



HAL
open science

Décodeur amélioré pour le codage de source avec information adjacente compressée

Anne Savard, Claudio Weidmann

► **To cite this version:**

Anne Savard, Claudio Weidmann. Décodeur amélioré pour le codage de source avec information adjacente compressée. GRETSI 2013, Sep 2013, Brest, France. pp.1-4. hal-00863891

HAL Id: hal-00863891

<https://hal.science/hal-00863891>

Submitted on 19 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Décodeur amélioré pour le codage de source avec information adjacente compressée

Anne SAVARD, Claudio WEIDMANN

Equipes Traitement de l'Information et Systèmes (ETIS)
CNRS UMR8051-ENSEA-Université de Cergy-Pontoise
Cergy, France
(anne.savard,claudio.weidmann)@ensea.fr

Résumé — Dans ce papier, nous proposons un nouvel algorithme de décodage itératif pour le problème de codage de source avec information adjacente compressée. L'information adjacente est compressée par un quantificateur qui génère un index. Plutôt que d'utiliser un seul représentant pour l'information adjacente (la reconstruction correspondante à l'index), nous allons projeter une solution intermédiaire du décodage de la source sur la cellule de Voronoï correspondante à l'index reçu. Grâce à cette projection, nous espérons rapprocher l'information adjacente du mot source recherché, accélérant ainsi le décodage itératif de la source. Des simulations utilisant un code LDPC pour la branche principale et une quantification sur treillis pour l'information adjacente montrent que pour un nombre d'itérations de décodage fixé, notre algorithme permet d'augmenter le nombre de mots correctement décodés.

Abstract — In this paper, we present a new iterative decoding algorithm for the source coding with coded side information problem. Side information is compressed to an index. Instead of using only one representative for the side information (the reconstruction of the index), we will project an intermediate estimate of the source word onto the Voronoi cell associated to the index, and so accelerate the iterative source decoding. Simulations using LDPC codes in the main branch and trellis-coded quantization for the side information show that our method increases the number of correctly decoded source words for a fixed number of decoding iterations.

1 Introduction

Nous considérons le problème de codage de source avec information adjacente compressée. Ce modèle de la théorie de l'information peut s'appliquer à un réseau de capteurs comme suit. Soient des capteurs avec des mesures discrètes corrélées, par exemple des températures de lieux proches (modélisables comme variables aléatoires discrètes corrélées). Le but est de transmettre sans perte la mesure X d'un capteur principal, en utilisant la mesure Y d'un capteur secondaire comme *information adjacente*, qui sera utilisée uniquement à ce but. Les capteurs doivent encoder les mesures de manière indépendante et les liens des capteurs vers la destination sont soumis à une contrainte de débit. Le problème théorique adresse la question du compromis entre le débit principal R_X et le débit secondaire R_Y nécessaires pour reconstruire X sans perte (Y n'est pas reconstruit). Si $R_Y < H(Y)$, l'information adjacente devra être compressée avant d'être envoyée au décodeur final.

Plus formellement, on considère deux sources discrètes corrélées X et Y à alphabets finis \mathcal{X} et \mathcal{Y} ayant $P_{X,Y}$ comme distribution jointe, encodées séparément par les encodeurs E_X et E_Y , au rendement R_X et R_Y respec-

tivement. Le décodeur D_X essaie de reconstruire X sans pertes, tandis que Y ne sera pas reconstruite et ne sert que d'information adjacente (pour le décodage de X), voir FIGURE 1.

Alhswede et Körner [1] ont caractérisé la région de rendements (R_X, R_Y) atteignables pour ce problème. Une paire de rendements (R_X, R_Y) est atteignable si

$$\begin{cases} R_X \geq H(X|U) \\ R_Y \geq I(Y; U) \end{cases} \quad (1)$$

pour une variable aléatoire auxiliaire $U \in \mathcal{U}$ vérifiant $|\mathcal{U}| \leq |\mathcal{Y}| + 2$ et telle que $X - Y - U$ forment une chaîne de Markov.

Pour le cas où X et Y sont binaires, équiprobables avec une corrélation symétrique, Gu et al. [2] ont montré qu'il est optimal de compresser Y avec comme critère de distorsion la distance de Hamming.

