Математические структуры и моделирование
2009, вып. 20, с. 97–106

УДК 519.688

КЛАСТЕРИЗАЦИЯ МАТРИЦ С ПРОПУСКАМИ КАК МЕТОД ВОССТАНОВЛЕНИЯ МУЛЬТИМЕДИЙНОЙ ИНФОРМАЦИИ

И.Б. Ларионов

Рассматривается метод кластеризации матриц с пропусками как метод восстановления мультимедийной информации на примере графических файлов.

Введение

Надежность современных накопителей достаточно высока. Но всегда есть вероятность того, что какие-либо файлы будут испорчены. Так же современные пиринговые сети используют достаточно эффективные алгоритмы хеширования. Однако иногда происходит изменение данных, которые не заметны для внутренних алгоритмов контроля целостности носителей или для алгоритмов хеширования. Подобные изменения иногда приводят к достаточно сильному изменению данных (на примере мультимедийных данных) в связи с тем, что современные алгоритмы сжатия с потерями неустойчивы к повреждениям.

В связи с этим встает задача восстановления безвозвратно потерянной информации. Существует множество способов сглаживания потерь в мультимедийных данных с использованием интерполяции. Для случаев небольших потерь такой метод, как интерполяция кубическими сплайнами, позволяет добиться достаточно хороших результатов, что будет показано ниже. Однако при потере существенного количества информации распространенные методы интерполяции неэффективны. Такие случаи требуют более интеллектуального подхода.

Рассмотрим способ восстановления поврежденных мультимедийных данных с помощью сингулярного разложения матриц с пропусками.

1. Сингулярное разложение матриц с пропусками

Идея сингулярного разложения матриц, содержащих пропуски, на сумму одноранговых матриц была предложена старшим научным сотрудником Новосибирского института математики С.В. Макаровым. И хотя материал этого раздела

Омский Государственный Университет им.Ф.М.Достоевского.
E-mail: `igor@1.ws`

непосредственно в обработке данных не используется, он дает нам простейший пример и прототип для дальнейших построений.

Пусть задана прямоугольная матрица $A = (a_{ij})$, клетки которой заполнены действительными числами или значком @, означающим отсутствие данных. Требуется представить исходную матрицу A в виде суммы одноранговых матриц P_q:

$$A = \sum_q P_q.$$

Таким образом, ставится задача поиска наилучшего приближения А матрицей вида $x_i y_j$ методом наименьших квадратов:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j)^2 \to \min. \tag{1}$$

Решение этой задачи дается последовательными итерациями по явным формулам. При фиксированном векторе y_i значения x_i, доставляющие минимум форме (1), однозначно и просто определяются из равенств $\delta\Phi/\delta x_i$:

$$\frac{\delta\Phi}{\delta x_i} = -2 \sum_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j) y_j = 0,$$

$$x_i = \left(\sum_{\substack{j \\ a_{ij} \neq @}} a_{ij} y_j \right) \Big/ \left(\sum_{\substack{j \\ a_{ij} \neq @}} (y_j)^2 \right).$$

Введем обобщенное на случай данных с пробелами определение скалярного произведения $(\bullet, \bullet)_a$ и нормы $|| \bullet ||_a$.

Определение 1. Скалярное произведение $(y_1, y_2)_a$ векторов y_1 и y_2 называется скалярным произведением по известным компонентам вектора a и считается следующим образом:

$$(y_1, y_2)_a = \sum_{\substack{i \\ a_i \neq @}} y_{1i} y_{2i}.$$

Определение 2. Норма $||y||_a$ вектора y называется нормой по известным компонентам вектора a и считается следующим образом:

$$||y||_a = \sqrt{(y, y)_a} = \sqrt{\sum_{\substack{i \\ a_i \neq @}} y_i^2}.$$

С учетом этих определений можно заметить, что значения проекций x_i находятся как нормированное скалярное произведение вектора данных a_i (i-ая строка матрицы A) на вектор y:

$$x_i = \frac{(a_i, y)_{a_i}}{||y||_{a_i}{}^2},$$

где, напомним, скалярное произведение $(\bullet, \bullet)_{a_i}$ и норма $|| \bullet ||_{a_i}$ вычисляются по известным компонентам вектора a_i, т.е. мы имеем дело с обобщенным на случай данных с пробелами скалярным произведением и нормой.

