course...). Alors : $H(\{p_n q_a\}) = H(\{p_n\}) + H(\{q_a\})$, l'entropie est la somme des entropies de chaque loi.
4. Dans le cas de mots composés associés à des événements non indépendants (par exemple n étant le cheval gagnant et a le cheval classé deuxième), avec donc une probabilité $P(n, a)$ qui n'est pas factorisée, on a le résultat suivant. Considérons les lois marginales : $p_n = \sum_a P(n, a)$; $q_a = \sum_n P(n, a)$. Alors on démontre facilement que l'entropie de la loi composée est plus petite que la somme des entropies de chacune des lois marginales : $H = -\sum_{n,a} P(n, a) \log_2 P(n, a) \leq H(\{p_n\}) + H(\{q_a\})$. En particulier, si les évènements n et a sont complètement corrélés (le cheval gagnant et le cheval arrivant deuxième dans le cas où il n'y a que deux chevaux), l'information de la loi jointe $P(n, a)$ est égale à celle d'une des lois marginales (si $N = 2$ et que l'on connaît le gagnant, on ne gagne aucune information à donner le numéro du cheval arrivé deuxième).
5. Additivité. Supposons que l'ensemble des mots possibles $\mathcal{X}$ soit décomposé en $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, avec $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$. Appelons $q_1 = \sum_{n \in \mathcal{X}_1} p_n$ la probabilité d'envoyer un mot de $\mathcal{X}_1$, et q_2 la probabilité d'un mot de $\mathcal{X}_2$. Pour chaque mot $n \in \mathcal{X}_1$, définissons comme d'habitude la probabilité conditionnelle de n, conditionnée au fait que $n \in \mathcal{X}_1$, par $r_n = p_n/q_1$ et définissons de même par $s_n = p_n/q_2$ la probabilité conditionnelle des mots $n \in \mathcal{X}_2$. Alors l'entropie totale s'écrit comme $H = -\sum_{n \in \mathcal{X}} p_n \log_2 p_n = H(q) + H(r) + H(s)$, où :

$$H(q) = -q_1 \log_2 q_1 - q_2 \log_2 q_2 \quad (2)$$

$$H(r) = -q_1 \sum_{n \in \mathcal{X}_1} r_n \log_2 r_n \quad (3)$$

$$H(s) = -q_2 \sum_{n \in \mathcal{X}_2} s_n \log_2 s_n \quad (4)$$

Cette propriété s'interprète ainsi. L'information associée au choix d'un mot n est additive ; c'est la somme de l'information associée au choix de l'un des deux sous-

ensembles $\mathcal{X}_1$, $\mathcal{X}_2$, plus l'information associée au choix du mot à l'intérieur du sous-ensemble (pondérée par la probabilité de chacun des sous-ensembles). C'est une propriété essentielle de l'entropie, qui justifie son choix comme mesure de l'information. En fait on peut montrer (cf par exemple le livre de Balian) que l'entropie de Shannon est la seule mesure d'information, fonction continue des p_n, possèdant cette propriété : c'est ainsi qu'elle a été découverte.

2.2 Physique statistique et maximisation de l'information

Nous avons construit la physique statistique à partir du postulat microcanonique : tous les microétats d'énergie E sont équiprobables, et l'entropie est $S = k \log W$, où W est le nombre de ces microétats d'énergie E. Il existe en fait une introduction alternative élégante de la physique statistique, qui part justement de la théorie de l'information. L'idée est la suivante : nous cherchons à trouver la loi p_n donnant la probabilité de chacun des états du système, et nous avons quelques éléments d'information sur ce système (par exemple nous pouvons savoir qu'il est isolé, ou bien qu'il échange de l'énergie avec un thermostat...). Le principe utilisé est de choisir pour p_n la loi qui maximise l'information manquante H, compte tenu des contraintes. L'idée centrale est que l'on doit énumérer toutes les informations dont nous disposons sur le système, qui peuvent être considérées comme autant de *contraintes* a priori sur la loi p_n, puis choisir la loi $\{p_n\}$ la moins "biaisée" possible compte tenu de ces contraintes, c'est-à-dire la loi qui *maximise l'information manquante*. C'est un principe, et comme tel, il ne peut être vraiment justifié que par la pertinence de ce qu'on en déduit. Voyons tout de suite ce qu'il en est pour différents cas de contraintes :

– Système isolé : seuls les états d'énergie E sont accessibles, donc $p_n = 0$ si $E_n \neq E$ (comme toujours il faudrait, plus précisément, introduire une petite résolution δE en énergie, petite à l'échelle macroscopique et grande à l'échelle microscopique...). La loi qui maximise H sous ces contraintes est la loi p_n uniforme sur tous les états accessibles, on retrouve bien l'ensemble microcanonique. Ce résultat étant important, regardons plus précisément comment le démontrer. Nous restreignons la discussion à tous les états n ayant la bonne énergie $E_n = E$, et nous cherchons la loi de probabilité p_n qui maximise $H = -\sum_n p_n \log_2 p_n$. Les contraintes à prendre en compte sont la positivité, $p_n \geq 0$, et la normalisation, $\sum_n p_n = 1$. La méthode employée pour faire cette maximisation sous contrainte est sans doute déjà connue du lecteur (elle intervient dans les cours d'analyse numérique et d'économie) : il s'agit de la méthode des *multiplicateurs de Lagrange*. Nous en résumons le principe dans l'appendice II. Dans notre cas, nous devons maximiser H sous la contrainte de normalisation globale : $\sum_n p_n = 1$. Pour cela, nous introduisons un multiplicateur de Lagrange λ et considérons la fonction :

$$G[\{p_n\}, \lambda] \equiv -\sum_n p_n \log_2 p_n - \lambda(\sum_n p_n - 1) \tag{5}$$

Et nous cherchons les extrema $\{p_n^*\}, \lambda^*$ de G :
* $\frac{\partial G}{\partial \lambda} = 0$ conduit à : $\sum_n p_n^* = 1$.
* $\frac{\partial G}{\partial p_n} = 0$ conduit à : $p_n^* = e^{-1-\lambda \ln 2}$.

Nous démontrons ainsi que H est bien maximale lorsque *tous les* p_n *sont égaux*. La somme intervenant dans la contrainte de normalisation fait intervenir les $W(E)$ états et nous obtenons ainsi :

$$p_n^* = \frac{1}{W(E)} \ , \tag{6}$$

c'est bien l'ensemble microcanonique.

– Système dont l'énergie interne est fixée. Tous les états sont accessibles, mais la loi p_n doit être telle que $\sum_n p_n E_n = U$. Nous introduisons deux multiplicateurs de Lagrange, l'un pour la contrainte $\sum_n p_n = 1$, l'autre pour $\sum_n p_n E_n = U$. Nous considérons donc la fonction

$$G[\{p_n\}, \lambda_1, \lambda_2] \equiv -\sum_n p_n \log_2 p_n - \lambda_1 (\sum_n p_n - 1) - \lambda_2 (\sum_n p_n E_n - U) \tag{7}$$

Et nous cherchons les extrema $\{p_n^*\}, \lambda_1^*, \lambda_2^*$ de G :
* $\frac{\partial G}{\partial \lambda_1} = 0$ conduit à : $\sum_n p_n^* = 1$.
* $\frac{\partial G}{\partial \lambda_2} = 0$ conduit à : $\sum_n p_n^* E_n = U$.
* $\frac{\partial G}{\partial p_n} = 0$ conduit à : $p_n^* = e^{-1-\lambda_1 \ln 2 - \lambda_2 E_n \ln 2}$.

Nous trouvons donc bien une loi de Boltzmann, du type $p_n = (1/Z)\ e^{-\beta E_n}$, dans laquelle $\beta = \lambda_2 \ln 2$ est fixé par la condition $\sum_n p_n^* E_n = U$: c'est l'ensemble canonique.

Le lecteur trouvera facilement de la même façon l'ensemble grand-canonique (on doit fixer $\sum_n p_n = 1$, $\sum_n p_n E_n = U$ et ainsi que le nombre moyen de particules $\sum_n N_n = N$), ou des ensembles généralisés dans lesquels on fixe le volume moyen par exemple.