On peut alors donner une expression analytique pour la région de rendements atteignables. Soit X une source binaire équiprobable. La corrélation entre X et Y sera modélisée par un canal binaire symétrique de probabilité d'erreur p (BSC- p). L'encodeur E_Y qui donne en sortie une version quantifiée de Y peut être représenté, de façon idéalisée, par un canal BSC- D , où D est la distorsion op-

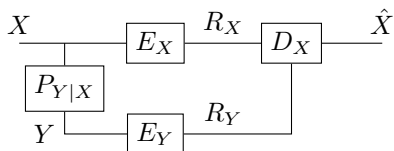


FIGURE 1 – Problème de l’information adjacente compressée : cas général

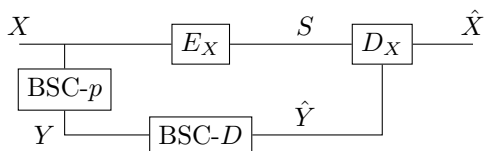


FIGURE 2 – Problème de l’information adjacente compressée : cas binaire

timale obtenue au rendement R_Y . La FIGURE 2 reprend ces notations.

Dans le cas binaire, le système (1) devient :

$$\begin{cases} R_X \geq h(p + D - 2pD) \\ R_Y \geq 1 - h(D) \end{cases} \quad (2)$$

où h est la fonction d’entropie binaire.

Une implémentation naïve pour le cas binaire est la suivante. L’encodeur E_Y est considéré idéal, ainsi la concaténation des 2 BSC précédents est équivalente à un BSC de probabilité d’erreur $\epsilon = (1 - p)D + (1 - D)p$. Cette situation correspond au problème de Slepian-Wolf [3] où le canal de corrélation est un BSC- ϵ et où X est encodé au rendement $H(X|Y)$. Une solution pratique basée sur des codes LDPC a été proposée par exemple par Liveris et al. [4].

La séquence binaire $X^n = (X_1, X_2, \dots, X_n)$ sera compressée en calculant son syndrome $S^{n-k} = X^n H^T$, $H \in GF(2)^{(n-k) \times n}$ étant la matrice de vérification de parité d’un code LDPC. L’encodeur E_Y enverra une version compressée W de Y^n . Pour ce faire, nous utiliserons un quantificateur binaire sur treillis basé sur un code convolutif décodé avec l’algorithme de Viterbi. Ce choix offre un excellent rapport complexité-performance.

L’algorithme de Viterbi est utilisé pour trouver le mot de code \hat{Y}^n le plus proche en terme de distance de Hamming de la séquence d’entrée Y^n , W correspondant à la séquence de bit d’information associée.

La reconstruction $\hat{Y}^n(W)$ correspondante à l’index W sera utilisée comme information canal au décodeur LDPC D_X . Comme pour [4], le décodeur doit estimer la séquence X^n à partir de son syndrome S^{n-k} et de l’information adjacente reconstruite \hat{Y}^n .

