

# BEP\_DM\_DM1

Clément Thion, Thomas Besognet

November 2019

## 1 Les grandes lignes

**le contexte** On note  $X$  la variable aléatoire décrivant les états d'une image  $x$  donnée.

On note  $s_{ij}$  un pixel données d'une image ligne  $i$  colonne  $j$ , et  $X_s$  la variable aléatoire décrivant son état  $x_s$ , à valeur dans  $E = -1, 1$  pour nous qui travaillons sur des images en noir et blanc.

Pour restaurer une image, on a besoin de connaître la probabilité de  $X_s$  sur l'ensemble de ses états possible  $E$ . Pour se faire, on va partir du principe qu'à partir des autres pixels de l'image ( $s_s \neq s_{ij}$ ), on peut déduire quel est l'état le plus logique, le plus probable, de  $s_{ij}$ . Par exemple si on a l'image d'une pelouse verte avec un pixel en plein milieu tout blanc, on va logiquement avoir tendance à mettre ce pixel en vert, puisqu'il est le seul pixel blanc et que tout est vert autour de lui.

On fait en plus une hypothèse un peu plus forte en supposant qu'il suffit de connaître les plus proches voisins de  $s_{ij}$  pour déterminer son état le plus probable, et non forcément toute l'image. C'est l'hypothèse Markovienne, et ça va nous permettre de ménager les programmes...

Deux questions se posent: comment définir le voisinage, et quel outil utiliser pour représenter l'ensemble des pixels de ce voisinage.

**le voisinage d'un pixel** Pour le voisinage  $V_s$  de  $s_{ij}$ , on OOOO

**le potentiel d'un pixel** Pour représenter numériquement le voisinage de  $s_{ij}$ , on va introduire la notion de potentiel  $U(x_s)$ , aussi appelé énergie, de  $s_{ij}$  dans un état  $x_s$  donné. La valeur de  $U(x_s)$  varie selon l'état de  $s_{ij}$ , mais aussi selon l'état des voisins de  $s_{ij}$ . L'expression de  $U$  diffère selon les modèles, et certains seront plus adaptés pour des images en noir et blanc qu'en couleur... Dans notre cas, on va utiliser l'expression du modèle d'Ising[1], c'est à dire:

$$U(x_s) = \beta x_s \sum_{t \in V_s} x_t + B x_s$$

Maintenant comment mesure t-on  $P(X_s = x_s)$  avec  $U(x_s)$ ? On va utiliser la mesure de Gibbs:

## 2 Démonstrations

démo hypothèse Markov pour potentiel locaux

$$\begin{aligned} P(X_s = x_s | X^s = x^s) &= \frac{P(X_s = x_s, X^s = x^s)}{P(X^s = x^s)} \text{ formule de Bayes} \\ &= \frac{P(X = x)}{P(X^s = x^s)} \\ &= \frac{e^{-U(x)}}{e^{-U(x^s)}} \end{aligned}$$

$$\text{On a } U(x^s) = U_{\bar{s}}(x) = \sum_{c \in \mathcal{C}, s \notin c} U_c(x)$$

$$\begin{aligned} \text{Et } U(x) &= \sum_{c \in \mathcal{C}} U_c(x) \\ &= \sum_{c \in \mathcal{C}, s \in c} U_c(x) + \sum_{c \in \mathcal{C}, s \notin c} U_c(x) \\ &= U_s(x_s, V_s) + U_{\bar{s}}(x) \end{aligned}$$

$$\begin{aligned} \text{Donc, } P(X_s = x_s | X^s = x^s) &= \frac{\exp(-U_s(x_s, V_s) - U_{\bar{s}}(x))}{\exp(-U_{\bar{s}}(x))} \\ &= \frac{\exp(-U_s(x_s, V_s))}{\exp(-U_{\bar{s}}(x) + U_{\bar{s}}(x))} \\ &= \exp(-U_s(x_s, V_s)) \text{ ce qui justifie l'hypothèse markovienne} \end{aligned}$$

## 3 Problématiques

- Quelle est la probabilité d'avoir une correspondance à c% après n itération d'algorithme de recuit simulé? De Gibbs? De Métropolis?  
*On va faire un algo de montecarlo dans montecarlo, et introduire une mesure du pourcentage de correspondance*
- Quelle est la probabilité d'avoir une correspondance à c% après k itérations d'algo sans modification d'état?  
*cas ou on demande à l'algo de s'arrêter uniquement si il aucun changement de site ne se fait, sur k itérations consécutives (attention donc à ce que k ne soit pas trop grand pour que l'on ne tombe pas en boucle infinie, ni trop petit pour que l'algo tourne quand même un minimum*
- Peut-on gagner en précision en faisant plusieurs fois un recuit simulé, puis en faisant une "moyenne" des images reçues?  
*générer plusieurs image, puis les "additionner"*
  - Là encore, comment évolue la proba de correspondance avec le nombre d'images additionnées?
- Peut-on gagner en rapidité dans nos algorithmes en parcourant les sites non plus un par un mais n par n ?  
*pour une image de N\*N pixels, on peut travailler sur N/2 pixels en même temps sans problème de modification de voisinage sur une même étape*

## 4 objectifs pratiques

Algorithme de Gibbs et Metropolis avec des champs de Markov aléatoires donnés (ising, potts, markovien-gaussien)

Recuit simulé

Interface graphique pour comparaison des différents algorithmes [1]

## References

[1]