2.3 Entropie et information : démon de Maxwell

Un cas très intéressant dans lequel le contact entre la notion d'entropie physique et celle d'information joue un rôle important est l'analyse du démon de Maxwell. En 1871, Maxwell étudiait la possibilité éventuelle d'extraire du travail en utilisant une seule source de chaleur. Son idée était de prendre un récipient séparé en deux parties, avec une petite trappe les mettant en communication, actionnée par un 'démon'. Si le démon pouvait sélectionner les molécules les plus rapides et manipuler la trappe pour les laisser passer dans un seul sens, il réussirait à créer un déséquilibre de température entre les deux moitiés du récipient. Il existe de nombreuses versions de tels démons, qui ont fasciné les physiciens depuis cette époque. Deux exemples sont donnés en Fig.1. La figure de droite montre en particulier un cas où on n'a pas besoin d'invoquer un être microscopique regardant les particules : une simple trappe avec un ressort permet de sélectionner les particules allant de la droite vers la gauche. Ceci crée un déséquilibre de pression qui permet d'engendrer du travail. Fondamentalement, chacun de ces systèmes s'est avéré être finalement une illusion : il n'y a pas de machine thermique fonctionnant avec une seule source de chaleur. L'exemple de la trappe est instructif. Pour fonctionner, la trappe doit être microscopique, avec un ressort de très faible raideur de telle sorte qu'elle puisse s'ouvrir lorsqu'une particule arrive de la droite avec suffisamment d'énergie. Pour que la trappe n'entre pas dans un mouvement oscillant, mais qu'elle se referme bel et bien après chaque passage de molécule, il faut qu'il y ait un système d'amortissement, donc dissipation d'énergie. La trappe

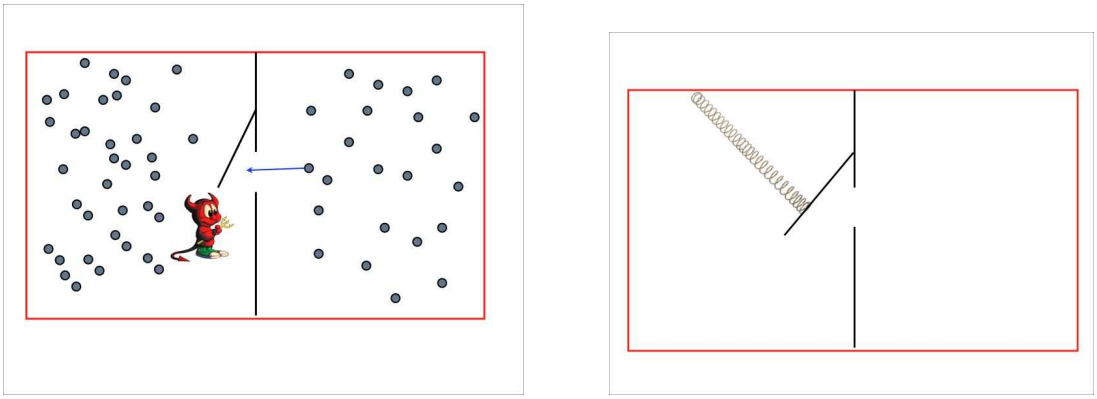


FIGURE 1 – Deux exemples de démons de Maxwell. Figure de gauche : un démon observe les particules et ouvre la trappe seulement quand une particule passe dedroite à gauche. Figure de droite : une trappe automatique

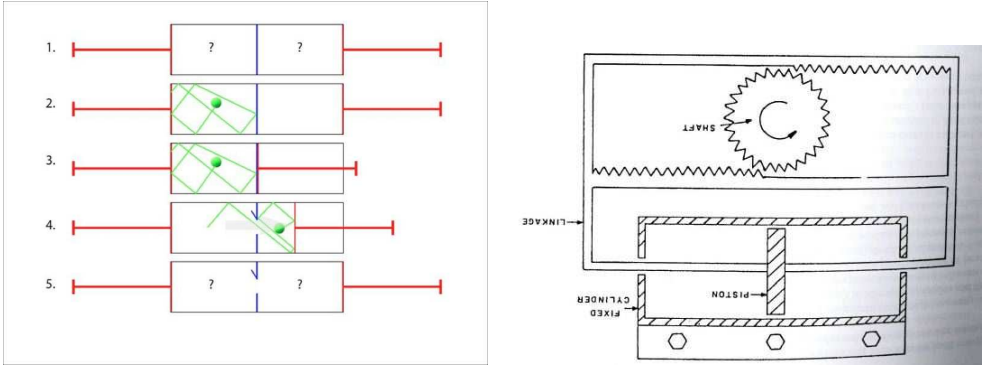


FIGURE 2 – Figure de gauche : la machine de Szilard. Figure de droite : un raffinement de la machine de Szilard, sans observateur.

va donc nécessairement s'équilibrer avec le monde environnant. A l'équilibre à la température T, elle va avoir des fluctuations, un mouvement Brownien, et donc des ouvertures spontanées, laissant passer des particules dans les deux sens. Une étude détaillée montre que ce mécanisme de trappe ne peut pas fonctionner, et il en est de même des nombreux types de cliquets qui ont été étudiés. L'analyse d'un démon ouvrant la trappe seulement au bon moment a nécessité une analyse plus fine, dûe à Smoluchovski et Brillouin. Le démon doit voir les molécules, donc il doit envoyer des photons, avec une longueur d'onde pas trop grande, donc avec une énergie minimale. Le photon doit être diffusé par la molécule et absorbé par un dispositif de détection, avec dissipation d'énergie. L'nalyse détaillée, dans laquelle nous ne rentrerons pas, a permis de nouveau de montrer que le démon basé sur la diffusion de photon ne fonctionne pas.

Pour aller au delà de l'analyse détaillée au cas par cas, il faut prendre en compte la notion d'information. Cette démarche, initiée par Landauer, a permis de donner une explication universelle à l'impossibilité de telles machines.

Cette notion s'étudie particulièrement bien sur le cas de la machine de Szilard (cf Fig.2). Il s'agit d'une machine idéale fonctionnant avec une seule particule. On détecte

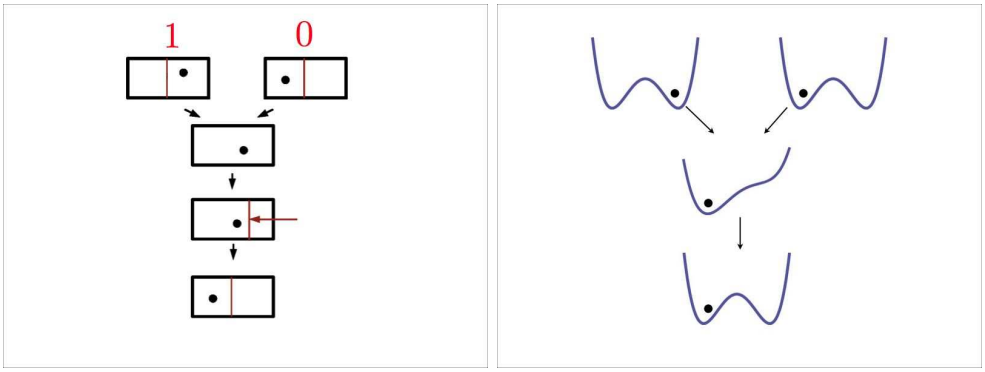
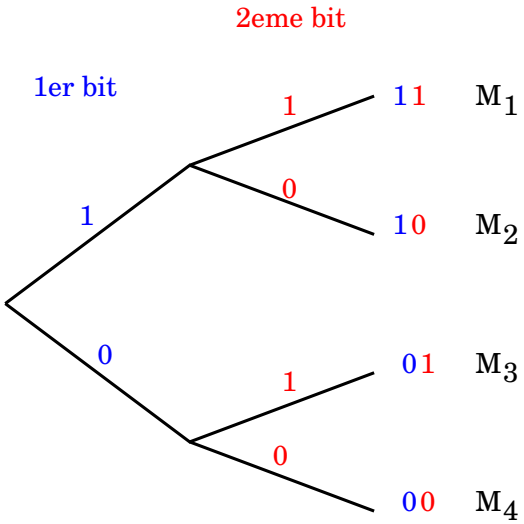


FIGURE 3 – Figure de gauche : Remise à zéro d'un bit d'information dans la boîte de Szilard. Figure de droite : Remise à zéro d'un bit d'information dans un double puits.

si la particule est à droite ou à gauche, et on agit en conséquence. Si elle est à gauche, on amène un piston de la droite, sans travail (étape 3), on ouvre la partition au milieu de la boîte, et on ramène très doucement le piston vers sa position initiale (étape 4). La particule exerce une pression sur le piston et donc elle fournit du travail. Son énergie reste fixée en moyenne car elle reçoit de la chaleur du monde extérieur, compensant le travail fourni. Regardons quel est le travail fourni par la machine pendant l'étape 4. On supposera qu'à l'instant t le volume de la boîte est $V(t) = SL(t)$ où $L(t)$ est la longueur et S est la section. La longueur double au cours de ce processus. Le système est toujours à l'équilibre à le température T, donc la pression exercée par la particule vaut $p(t) = \frac{kT}{V(t)}$ et la force moyenne qu'elle exerce sur le piston à l'instant t vaut $\langle F(t)\rangle = \frac{kT}{V(t)}S = \frac{kT}{L(t)}$. Le travail reçu par le piston entre t et $t + dt$ vaut donc $dW(t) = \langle F(t)\rangle dL = kT\frac{dL}{L(t)}$ et le travail total reçu au cours de l'étape 4 vaut $kT\int_{L_0}^{2L_0}\frac{dL}{L} = kT\ln 2$. Voici donc une machine qui fournit un travail $kT\ln 2$ à chaque cycle, mais elle utilise toujours un démon dans l'étape 1. Diverses modifications ont été proposées, sans démon, comme celle de la figure 2.