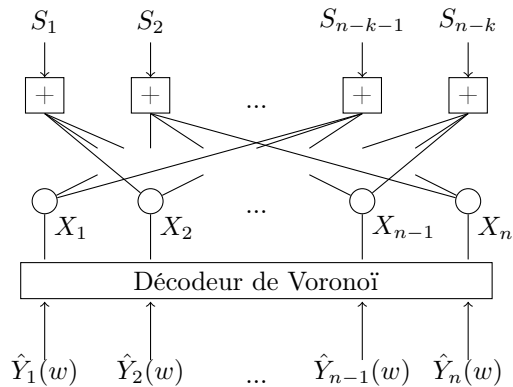


FIGURE 3 – Graphe du décodeur proposé

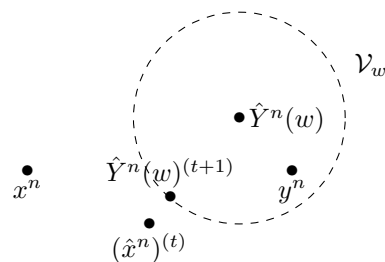


FIGURE 4 – Intuition géométrique

2 Méthode proposée

Notre méthode repose sur l’observation que plusieurs séquences d’information adjacente sont mappées sur un même index w par l’opération de compression. Cet ensemble de séquences correspond à la cellule de Voronoï $\mathcal{V}_w = \{y^n \in \{0, 1\}^n \mid E_Y(y^n) = w\}$. Comme le code convolutif est linéaire, $\mathcal{V}_w = \mathcal{V}_0 \oplus \hat{Y}^n(w)$, où \mathcal{V}_0 désigne la cellule de Voronoï associée à l’origine (index nul). (\oplus représente la somme modulo 2 bit à bit.) Il nous faut donc caractériser l’ensemble \mathcal{V}_0 afin de réaliser notre projection.

Comme pour la méthode classique, nous allons démarrer le décodage de X^n par T itérations de décodage de propagation de croyances du LDPC. En cas d’échec du décodeur après T itérations, nous allons modifier les probabilités a priori sur l’information adjacente par projection sur la cellule de Voronoï correspondant à w . La FIGURE 3 montre le graphe du décodeur où le «décodeur de Voronoï» est un bloc qui modifie les valeurs des log-rapport de vraisemblance (LLR) donnés au décodeur LDPC. L’intuition géométrique sur laquelle repose ce décodeur est illustrée sur la FIGURE 4.

2.1 Caractérisation de la cellule de Voronoï \mathcal{V}_0

Il a été montré dans [5] que la cellule de Voronoï \mathcal{V}_0 peut être caractérisée par l’évolution d’une machine à états finis (FSM), que nous dénoterons Voronoï FSM.

Pour construire la Voronoï FSM de \mathcal{V}_0 , nous allons étudier l'évolution de l'algorithme de Viterbi du quantificateur binaire en termes de différences de métriques des séquences mappées sur le mot de code nul. Nous utilisons le code convolutif (5,7) $(1 + D^2, 1 + D + D^2)$ de rendement $R_Y = \frac{1}{2}$ comme exemple.

L'encodeur pour ce code possède 4 états, ainsi, chaque section du treillis est caractérisée par un 4-uple de métriques, que nous appellerons état de métrique. Ces 4-uples sont calculés en utilisant la métrique de Hamming entre les étiquettes du treillis et la séquence d'entrée. Les états de métrique pourront être translatés par un 4-uple constant afin que leur minimum soit 0. La Voronoï FSM est obtenue en cherchant toutes les transitions entre les états de métrique telles que la métrique minimale soit celle correspondant au mot de code nul.

Un état de métrique $[a \ b \ c \ d]$ doit vérifier la condition suivante pour être inclus dans la Voronoï FSM :

$$\arg \min \{a + d_H(l_{E_1 \rightarrow E_1}, s), b + d_H(l_{E_2 \rightarrow E_1}, s)\} = E_1 \quad (3)$$

où $l_{E_i \rightarrow E_j}$ est le label de la transition de l'état du treillis E_i à l'état du treillis E_j et s la séquence en entrée correspondant à cette section du treillis. Nous supposons que l'encodeur démarre à l'état zéro.

Cette procédure peut être étendue pour tout code convolutif. La condition (3) devient :

$$\arg \min_{i: \exists E_i \rightarrow E_1} \{M_i + d_H(l_{E_i \rightarrow E_1}, s)\} = E_1, \quad (4)$$

où M_i est la i^{eme} composante de l'état de métrique et $\exists E_i \rightarrow E_1$ signifie qu'il existe une transition entre les états E_i et E_1 .

2.2 Décodeur

Le décodeur reçoit le syndrome s^{n-k} de l'encodeur E_X et l'index w de l'encodeur E_Y . La procédure de décodage commence avec le mot de code $\hat{Y}^n(w)$. Après T itérations, si le décodeur LDPC n'a pas convergé, nous calculons les probabilités a posteriori pour obtenir $(\hat{x}^n)^{(T)}$ et nous cherchons dans l'ensemble $\mathcal{V}_w = \mathcal{V}_0 \oplus \hat{Y}^n(w)$ la séquence la plus proche.

$$\hat{Y}^n(w)^{(T+1)} = \hat{Y}^n(w) \oplus \arg \min_{v \in \mathcal{V}_0} d_H(v, (\hat{x}^n)^{(T)} \oplus \hat{Y}^n(w)) \quad (5)$$

Cette séquence est ensuite utilisée pour modifier les LLRs avant de poursuivre avec des itérations du décodeur LDPC (les messages ne sont pas remis à zéro). Si après t itérations supplémentaires, le décodeur n'a toujours pas convergé, nous réitérons la procédure décrite précédemment.

