Structures de données et algorithmes
Projet 3: analyse d'images
Éléments de solution

Pierre Geurts – Jean-Michel Begon – Romain Mormont

1 Détermination analytique des v_i

Si les p_j sont connus, l'erreur associée à une fonction g peut se réécrire sous la forme suivante :

$$Err(g) = \sum_{j=1}^{k} \sum_{i=p_{j-1}}^{p_j-1} h[i](i-v_j)^2,$$

où on a défini $p_0 = 0$ et $p_k = n$ pour simplifier les notations. Les valeurs de v_j pouvant être choisies de manière indépendante et chaque terme de la somme sur j ne faisant intervenir qu'un seul v_j, l'erreur peut être minimisée en determinant chaque v_j de manière à minimiser le $j$ème terme de la somme, c'est-à-dire en calculant :

$$v_j^* = \arg\min_v \sum_{i=p_{j-1}}^{p_j-1} h[i](i-v)^2.$$

Cette équation est celle d'une parabole (en v). Effectivement, on peut la réécrire

$$\sum_{i=p_{j-1}}^{p_j-1} h[i](i-v)^2 = v^2 \sum_{i=p_{j-1}}^{p_j-1} h[i] - 2v \sum_{i=p_{j-1}}^{p_j-1} h[i]i + \sum_{i=p_{j-1}}^{p_j-1} h[i]i^2 = av^2 + bv + c$$

Puisque le coefficient en v^2 est positif, la parabole atteint un minimum en

$$v_l^* = \frac{-b}{2a} = \frac{\sum_{i=p_{j-1}}^{p_j-1} h[i]i}{\sum_{i=p_{j-1}}^{p_j-1} h[i]}. \quad (1)$$

Cette valeur n'est cependant pas nécessairement un entier. Etant donné que la parabole est symétrique, la solution que nous cherchons est donc l'arrondi au plus proche entier.

Etant donné les p_j, il est donc possible de déterminer directement les v_j qui minimisent l'erreur et le problème algorithmique peut donc se réduire à la détermination des p_j.

2 Approche par recherche exhaustive

L'approche par recherche exhaustive consiste à énumérer toutes les combinaisons possibles de p_j, à calculer l'erreur correspondant à chacune de ces combinaisons (en déterminant les v_j à partir du résultat précédent) et finalement à renvoyer celle qui minimise l'erreur.

On souhaite compresser n niveaux en k niveaux. Il faut donc choisir $k-1$ valeurs seuils $(p_1, \ldots, p_{k-1})$. Ces $k-1$ valeurs de seuils sont à choisir dans l'ensemble $\{1, \ldots n-1\}$, la valeur 0 étant exclue car elle créerait un intervalle vide. Il y a donc C_{n-1}^{k-1} combinaisons possibles de p_j. Calculer les v_j selon (1) et évaluer l'erreur de chacune est $\Theta(n)$ au total et retenir l'erreur minimale est linéaire par rapport au nombre de combinaisons. Au final, la complexité de l'approche exhaustive est donc $\Theta(nC_{n-1}^{k-1})$.

3 Approche gloutonne

Il y avait plusieurs possibilités. L'énoncé mentionnait l'approche suivante :

> [...] tirer au hasard un seuil p_j et partitionner l'intervalle $[p_j, p_{j+1}[$ en deux en ajoutant un seuil en $\frac{p_j+p_{j+1}}{2}$.

Cette approche a deux inconvénients. D'abord, rien n'indique que mettre le nouveau seuil en $\frac{p_j+p_{j+1}}{2}$ soit optimal. Une meilleur approche consiste à déterminer la position du nouveau seuil p^* dans $[p_j, p_{j+1}[$ de manière à minimiser l'erreur. Etant donné (1), il est possible de calculer ce p^* en un temps linéaire par rapport à la longueur de l'intervalle $p_{j+1} - p_j$.