Ce qu'a réalisé Landauer, c'est que dans toutes ces machines, à un moment ou à un autre, la machine va acquérir l'information disant dans quelle moitié de la boîte se trouvait la particule au début du cycle. Et pour que le cycle soit un cycle parfait, il faut pouvoir effacer cette information. Or cet effacement a un coût. Le plus simple est d'utiliser justement pour stocker un bit d'information une boîte de Szilard (cf Fig2) : si la particule est dans la moitié de gauche, le bit vaut 0, si elle est dans la moitié de droite, le bit vaut 1. L'effacement consiste à remettre tous les bits à 0. Pour ce faire, il suffit de faire l'opération montrée en Fig.3, gauche. On enlève la proi du milieu et on pousse un piston à partir de la droite jusqu'au milieu. Mais pousser un tel piston nécessite un travail, puisque la molécule exerce une pression. Le calcul de ce travail est exactement le même que celui que nous venons de faire : le travail nécessaire pour effacer un bit d'information dans la boîte de Szilard vaut $kT\ln 2$. On peut analyser de même d'autres façons de stocker un bit d'information, comme dans un double puits de potentiel avec une barrière grande devant kT (voir Fig.3, droite) : dans tous les cas, on peut montrer que le coût d'effacement est toujours $\geq kT\ln 2$, soit T fois l'entropie $k\ln 2$ associée aux deux états d'un bit.



Decodage:

11010010110010= $M_1\, M_3\, M_4\, M_2\, M_1\, M_4\, M_2$

FIGURE 4 – Code A. Un exemple de codage simple, pour le cas $N = 4$. Le premier mot est codé par 11, le deuxième par 10, le troisième par 01, le quatrième par 00. On décode facilement le message reçu.

2.4 Entropie et compression de données

Les deux problèmes majeurs de la théorie de l'information sont la compression des données et leur transmission fidèle. L'un et l'autre sont intimement reliés à la physique statistique. Nous étudierons quelques aspects de transmission et codage dans le prochain paragraphe, mais nous voudrions ici tout d'abord résumer quelques faits marquants concernant la compression de données, où le rôle de l'entropie comme mesure de la quantité d'information apparaît de façon particulièrement claire.

Considérons toujours un ensemble de N mots possibles, où le mot numéro n apparaît avec une probabilité p_n. La question du codage 'de source' est la suivante. On souhaite représenter chacun des mots par un 'mot de code', une succession de bits, 0 ou 1, qui représente ce mot. Par exemple un mot n donné pourrait être représenté par $C_n = 011011001$. Notre problème est de trouver un codage tel que le nombre de bits moyens nécessaires po qui sélectionnerait desur coder ces mots soit le plus petit possible. Cela pourra être utile par exemple si on souhaite ensuite envoyer ces mots à quelqu'un d'autre en utilisant le moins d'énergie possible. Pour ce faire nous devons bien entendu essayer de coder des mots fréquents par des codes courts. Nous nous restreindrons ici par simplicité au cas des codes instantanés qui peuvent être écrits sans virgule pour séparer les mots. Ces codes sont obtenus en choisissant les mots de code comme les feuilles d'un arbre binaire, comme expliqué en Figs.4,5. Le lecteur se convaincra facilement que, dans ces cas-là, on sait reconstituer la série des mots associés à une série de mots de codes, sans ambiguïté.

Des deux codes introduits en Figs. 4,5, lequel est le meilleur ? Tout dépend des probabilités p_n. Regardons ce problème de plus près sur deux exemples.

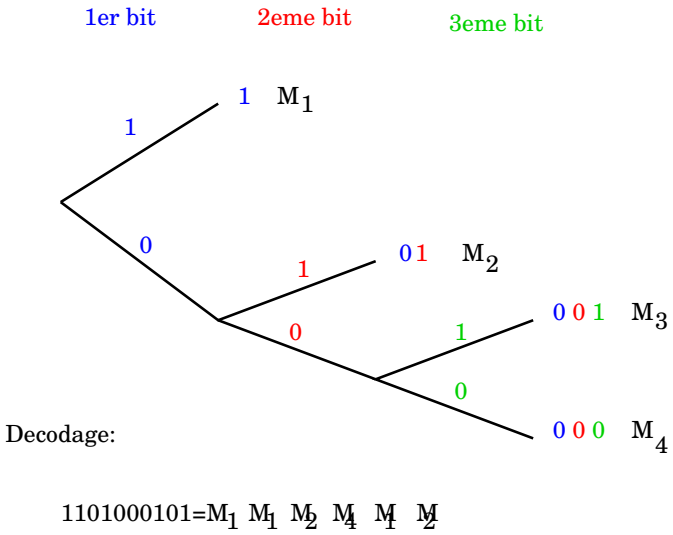


FIGURE 5 – Code B. Un deuxième exemple de codage simple, pour le cas $N = 4$. Le premier mot est codé par 1, le deuxième par 01, le troisième par 001, le quatrième par 000. On décode facilement le message reçu.

Le premier exemple est le cas d'une loi uniforme sur 4 événements, avec donc $p_n = 1/4$. Il est clair que, avec le code A, le nombre moyen de bits utilisés pour coder chaque mot est 2. Avec le code B, ce même nombre vaut $(1/4)(1+2+3+3) = 9/4$. Donc dans le cas des p_n uniformes le code A est préférable.

Notre deuxième exemple est celui d'une loi où $p_1 = 1/2$, $p_2 = 1/4$, $p_3 = p_4 = 1/8$. Avec le code A, le nombre moyen de bits utilisés pour coder chaque mot est toujours 2. Avec le code B, ce même nombre vaut $1/2 + 2/4 + 3/8 + 3/8 = 7/4$. Donc dans ce deuxième exemple c'est le code B qui est préférable, car il a su exploiter le biais de la distribution p_n en codant par un mot court, de 1 bit seulement, l'événement le plus probable.

Il est très instructif de mesurer l'entropie de chacune des deux lois que nous venons de prendre comme exemple. Pour la première on trouve $H = 2$, pour la deuxième $H = 7/4$. Remarquons que c'est aussi le nombre moyen de bits utilisés pour coder chaque mot en utilisant pour chacun le code optimal.

Ce résultat obtenu ici sur un cas très simple avec $N = 4$ est une coïncidence grâce à un exemple bien choisi, mais son contenu fondamental est correct. Plus petite est l'entropie de la loi p_n (et donc plus elle est biaisée), plus petit sera le nombre de bits nécessaire pour la coder (en utilisant un code bien choisi). Ceci a été démontré par un théorème fondamental dû à Shannon : le nombre minimal de bits nécessaire par mot, obtenu donc avec le meilleur code, est égal à l'entropie H de la loi statistique p_n des mots. Plus précisément, le théorème de Shannon est un résultat asymptotique qui se formule ainsi. Imaginons que nous fassions des paquets de k mots successifs, tirés indépendamment avec la loi p_n. L'entropie d'un paquet est $H(\text{paquet}) = kH$. Le théorème démontre que le nombre moyen minimal de bits nécessaire au codage des paquets, $< L >_{\text{minimal}}$, est dans l'intervalle $[kH, kH + 1]$. Donc, dans la limite où k est grand, on trouve bien une longueur moyenne des mots de

codes égale à l'entropie :

$$\frac{< L >_{\text{minimal}}}{k} \rightarrow H \tag{8}$$

Esquissons la preuve de ce théorème qui n'est pas difficile, et nous indique comment choisir les mots de codes de façon optimale.

Soit L_n la longueur du mot de code C_n associé à n. Le fait que les C_n soient choisis comme les feuilles d'un arbre (Figs.4,5) impose que les L_n ne peuvent pas être tous trop petits. Quantitativement, on peut démontrer l'inégalité de Kraft : $\sum_n 2^{-L_n} \leq 1$. (Preuve : soit L_{max} la longueur du mot de code le plus long : un mot de code de longueur L_n possède $2^{L_{max}-L_n}$ descendants dans l'arbre au niveau L_{max}. Les descendants au niveau L_{max} des divers mots de code forment des ensembles disjoints, dont le nombre est $\leq 2^{L_{max}}$).