Dans les sections suivantes, nous allons présenter différentes façons de réaliser la projection sur la cellule de Voronoï et les mises à jour des LLR associées.

3 Décodeur de Voronoï basé sur l'algorithme de Viterbi

Un décodeur de Voronoï à faible complexité peut être construit en utilisant l'algorithme de Viterbi à entrée souple sur la Voronoï FSM pour réaliser la projection. En se basant sur nos expériences, nous proposons la mise à jour des LLR suivante :

$$LLR_i = \begin{cases} 0.9^j * LLR(\hat{Y}_i(w)) & \text{si } \hat{Y}_i(w) \neq \hat{Y}_i(w)^{(T+1)} \\ LLR(\hat{Y}_i(w)) & \text{sinon} \end{cases} \quad (6)$$

où j est le nombre de fois que nous avons effectué la projection. Cette mise à jour, heuristique, est basée sur l'intuition suivante : plus on a fait de projections (et d'itérations de décodage) plus la séquence \hat{x}^n devrait être proche de la séquence source. Ainsi pour les premières projections, nous réduisons les LLR de façon plus douce car le décodeur LDPC aurait du mal à les corriger, alors que pour les dernières projections, cela ne posera plus de problème.

Nous donnons les résultats obtenus pour un code LDPC optimisé de rendement $R_X = 0.5$, de taille $N = 1200$ avec comme distribution des nœuds de variable

$$\lambda(x) = 0.24426x + 0.25907x^2 + 0.01054x^3 + 0.05510x^4 + 0.01455x^7 + 0.01275x^9 + 0.40373x^{11}$$

et des nœuds de parité de degrés concentrés, trouvé dans [6]; la matrice H est construite avec l'algorithme PEG. Le code convolutif du quantificateur, pris dans [7] est de rendement $R_Y = \frac{5}{6}$. Les résultats sont donnés pour 10000 échantillons. Nous effectuons $T = 30$ itérations de décodage avec les LLR associés à $\hat{Y}^n(w)$. En cas d'échec, nous pouvons rechercher jusqu'à $N_t = 18 \hat{Y}^n(w)^{(T+kt+1)}$ en effectuant pour chacun d'eux jusqu'à $t = 15$ itérations de décodage.

Nous pouvons constater sur la FIGURE 5 et la FIGURE 6 que la méthode que nous proposons permet d'améliorer les performances de convergence pour le problème du codage de source avec information adjacente compressée.

4 Décodeur de Voronoï basé sur l'algorithme BCJR

Le décodeur précédent, basé sur un algorithme de Viterbi à sortie dure a l'avantage de trouver une séquence de \mathcal{V}_w , cependant trouver une règle optimale de mise à jour des LLR est non trivial. Afin de pallier à ce problème, nous sommes en train de considérer des algorithmes à entrée souple et sortie souple, comme le BCJR.

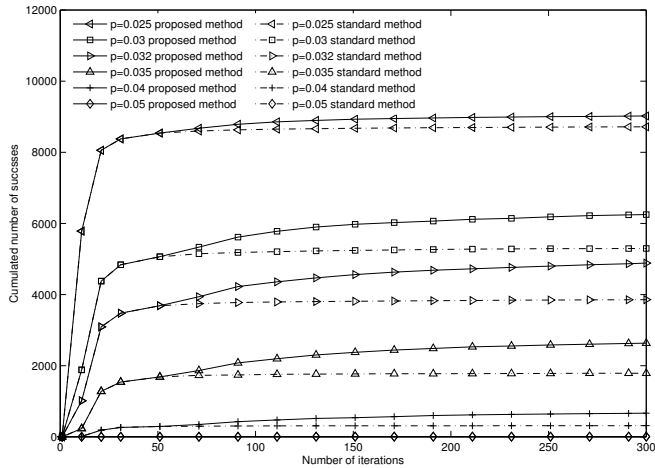


FIGURE 5 – Comparaison entre la méthode standard et la méthode proposée avec $R_X = 0.5$, $R_Y = 5/6$, $T = 30$, $t = 15$, $N_t = 18$ et un Viterbi à entrée souple pour effectuer la projection : Nombre cumulé de succès en fonction du nombre d’itérations de décodage.