En outre, tirer un des seuils au hasard n'est pas un choix local optimal. Un meilleur choix glouton est de choisir l'intervalle $[p_{i-1}, p_i]$ sur lequel la réduction d'erreur serait la plus grande en choisissant un nouveau seuil p^* dans cet intervalle comme décrit précédemment. Il est possible également de déterminer cet intervalle en un temps linéaire par rapport à la taille de l'histogramme n.

Cette solution gloutonne serait donc au final $\Theta(nk)$.

4 Programmation dynamique

Dans ce qui suit, les niveaux de l'histogramme sont indexés de 0 *à* $N-1$ *comme dans l'énoncé.*

Soient

N le nombre de niveaux de l'histogramme H.

K le nombre de niveaux pour la compression ($1 \leq K \leq N$).

$ErrMin(n, k)$ l'erreur de compression minimale des n premiers niveaux de l'histogramme en k niveaux ($1 \leq n \leq N$, $1 \leq k \leq K$).

$L(i, j)$ l'erreur minimale pour compresser l'histogramme H entre les niveaux i (inclus) et j (exclu) en un seul niveau ($0 \leq i < j < N$).

On peut définir $ErrMin(n, k)$ récursivement comme suit :

$$ErrMin(n,k) = \begin{cases} 0, & \text{si } k \geq n \\ L(0,n), & \text{si } k = 1,\ k < n \\ \min_{k-1 \leq j < n} \left(ErrMin(j, k-1) + L(j,n) \right), & \text{sinon} \end{cases} \quad (2)$$

Le premier cas de base signifie que si on dispose d'au moins n niveaux ($k \leq n$), on peut compresser sans perte (l'erreur vaut 0). Le second cas de base est le coût de la compression en un seul niveau des n premières valeurs de l'histogramme.

Le cas inductif trouve où placer le seuil le plus à gauche pour minimiser l'erreur. On doit faire $k-1$ compressions optimales sur les j premiers niveaux et une sur les niveaux allant de j à n.

Une alternative consiste à étendre la définition de $ErrMin(n,k)$ aux cas $n=0$ et $k=0$:

$$ErrMin(n,k) = \begin{cases} 0, & \text{si } k \geq n \\ +\infty, & \text{si } k = 0,\ k < n \\ \min_{0 \leq j < n} \left(ErrMin(j, k-1) + L(j,n) \right), & \text{sinon} \end{cases} \tag{3}$$

Le second cas de base signifie qu'il est impossible de compresser un histogramme d'au moins un niveau en zéro niveau.

Lorsque $k=1$, le cas inductif se réduit à $L(0,n)$. En effet, tous les membres du minimum tels que $j \geq 1$ vaudront $+\infty$, puisqu'on atteint le second cas de base. Quant au membre en $j=0$, $ErrMin(0,0)=0$, il reste donc seulement le second terme : $L(0,n)$.

En ce qui concerne l'implémentation, le calcul de $L(i,j)$ demandant de l'ordre de $j-i$ opérations (en utilisant la formulation analytique (1)), une implémentation naïve directe du calcul de $ErrMin(N,K)$ demandera de l'ordre de KN^3 opérations : on doit remplir une table de taille NK et le calcul pour chaque case correspondant au cas inductif demande N^2 opérations (N opérations pour le $\min_j$ fois N opérations pour le calcul de $L(j,n)$). Une manière de réduire cette complexité est de pré-calculer les valeurs $L(i,j)$ en les stockant dans une table de taille $N \times N$. Le pré-calcul naïf de cette table demandera N^3 opérations, ce qui permet d'atteindre la complexité $\Theta(N^3 + KN^2)$ suggérée dans l'énoncé. Il est cependant possible d'effectuer le pré-calcul de toutes les valeurs $L(i,j)$ en N^2 et donc de réduire la complexité globale à $\Theta(N^2 + KN^2) = \Theta(KN^2)$. Cette optimisation, non détaillée ici, exploite le fait qu'on peut calculer $L(i,j)$ à partir de $L(i,j-1)$ en $O(1)$ opérations.