Pour trouver le code optimal associé à une loi p_n, on doit minimiser $< L >= \sum_n p_n L_n$ avec la contrainte $\sum_n 2^{-L_n} \leq 1$. La méthode des multiplicateurs de Lagrange nous amène à minimiser $\Phi = \sum_n p_n L_n + \lambda \sum_n 2^{-L_n}$. Le résultat est : $2^{-L_n} = \frac{p_n}{\lambda \log 2}$. La contrainte est saturée, i.e. $\sum_n 2^{-L_n} = 1$, pour $\lambda = 1/\log(2)$, d'où :

$$L_n = -\log_2 p_n \tag{9}$$

Le résultat de Shannon s'obtient en prenant pour L_n le plus petit entier supérieur ou égal à $-\log_2 p_n$.

3 Codes de correction d'erreur

3.1 Cadre général

Nous abordons ici la question de la transmission d'information par un canal bruité. Qu'il s'agisse d'écrire sur son disque d'ordinateur, de communiquer avec un satellite, ou tout simplement de téléphoner sur un portable, le canal physique de transmission possède toujours un certain niveau de bruit, qui risque de perturber le message. Pour éviter les erreurs de transmission, on utilise des codes correcteurs d'erreur. Le mot à envoyer, écrit sous forme d'une succession de bits, des 0 et des 1, est codé par l'ajout d'un certain nombre de bits redondants. C'est ce mot codé qui est envoyé. A l'arrivée, le processus de décodage utilise l'information redondante contenue dans les bits supplémentaires pour détecter et corriger les erreurs induites par la transmission (voir Fig.3.1).

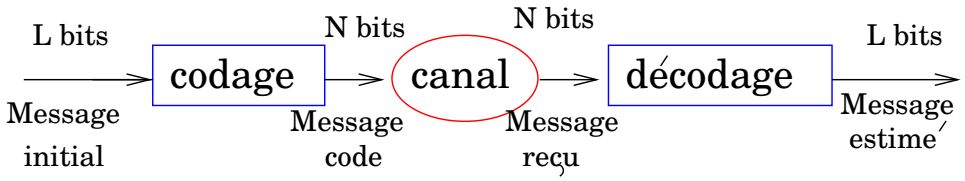


FIGURE 6 – Schéma de principe du codage pour la transmission d'information. Le bruit du canal de transmission altère le message, mais un dispositif de codage/décodage permet de corriger ces erreurs de transmission lorsque le bruit n'est pas trop fort.

Regardons un exemple. Le codage par répétition est l'un des plus simples. Il consiste à recopier chaque bit trois fois à l'identique : pour transmettre 1, on envoie 111. A la réception, on regarde chaque triplet de bits reçus, et on le décode par la règle de majorité : la réception de 101, au lieu de 111, est interprétée comme l'envoi du bit 1. Si le bruit lors de la transmission n'est pas trop élevé, on corrige ainsi des erreurs. Par exemple dans le cas où le canal retourne les bits avec une probabilité p (ce paramètre p est donc une mesure du niveau de bruit dans le canal), la probabilité d'erreur par bit transmis vaut $3p^2(1-p)+p^3$ (correspondant à tous les cas où le canal a retourné deux ou trois bits du triplet). Dans ce code on a un taux de transmission de 1/3 : on envoie dans le canal trois fois plus de bits que ceux du message que l'on souhaite transmettre. En répétant le bit envoyé $2k+1$ fois au lieu de 3, on arrive à une probabilité d'erreur qui se comporte à petit p proportionnellement à p^{k+1}, mais avec un taux de transmission égal à $1/(2k+1)$. Plus k est grand, plus le code est redondant, mieux on corrige les erreurs.

En général, chaque mot de L bits sera codé en un 'mot de code' de $N > L$ bits, où les bits supplémentaires sont utilisés pour la redondance. On définit le taux de transmission comme $r = L/N$: plus il est petit, plus le code est redondant. Le code est défini par l'ensemble de tous les 2^L mots de codes, ce que l'on appelle le dictionnaire, qui est supposé connu de l'envoyeur et du récepteur. Ayant envoyé un mot de code, on récupére à la sortie un mot de N bits. Pour un canal simple où le bruit retourne les bits avec probabilité p, le décodage consistera simplement à rechercher le mot de code le plus proche de ce mot reçu. Pour obtenir un code performant, on va donc chercher à ce que les mots du dictionnaire soient les plus éloignés les uns des autres, de sorte que, si le bruit de transmission est suffisamment faible, le décodage nous ramène sur le bon mot de code qui avait été envoyé. (voir Fig. 7). Nous allons maintenant décrire brièvement une famille de codes très performants découverte récemment.

3.2 Codes LDPC : construction

Parmi tous les mots de N bits, $(x_1, \cdots, x_N)$, où x_i prend les valeurs 0 ou 1, on va construire un dictionnaire constitué par tous les mots qui satisfont un certain nombre de relations de parité. Un code LDPC (pour 'Low Parity Density Check') est donc donné par l'ensemble des équations de parité qui doivent être satisfaites. Il est commode de le représenter par le "graphe de Tanner" de la Fig.8.

Pour obtenir de bons codes LDPC il faut utiliser deux ingrédients, la limite des longs mots et le désordre. La "limite thermodynamique" est obtenue en prenant $N >> 1$, mais en imposant également un grand nombre, $M = (1-r)N$, de contraintes de parité, de sorte que le dictionnaire comporte 2^{rN} mots de code. Par exemple une famille de codes réguliers consiste à utiliser un nombre fixé de variables, K, dans chaque équation de parité, et à imposer que chaque variable apparaisse dans L équations. Le code de la Fig.8 est de ce type, avec $K = 4$, $L = 2$. Le taux de transmission est alors défini comme $r = 1 - L/K$. On engendre un code (ou un dictionnaire) en choisissant le système d'équations au hasard parmi ceux qui satisfont ces contraintes. Ces codes sont donc construits en engendrant au hasard des équations de parité, de telle sorte que chaque équation relie K variables, et chaque variable apparaît dans L équations. On voit donc que, pour N grand, chacune des équations ne relie qu'une très faible fraction ($K/N \ll 1$) des variables, d'où le nom

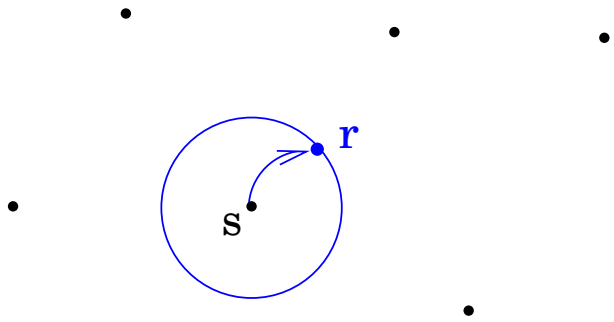


FIGURE 7 – Dans un canal dit "binaire symétrique", le bruit peut induire le retournement de chaque bit avec une probabilité p. Le mot de code transmis, $s_1, \ldots, s_N$ est donc altéré, et pour N grand le mot reçu, $r_1, \ldots, r_N$ est sur une sphère centrée en $s_1, \ldots, s_N$ et de rayon Np. La meilleure stratégie de décodage consiste en principe en la recherche du mot de code le plus proche de $r_1, \ldots, r_N$. Si le bruit p est assez petit, et si les mots de codes (représentés ici par des points noirs) sont éloignés, ce mot de code sera bien égal à $s_1, \ldots, s_N$, et le décodage sera parfait. En pratique cette recherche du mot de code le plus proche est beaucoup trop lente et il faut utiliser des algorithmes plus rapides, comme le passage de messages.

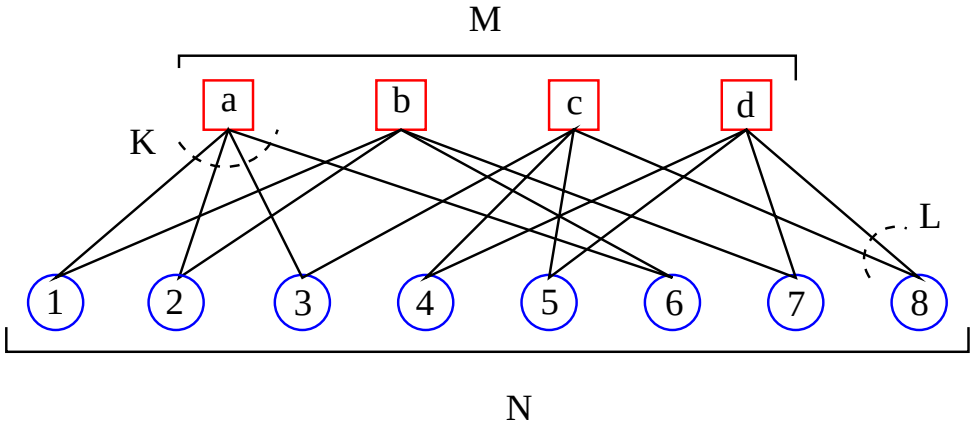


FIGURE 8 – Graphe de Tanner d'un petit code LDPC. Chaque bit est représenté par un cercle. Chaque carré représente une contrainte de parité, qui impose que la somme des bits qui lui sont connectés est paire. Le dictionnaire de ce code est constitué des sommets de l'hypercube (les variables x_i sont à valeur dans $\{0, 1\}$) qui satisfont les 4 équations (modulo 2) : $x_1 + x_2 + x_3 + x_6 = 0$, $x_1 + x_2 + x_6 + x_7 = 0$, $x_3 + x_4 + x_5 + x_8 = 0$, $x_4 + x_5 + x_7 + x_8 = 0$. Dans cet exemple, les $N = 8$ bits vérifient $M = 4$ équations de parité. Il n'y a donc que 4 bits indépendants, et le taux de compression vaut 1/2. Le dictionnaire comporte $2^4 = 16$ mots de code, parmi les $2^8 = 256$ mots possibles.

de 'Low Parity Density Check'.