5 Conclusion

Nous avons présenté un nouvel algorithme de décodage pour le problème du codage de source avec information adjacente compressée et avons comparé plusieurs implémentations avec une implémentation standard. L’idée principale est de tirer avantage de la connaissance de la cellule de Voronoï afin d’accélérer le décodeur principal. En projetant une solution intermédiaire sur cette cellule, nous améliorons l’information adjacente donnée au décodeur principal. Nous obtenons un gain en terme de nombre de succès vs. le nombre d’itérations de décodage en utilisant un décodeur Viterbi pour effectuer la projection.

Pour la suite de nos travaux, nous comptons développer un décodeur basé sur l’algorithme BCJR afin de l’étudier de façon plus théorique, en commençant par une étude en terme d’évolution de densité similaire à l’approche proposée dans [8].

Références

- [1] R. Ahlswede and J. Körner, “Source coding with side information and a converse for degraded broadcast channels,” *IEEE Trans. on Information Theory*, vol. 21, pp. 629–637, 1975.
- [2] W. Gu, R. Koetter, M. Effros, and T. Ho, “On source coding with coded side information for a binary source with binary side information,” *ISIT 2007, Nice, France, June 24 - June 29*, pp. 1456–1460, 2007.
- [3] T. M. Cover and J. A. Thomas, *Elements of information theory, 2nd edition*. Wiley, 2006.

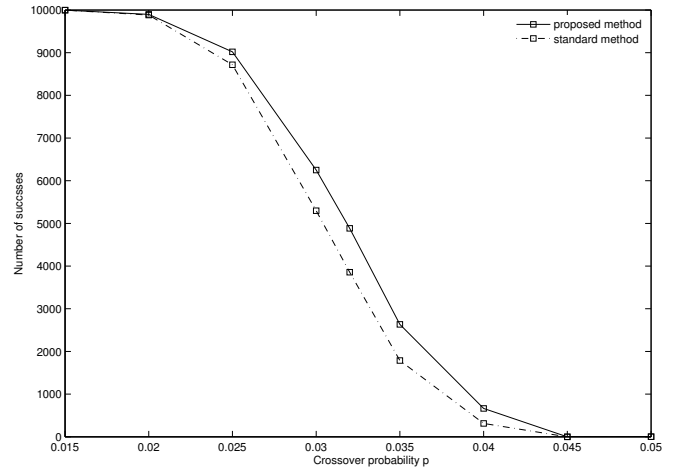


FIGURE 6 – Comparaison entre la méthode standard et la méthode proposée avec $R_X = 0.5$, $R_Y = 5/6$, $T = 30$, $t = 15$, $N_t = 18$ et un Viterbi à entrée souple pour effectuer la projection : Nombre de succès en fonction de la probabilité d’erreur du canal p .

- [4] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Communications Letters*, vol. 6, pp. 440–442, 2002.
- [5] A. R. Calderbank, P. C. Fishburn, and A. Rabinovich, “Covering properties of convolutional codes and associated lattices,” *IEEE Trans. on Information Theory*, vol. 41, pp. 732–746, 1995.
- [6] T. Richardson and R. Urbanke, “Design of capacity-approaching irregular low-density parity check codes,” *IEEE Trans. on Information Theory*, vol. 47, pp. 619–637, 2001.
- [7] H.-H. Tang and M.-C. Lin, “On $(n, n - 1)$ convolutional codes with low trellis complexity,” *IEEE Trans. on Communications*, vol. 50, pp. 37–47, 2002.
- [8] A. Kavcic, X. Ma, and M. Mitzenmacher, “Binary intersymbol interference channel : Gallager codes, density evolution, and code performance bounds,” *IEEE Trans. on Information Theory*, vol. 49, pp. 1636–1652, 2003.