Ayant vu comment on construit des codes LDPC, reste à trouver comment décoder un message reçu, c'est-à-dire comment trouver le mot de code le plus proche de ce message. En exploitant habilement la structure du code et le graphe de Tanner, on arrive à fabriquer des algorithmes de décodage qui corrigent toutes les erreurs lorsque le bruit est inférieur à un bruit critique p_d et fonctionnent très rapidement, avec un temps de calcul qui croît à peu près linéairement en N. Il se trouve que ces algorithmes utilisent en fait, à la base, une méthode de physique statistique bien connue, la méthode de Bethe et Peierls, qui est une version un peu améliorée de la théorie de champ moyen. C'est donc la théorie de champ moyen qui a permis la révolution de ces dix dernières années dans les codes de correction d'erreur ! Pour comprendre comment nos approches des la théorie des transitions de phase peuvent être utiles dans la théorie du cidage, nous allons tout d'abord formuler un peu plus précisément le problème du décodage, et voir son lien avec des spins d'Ising.

3.3 Décodage et spins en interaction

3.3.1 Un exemple de canal : le canal binaire symétrique

Commençons par préciser l'effet de la transmission. Le canal de transmission est supposé agir ainsi : lorsque la lettre $x_i = x \in \{0,1\}$ ($i \in \{1,\dots,N\}$) se présente à l'entrée, l'utilisateur va recevoir, à la sortie du canal, une lettre $y_i = y$. Le canal est décrit par la probabilité que la sortie soit y lorsque l'entrée est x, dénotée par $Q(y|x)$. Par exemple pour le canal binaire symétrique (CBS) la lettre reçue est $y \in \{0,1\}$, et $Q(y|x) = (1-p)\delta_{y,x} + p\delta_{y,1-x}$. S'il n'y avait pas de codage, l'utilisateur ne pourrait qu'exploiter le message reçu, $\underline{y} = (y_1,\dots,y_N)$, pour essayer de deviner celui envoyé. Supposant que la source de messages envoie les mots $\underline{x} = (x_1,\dots,x_N)$ avec une probabilité uniforme, la formule de Bayes sur les probabilités conditionnelles nous indique que, connaissant le message reçu $\underline{y}$, la probabilité $P(\underline{x}|\underline{y})$ pour que ce soit $\underline{x}$ qui ait été envoyé vaut

$$P(\underline{x}|\underline{y}) \cong \prod_{i=I}^{N} Q(y_i|x_i) \tag{10}$$

La stratégie de décodage consisterait alors à deviner la valeur de chacun des bits x_i envoyés comme étant celle qui maximise $Q(y_i|x_i)$. Bien entendu, dès que le canal est bruité, ce décodage commet des erreurs. C'est tout l'intérêt du codage, qui va permettre de les corriger.

3.3.2 Code LDPC et spins d'Ising

Dans un code LDPC aléatoire, le dictionnaire (c'est-à-dire l'ensemble des mots de codes) est défini par M équations de parité :

$$\forall a \in \{1,\dots,M\} : \quad x_{i_1(a)} \oplus \cdots \oplus x_{i_K(a)} = 0 \tag{11}$$

où la notation $\oplus$ indique la somme modulo 2. Chaque équation implique K indices choisis au hasard : chaque K-uplet $i_1(a),\dots,i_K(a)$ est choisi au hasard avec la loi uniforme sur tous les $\binom{N}{K}$ K-uplets possibles.

On considère un message envoyé $\underline{x} = (x_1, \ldots, x_N)$, qui par définition est un mot de code, satisfaisant donc toutes les M équations de parité. Si on reçoit à la sortie du canal le mot $\underline{y}$, la loi de Bayes nous donne comme précédemment la probabilité $P(\underline{x}|\underline{y})$ pour que ce soit $\underline{x}$ qui ait été envoyé, sous la forme :

$$P(\underline{x}|\underline{y}) \cong \prod_{i=I}^{N} Q(y_i|x_i) \prod_{a=1}^{M} \mathbb{I}\left(x_{i_1(a)} \oplus \cdots \oplus x_{i_K(a)} = 0 \right) \tag{12}$$

La modification par rapport à (19) est évidente : on cherche toujours à trouver le $\underline{x}$ le plus probable, mais cette recherche est maintenant limitée à tous les mots du dictionnaire.

Du point de vue de la physique statistique, on peut voir (12) comme une mesure de Boltzmann. Pour préciser cette assertion, il est commode d'utiliser des notations plus proches de celles des physiciens. On va donc introduire, en place des variables x_i, des spins d'Ising s_i définis par

$$s_i = 1 - 2x_i \tag{13}$$

Le facteur $Q(y_i|x_i)$ est alors une fonction de s_i qui peut se mettre sous la forme

$$Q(y_i|x_i) = e^{B_i s_i}/(2\cosh(B_i)) \ , \tag{14}$$

où :

$$B_i = \begin{cases} +(1/2)\ \log[(1-p)/p] \text{ si } y_i = 0 \\ -(1/2)\ \log[(1-p)/p] \text{ si } y_i = 1 \end{cases} \tag{15}$$

L'équation linéaire $x_{i_1(a)} \oplus \cdots \oplus x_{i_K(a)} = 0$ s'écrit quant à elle :

$$s_{i_1(a)} \times \cdots \times s_{i_K(a)} = 1 \tag{16}$$

Avec ces notations, la probabilité $P(\underline{x}|\underline{y})$ définie en (12) peut se réécrire sous la forme :

$$P(\underline{s}|\underline{B}) \ \cong \ \prod_i \exp\left(B_i s_i\right) \prod_{a=1}^{M} \mathbb{I}\left[s_{i_1(a)} \times \cdots \times s_{i_K(a)} = 1\right] \tag{17}$$

$$\cong \ \lim_{\gamma \to \infty} \exp\left(\sum_{i=I}^{N} B_i s_i \ + \ \gamma \sum_{a=1}^{M} [1 - s_{i_1(a)} \times \cdots \times s_{i_K(a)}] \right) \tag{18}$$

Sous cette forme, le problème du décodage, qui consiste à trouver la configuration de spins $\underline{s} = (s_1, \ldots, s_N)$ la plus probable, selon la loi de probabilité (18), n'est autre que la recherche de l'état fondamental d'un système de spins d'Ising, où chaque spin est plongé dans un champ magnétique extérieur B_i, et les spins interagissent par des interactions à K corps. Le champ extérieur traduit l'information que nous avons sur chaque spin grâce au signal reçu à la sortie du canal, les interactions entre groupes de K spins traduisent l'effet du codage.

L'étude de ce système de spins en interaction par des méthodes de champ moyen est à votre portée. Pour le lecteur curieux, elle est incluse dans l'appendice I. Le résultat est un ensemble d'équations de champ moyen reliant les valeurs des aimantations (ou en fait des champs magnétiques locaux) sur les différents sites.

L'analyse générale de ces équations dépasse le cadre de ce cours. Cette analyse démontre que le décodage par BP fonctionne parfaitement, au sens où il permet (à la limite $N \to \infty$) de corriger toutes les erreurs, lorsque le bruit du canal est inférieur à un certain seuil. La table suivante donne les résultats pour des codes $LDPC$ dans lesquels chaque variable apparaît dans l équations de parité, et chaque équation contient k variables. $R_{\rm des} = 1 - l/k$ est le taux de transmission du code (plus il est petit, plus le code est redondant, plus il sera capable de corriger des erreurs). Le nombre $p_{\rm d}$ est le seuil de décodage de l'algorithme BP pour ces codes LDPC agissant sur un canal binaire symétrique. Pour un bruit inférieur au seuil, $p < p_{\rm d}$, le décodage se fait sans erreur. Par exemple, avec un code $l = 3$, $k = 6$, on sait décoder sans erreur des niveaux de bruit de $p = 0.08$: on a doublé le nombre de bits envoyés, mais grâce à cette redondance on sait corriger toutes les erreurs de transmission si le canal a une probabilité de retourner un bit de moins de 8 pour cent.

l	k	$R_{\rm des}$	$p_{\rm d}$
3	4	1/4	0.1669(2)
3	5	2/5	0.1138(2)
3	6	1/2	0.0840(2)
4	6	1/3	0.1169(2)

4 Appendice I : L'approximation de champ moyen de Bethe et Peierls ('hors programme')

Les méthodes de décodage modernes s'expriment assez simplement dans le langage de la physique statistique : nous allons utiliser une forme d'approximation de champ moyen, dite approximation de Bethe-Peierls, introduite en physique statistique par H. Bethe en 1935, et développée dans le cadre des verres de spin dans les années 1980, en particulier à la suite des travaux de D. Thouless, P.W. Anderson et R. Palmer. Cette méthode très générale a été développée indépendamment dans le cadre des codes par Gallager dans les années 60. Passée alors inaperçue, à cause de la faiblesse des moyens de calculs, elle est tombée dans l'oubli et n'a été redécouverte que dans les dix dernières années. Elle a aussi été découverte indépendamment en intelligence artificielle en particulier par Pearl, qui lui a donné le nom de "Belief Propagation " (traduit en français par "Propagation des convictions" ou parfois "Propagation des croyances"), ou simplement "BP". Nous adopterons par la suite ce nom d'approximation BP, qui vaut aussi bien pour "Belief Propagation" que pour "Bethe-Peierls".

4.1 Modèle d'Ising

Avant d'appliquer cette méthode BP au problème du décodage, nous allons faire une petite digression pour expliquer comment elle fonctionne sur un problème bien connu de physique statistique, le modèle d'Ising. On considère donc un ensemble de spins d'Ising $\underline{s} = (s_1, \ldots, s_N)$ qui sont aux noeuds d'un réseau cubique, avec des conditions aux limites périodiques, et qui interagissent par un couplage d'échange entre plus proches voisins. L'énergie d'une configuration est

$$E(\underline{s}) = -J \sum_{<ij>} s_i s_j \ , \tag{19}$$

où la somme porte comme toujours sur tous les liens $< i, j >$ du réseau, et la probabilité de Boltzmann, à l'équilibre à la température $T = 1/k\beta$, est

$$P(\underline{s}) \cong \exp\left(\beta J \sum_{<ij>} s_i s_j\right) \tag{20}$$

où la notation $\cong$ signifie que la loi de probabilité doit être normalisée : il y a dans le membre de droite une constante de normalisation (l'inverse de la fonction de partition) que nous n'écrivons pas, qui garantit que $\sum_{\underline{s}} P(\underline{s}) = 1$. Cette notation sera utile pour simplifier les formules dans la suite de ce chapitre.

L'approximation de champ moyen habituelle suppose que chaque spin peut être vu comme ressentant un champ magnétique effectif dû à l'effet moyen de ses voisins, et conduit, comme nous l'avons vu, à prédire une transition de phase ferromagnétique à la température définie par $\beta_{MF} J = 1/p$, où p est le nombre de voisins d'un spin (égal à $2d$ pour un réseau cubique en dimension d). L'approximation BP est un peu plus raffinée. Considérons un spin s_0, et ses p voisins $s_1, \ldots, s_p$. Si nous connaissons la probabilité jointe des spins $s_1, \ldots, s_p$ *en l'absence de* s_0 (voir Fig.9), appelons-la $P^{(0)}(s_1, \ldots, s_p)$, nous pouvons en déduire la probabilité de s_0 par la relation

$$P(s_0) = \sum_{s_1,\ldots,s_p} P^{(0)}(s_1, \ldots, s_p) \exp\left(\beta J s_0 \sum_{r=1}^{p} s_r\right) \ . \tag{21}$$

L'idée de l'approximation BP est de négliger les corrélations entre $s_1, \ldots, s_p$ en l'absence de s_0 (on voit bien qu'il s'agit d'une approximation de type champ moyen). On suppose donc :

$$P^{(0)}(s_1, \ldots, s_p) = \mu_1^{(0)}(s_1) \ldots \mu_p^{(0)}(s_p) \tag{22}$$

où $\mu_i^{(0)}(s_i)$ dénote la probabilité de s_i en l'absence de s_0 (à cause de l'invariance par translation, cette probabilité est indépendante de i). L'idée fondamentale est donc la suivante : on néglige les corrélations entre $s_1, \ldots, s_p$ lorsque s_0 est absent, mais on prend en compte les corrélations entre $s_1, \ldots, s_p$ lorsque s_0 est présent. La probabilité jointe de $s_0, s_1, \ldots, s_p$ est approximée par

$$P(s_0, s_1, \ldots, s_p) = \mu_1^{(0)}(s_1) \ldots \mu_p^{(0)}(s_p) \ \exp\left(\beta J s_0 \sum_{r=1}^{p} s_r\right) \ . \tag{23}$$

Toujours dans l'esprit du champ moyen, il nous faut maintenant calculer les lois de probabilités $\mu_i^{(0)}$ de manière auto-cohérente. Dans le cadre du modèle d'Ising, l'invariance par translation nous dit que les lois de probabilités sur les différents sites sont toutes égales : on écrit $\mu_i^{(0)}(s_i) = \mu(s_i)$. La probabilité $\mu(s_i)$ est la probabilité d'un spin s_i lorsque l'un de ses voisins est absent. Pour la calculer de manière autocohérente, on va regarder notre spin central s_0 lorsqu'un de ses voisins est absent. Dans ce cas, s_0 n'a que

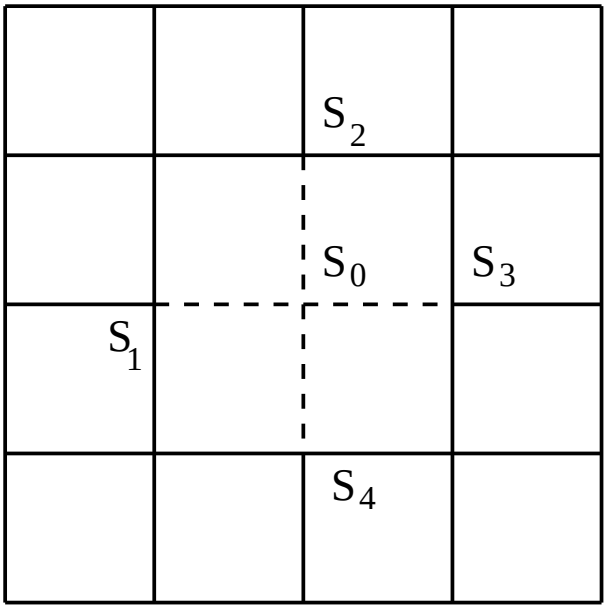


FIGURE 9 – Une petite partie d'un réseau de spins d'Ising en $d = 2$. Les spins interagissent entre plus proches voisins. Pour connaitre la probabilité de s_0, il suffit de connaitre la probabilité jointe des spins $s_1, \ldots, s_p$ *en l'absence de* s_0 et d'utiliser (21).

$p-1$ voisins, par exemple $s_1, \ldots s_{p-1}$, et la loi jointe de s_0 et de ses voisins est la même que celle trouvée en (23), mais avec $p-1$ voisins (voir Fig.10). On en déduit :

$$\mu(s_0) = \sum_{s_1,\ldots,s_{p-1}} \mu(s_1)\ldots\mu(s_{p-1}) \exp\left(\beta J s_0 \sum_{r=1}^{p-1} s_r\right) \ . \tag{24}$$

Ceci est l'équation d'autocohérence fondamentale qui permet de trouver $\mu(s)$.

Une méthode alternative conduisant au même résultat est la suivante. On remarque que, dans le cadre de l'approximation de factorisation de l'équation (23), on est en fait en train d'approcher notre problème par celui d'un graphe dans lequel les voisins du site s_0 n'interagissent pas, graphe appelé parfois réseau de Bethe, représenté en Fig.11. Si on enlève un voisin de s_0, on a un graphe dont la racine est s_0. Pour calculer la probabilité de s_0 dans ce graphe, on doit utiliser $P(s_0, s_1, \ldots, s_p) = \mu_1^{(0)}(s_1)\ldots\mu_{p-1}^{(0)}(s_{p-1}) \ \exp\left(\beta J s_0 \sum_{r=1}^{p-1} s_r\right)$. μ se calcule alors par récurrence et conduit bien à l'équation d'autocohérence (24).

Etudions maintenant l'équation d'autocohérence (24). Elle peut s'exprimer simplement ainsi : comme s est une variable d'Ising, prenant seulement les valeurs ± 1, toute distribution $\mu(s)$ peut s'écrire sous la forme $\mu(s) \cong e^{\beta h s}$, où h est un champ magnétique local effectif, caractérisant μ. Dans le membre de droite de (24), on doit alors calculer des sommes du type

$$\sum_{s_1} \mu(s_1) \exp\left(\beta J s_0 s_1\right) \cong \sum_{s_1} \exp\left(\beta J s_0 s_1 + \beta h s_1\right) \cong \exp\left(\beta u(\beta, J, h) s_0\right) \ , \tag{25}$$

où

$$u(\beta, J, h) = \frac{1}{\beta} \mathrm{atanh}(\tanh(\beta J)\tanh(\beta h)) \ . \tag{26}$$

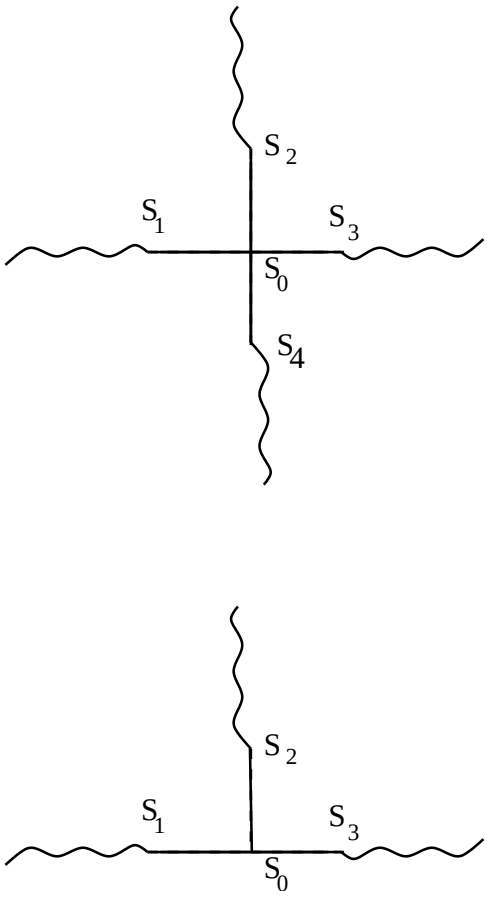


FIGURE 10 – Approximation BP. En haut, une représentation de l'équation (23) donnant la probabilité jointe des spins s_0, s_1, s_2, s_3, s_4. Les lignes ondulées indiquent la présence d'un facteur $\mu(s_i)$, les lignes droites indiquent le présence d'un facteur d'interaction $e^{\beta J s_i s_j}$.En bas, la loi de probabilité de s_0, s_1, s_2, s_3, donc lorsqu'on a enlevé un voisin de s_0. En la sommant sur s_1, s_2, s_3, on obtient la loi de probabilité de s_0 en l'absence d'un de ses voisins, qui est égale à $\mu(s_0)$. C'est l'équation d'autocohérence (24).

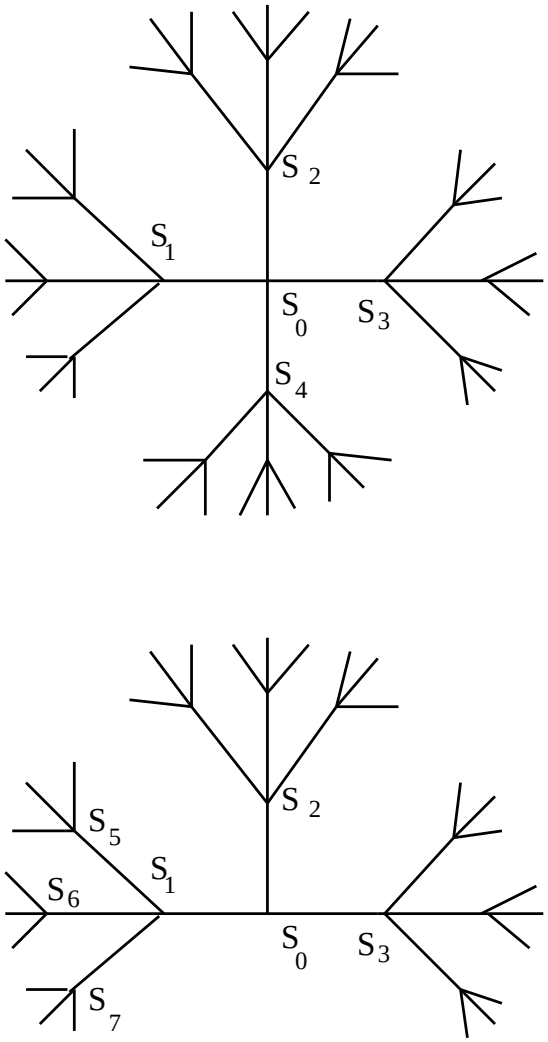


FIGURE 11 – Approximation BP : on peut voir cette approximation comme l'étude d'un système de spins sur un réseau localement en arbre (figure du haut). Le calcul de la distribution de s_0 en l'absence de s_4 revient à étudier le problème sur le graphe du bas, où s_0 est la racine de l'arbre (le seul site n'ayant que trois voisins). Si on enlève s_0, on obtient trois arbres identiques disconnectés, de racines s_1, s_2, s_3. Ceci permet d'établir l'équation de récurrence (24).

(Cette expression s'obtient facilement si on se souvient que, pour un spin d'Ising s et pour tout réel a, $e^{as} = \cosh(a)[1 + s\tanh(a)]$). L'équation d'autocohérence (25) devient simplement une équation pour h :

$$h = (p-1)u(\beta, J, h) = \frac{p-1}{\beta}\text{atanh}(\tanh(\beta J)\tanh(\beta h)) \ , \tag{27}$$

Dans ces mêmes notations, l'équation de champ moyen simple serait $h = \frac{p}{\beta}\text{atanh}(J\tanh(\beta h))$. On voit bien que BP modifie cette équation, mais le résultat est qualitativement du même type et la discussion de cette nouvelle équation est analogue à celle faite dans le champ moyen simple. La résolution graphique de (28) montre une transition de phase entre une phase paramagnétique à $\beta < \beta_{BP}$, caractérisée par $h = 0$, et une phase ferromagnétique avec $h > 0$ obtenue pour $\beta > \beta_{BP}$. La température inverse de transition, β_{BP}, est obtenue lorsque le membre de droite de (28), vu comme fonction de h, a une pente égale à 1 en $h = 0$. Donc :

$$\beta_{BP} = \frac{1}{J}\text{atanh}(1/(p-1)) \ , \tag{28}$$

La table ci-dessous montre les diverses prédictions pour l'inverse de la température critique obtenues par le champ moyen et par l'approximation BP, comparées au résultat numérique β_c, pour des modèles d'Ising sur des réseaux (hyper-)cubiques en dimension d (avec donc $p = 2d$). On voit que BP améliore sensiblement le champ moyen habituel, et devient très bon lorsque la dimension augmente. L'approximation BP est également exacte lorsque la dimension vaut 1 (le graphe d'interaction, une ligne, est bien 'localement en arbre', ce qui assure sa validité), et on voit bien qu'elle prédit une transition seulement à température nulle dans ce cas. C'est un bon exercice d'étudier par cette méthode le modèle d'Ising en une dimension en présence d'un champ magnétique extérieur.

d	$\beta_{MF}J$	$\beta_{BP}J$	$\beta_c J$
2	0,250	0,347	$1/2\ \log(1+\sqrt{2}) \simeq 0{,}441$
3	0,167	0,203	0,222
4	0,125	0,144	0,150

4.2 Approximation de Bethe-Peierls pour les codes

Voyons comment généraliser l'approximation BP du modèle d'Ising pour pouvoir l'appliquer aux codes. Il y a deux éléments nouveaux : d'une part les interactions entre spins sont de type multispins, et non plus simplement interactions de paires, et d'autre part le système n'est pas homogène (tous les spins ne sont pas équivalents). Regardons un problème d'Ising général contenant ces deux ingrédients, pour lequel la probabilité de la configuration $\underline{s} = (s_1, \ldots, s_N)$ s'écrit comme :

$$P(\underline{s}) \cong \exp\left(B_i s_i\right) \prod_{a=1}^{M} \psi_a(\underline{s}_{\partial a}) \tag{29}$$

où $\partial a = \{i_1(a), \ldots, i_K(a)\}$ représente l'ensemble des spins impliqués dans le terme d'interaction numéro a.

Regardons un spin, numéro i, impliqué dans un certain nombre d'interactions dont a, et considérons la probabilité de s_i en l'absence de a, que nous dénoterons par $\nu_{i\to a}(s_i)$. Nous introduisons aussi la probabilité de s_i en l'absence du champ B_i et de toutes les autres interactions auxquelles il participe, sauf a, que nous dénotons $\widehat{\nu}_{a\to i}(s_i)$. On a bien évidemment :

$$\nu_{i\to a}(s_i) \cong e^{B_i s_i} \prod_{b\in\partial i\backslash a} \widehat{\nu}_{b\to i}(s_i)\,, \tag{30}$$

où $\partial i\backslash a$ désigne l'ensemble des variables, autres que i, impliquées dans l'interaction a. Pour calculer $\widehat{\nu}^{(t)}_{a\to i}(x_i)$, on procède comme au paragraphe 4 en supposant que les autres variables liées à l'interaction a (et différentes de i) sont non corrélées lorsque a est absente. Il en résulte que :

$$\widehat{\nu}^{(t)}_{a\to i}(s_i) \cong \sum_{\{s_j\}_{æ\in\partial a\backslash i}} \mathbb{I}(s_{j_1}\times\cdots\times s_{j_{k-1}} = s_i) \prod_{j\in\partial a\backslash i} \nu^{(t)}_{j\to a}(s_j)\,. \tag{31}$$

En utilisant le fait que les distributions $\nu_{i\to a}(s_i)$ et $\widehat{\nu}_{b\to i}(s_i)$ sont des fonctions d'une variable d'Ising, il est commode de les représenter par des champs magnétiques locaux :

$$h_{i\to a} = \frac{1}{2}\log\frac{\nu_{i\to a}(1)}{\nu_{i\to a}(-1)}\,, \qquad u_{a\to i} = \frac{1}{2}\log\frac{\widehat{\nu}_{a\to i}(1)}{\widehat{\nu}_{a\to i}(-1)}\,. \tag{32}$$

En terme des champs locaux, les équations BP s'écrivent plus simplement :

$$h_{i\to a} = B_i + \sum_{b\in\partial i\backslash a} u_{b\to i}\,, \qquad u_{a\to i} = \mathrm{atanh}\Big\{\prod_{j\in\partial a\backslash i}\tanh h_{j\to a}\Big\}\,. \tag{33}$$

L'idée du décodage par BP est d'itérer ces équations, en partant de conditions initiales où $h = u = 0$, jusqu'à obtenir un point fixe qui sera une solution des équations. On va donc implémenter l'itération suivante :

$$u^{(t)}_{a\to i} = \mathrm{atanh}\Big\{\prod_{j\in\partial a\backslash i}\tanh h^{(t)}_{j\to a}\Big\} \qquad h^{(t+1)}_{i\to a} = B_i + \sum_{b\in\partial i\backslash a} u^{(t)}_{b\to i}\,,\,. \tag{34}$$

Lorsque le décodage se passe bien, on obtient un point fixe, solution des équations BP, et on peut décoder en regardant le champ total sur le site i, et en alignant s_i sur ce champ :

$$s_i = Signe(B_i + \sum_{b\in\partial i} u_{b\to i})\,. \tag{35}$$

5 Appendice II : La méthode des multiplicateurs de Lagrange

Supposons [1] que l'on veuille trouver les extrema d'une fonction F de N variables $x_1, x_2, ..., x_N$ et que ces N variables ne soient pas indépendantes mais liées par n contraintes

$$f_a(x_1, x_2, ..., x_N) = 0\ ; a = 1, ..., n\ ; \quad n < N.$$

1. Cet appendice est emprunté au polycopié d'E.Brézin.

(On suppose toutes les fonctions dérivables, les contraintes indépendantes, etc.). La méthode de Lagrange consiste à introduire une nouvelle fonction

$$G(x_1, x_2, ..., x_N\,; \lambda_1, \lambda_2, ..., \lambda_n) = F(x_1, x_2, ..., x_N) - \lambda_1 f_1(x_1, x_2, ..., x_N) - ... - \lambda_n f_n(x_1, x_2, ..., x_N)$$

où les n paramètres λ_a, appelés multiplicateurs, sont des constantes qui seront déterminées plus loin. On démontre aisément, en suivant Lagrange, que les extrema de F peuvent être déterminés maintenant de la manière suivante :

(i) On cherche les extrema de G en supposant les N variables x_i indépendantes. On résout donc les N équations $\partial G \;/\; \partial x_i = 0, i = 1, 2, ..., N$. Une solution x_i^* de ce système de N équations, est une fonction des n multiplicateurs λ_a.

(ii) Pour déterminer les valeurs de ces multiplicateurs on reporte la solution $x_i^*(\lambda_1, \lambda_2, ..., \lambda_n)$ dans les n contraintes $f_a(x_1^*, x_2^*, ..., x_N^*) = 0$. Ce sont n équations pour les n multiplicateurs.

(iii) Après résolution de ces n équations on reporte le résultat dans les x_i^* et l'on obtient un extremum de F.

[*La démonstration de la méthode des multiplicateurs de Lagrange est simple* : les n contraintes $f_p(x_1, ..., x_N) = 0$ avec $p = 1, ..., n$ définissent une surface Γ de co-dimension n (i.e. de dim. $N-n$) dans $\mathbb{R}^N$. L'hyperplan tangent à Γ en un point quelconque est défini par les n vecteurs $\overrightarrow{\nu_p}$ de composantes $(\partial f_p \;/\; \partial x_1, ..., \partial f_p \;/\; \partial x_N)$, qui sont normaux à cet hyperplan. Au voisinage d'un extremum la différentielle dF s'annule ; il doit en être ainsi pour tout vecteur $(dx_1, ..., dx_N)$ appartenant à l'hyperplan tangent à Γ, puisque les seules variations permises sont sur Γ. Par conséquent le vecteur $\overrightarrow{v} = (\partial F \;/\; \partial x_1, ..., \partial F \;/\; \partial x_N)$ doit être normal à cet hyperplan ; il est donc combinaison linéaire des n vecteurs $\overrightarrow{\nu_p}$. Il existe donc n nombres $\lambda_1, ..., \lambda_n$ tels que $\overrightarrow{v} = \lambda_1 \overrightarrow{\nu_1} + ... + \lambda_n \overrightarrow{\nu_n}$. Revenant à la définition de ces vecteurs on voit que cela implique que les dérivées $\frac{\partial}{\partial x_k}\{F - \lambda_1 f_1 - ... - \lambda_n f_n\}$, pour $k = 1, ..., N$, sont nulles au voisinage d'un extremum, ce qui constitue bien le théorème annoncé].

Exemple : après des jets répétés d'un dé, on effectue la moyenne des résultats. Il se trouve que cette moyenne, qui serait évidemment de 3,5 pour un dé honnête, est égale à 4. En l'absence de plus ample information, quelle loi de probabilité convient-il d'associer à ce dé ?

Nous déterminons les probabilités p_n, $n = 1, ..., 6$, qui maximisent l'information manquante $S_{stat} \propto -\sum_n p_n \ln p_n$, compte tenu des deux contraintes

$$\sum_{n=1}^{6} n p_n = 4 \quad \text{et bien entendu} \quad \sum_{n=1}^{6} p_n = 1$$

Pour déterminer la solution de ce problème on peut utiliser ces deux relations afin d'éliminer deux des six variables, tirer par exemple p_5 et p_6 de ces deux relations, reporter dans S et annuler les dérivées par rapport aux quatre variables indépendantes restantes. Le résultat est $p_1 = 0,103\,; p_2 = 0,123\,; p_3 = 0,146\,; p_4 = 0,174\,; p_5 = 0,207\,; p_6 = 0,247$.

La méthode de Lagrange fournit cependant une solution bien plus rapide, symétrique et élégante. Nous introduisons la fonction

$$\begin{aligned} G(p_1, ..., p_6) = & -p_1 \ln p_1 - p_2 \ln p_2 - ... - p_6 \ln p_6 \\ & - \lambda_1(p_1 + ... + p_6 - 1) - \lambda_2(p_1 + 2p_2 + ... + 6p_6 - 4) \quad . \end{aligned}$$

Les six équations $\partial G/\partial p_n = 0$ s'écrivent $-1 - \ln p_n - \lambda_1 - \lambda_2 n = 0$, dont la solution est $p_n^* = \frac{1}{Z} e^{-\lambda n}$, où $\ln Z = 1 + \lambda_1$ et $\lambda_2 = \lambda$.

La normalisation $\Sigma p_n = 1$ fixe la constante de normalisation : $Z = \frac{1-e^{-6\lambda}}{e^{\lambda}-1}$, puis la moyenne $\Sigma n p_n = 4$ s'obtient en remarquant que

$$Z = \sum_{n=1}^{6} e^{-\lambda n} \text{ et donc } -\frac{1}{Z}\frac{dZ}{d\lambda} = \sum_{n=1}^{6} n \frac{e^{-\lambda n}}{Z} = \sum_{n=1}^{6} n p_n^* = 4 \ ;$$

cela conduit à une équation pour λ dont la solution est $\exp(\lambda) = 0,839769...$, dont on tire les valeurs des 6 probabilités cherchées, données ci-dessus.