Аналогично и при фиксированном векторе x_i значение y_i, доставляющее минимум форме (1), определяется явно из равенств $\delta\Phi/\delta y_j = 0$:

$$\frac{\delta\Phi}{\delta y_j} = -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - x_i y_j) x_i = 0,$$

$$y_i = \Bigg(\sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i \Bigg) \Bigg/ \Bigg(\sum_{\substack{i \\ a_{ij} \neq @}} (x_i)^2 \Bigg).$$

Аналогично вычислению проекции x_i через скалярное произведение можно записать соответствующую проекцию j-го столбца a_j матрицы A на вектор x:

$$y_j = \frac{(a_j, x)_{a_j}}{||x||_{a_j}{}^2},$$

где скалярное произведение и норма вычисляются по известным компонентам вектора a_j. Процесс вычисления итерационный, поэтому в качестве начального приближения вектора y возьмем случайное значение, но потребуем, чтобы y был единичной длины:

$$y - \text{случайный, нормирован на 1 (т.е.}||y||^2 = \sum_j y_j{}^2 = 1).$$

В качестве критерия остановки будем использовать малость относительного улучшения значения минимизируемого функционала на итерации, т.е. критерий остановки малость относительного улучшения $\Delta\Phi/\phi$, где $\Delta\Phi$ – полученное за цикл уменьшение значения Φ, а Φ – само текущее значение. Естественно и использование второго критерия остановки - малость самого значения Φ.

Таким образом, итерационная процедура останавливается, если $\Delta\Phi/\Phi < \epsilon$ или $\Phi < \delta$ для некоторых ϵ, $\delta < 0$.

В результате для матрицы A получили наилучшее приближение матрицей P_1 вида $x_i y_j$. Далее, из матрицы A вычитаем полученную матрицу P_1, и для полученной матрицы уклонений $A - P_1$ вновь ищем наилучшее приближение P_2 этого же вида и т.д., пока, например, норма A не приблизится в достаточной степени к нолю.

В результате получили опять же итерационную процедуру разложения матрицы A в виде суммы матриц ранга 1, т.е. $A = P_1 + P_2 + \cdots + P_q$.

Из теории сингулярного разложения матрицы в виде суммы одноранговых матриц известно, что в случае полной (без пробелов) матрицы число полученных одноранговых матриц не превышает число столбцов исходной матрицы. В общем же случае при наличии пробелов это не так.

2. Описание наиболее распространенных повреждений графических файлов

В ходе эксперимента были предприняты попытки восстановить графические файлы формата BMP. Этот формат был выбран как наиболее простой и как один из форматов, в котором отсутствует сжатие, при котором изменение одного бита повлечет за собой изменение всех данных, идущих после этого бита (как это происходит в алгоритме JPEG). Также этот формат наиболее похож на формат RAW, который используют все профессиональные фотографы. Формат RAW отличается от BMP только заголовком, что позволяет рассматривать два этих формата как один и тот же. При стандартной потере одного блока (будь то блок на жестком диске или пакет в сети) данный формат теряет горизонтальный блок точек (пикселей) шириной в один пиксель. Это вызвано тем, что все 3 (а иногда и 4, если используется alpha-канал) канала изображения идут последовательно друг за другом, сгруппированные в блок из 3 (4) байт для одного пикселя.

В рассматриваемом формате данные хранятся в формате (a)RGB, т.е. хранятся отдельные составляющие для красного, зеленого и синего цветов (так же прозрачность alpha-канала). При моделировании восстановления данных было принято решение об использовании цветовой модели YUV, т.к. эта модель наиболее распространена (современное аналоговое телевидение, MPEG-подобные кодеки).

В модели YUV есть три составляющие: Y - яркость и две цветоразностных - U и V. Преобразование из RGB в YUV происходит по следующим формулам:

$$Y = 0.299R + 0.587G + 0.114B,$$

$$U = -0.14713R - 0.28886G + 0.436B,$$

$$V = 0.615R - 0.51499G - 0.10001B.$$

Из YUV в RGB:

$$R = Y + 1.13983V,$$

$$G = Y - 0.39465U - 0.58060,$$

$$B = Y + 2.03211U.$$

Как видно из этих формул, данная цветовая модель позволяет проводить эксперименты по восстановлению, используя только одну составляющую - яркость, которая в свою очередь содержит информацию о всех трех цветовых составляющих. Результатов данных испытаний вполне достаточно для сравнения результатов метода восстановления с результатами распространенных методов интерполяции.

В качестве эталонного метода интерполяции был выбран метод бикубических сплайнов, как наиболее ресурсоёмкий и наиболее распространенный при обработке изображений.

3. Интерполяция бикубическими сплайнами

Пусть на отрезке $a \leq \xi \leq b$ задана сетка $\omega = \{x_i : x_0 = a < x_1 < \dots < x_i < x_n = b\}$ и в её узлах заданы значения функции $y(x)$, равные $y(x_0) = y_0, \cdots, y(x_i) = y_i, \cdots, y(x_n) = y_n$. Требуется построить интерполянту — функцию $f(x)$, совпадающую с функцией $y(x)$ в узлах сетки:

$$f(x_i) = y_i, i = 0, 1, \cdots, n. \tag{2}$$

Основная цель интерполяции — получить быстрый (экономичный) алгоритм вычисления значений $f(x)$ для значений x, не содержащихся в таблице данных.

Интерполируюшие функции $f(x)$, как правило, строятся в виде линейных комбинаций некоторых элементарных функций:

$$f(x) = \sum_{k=0}^{N} c_k \Phi_k(x_i),$$

где $\{\Phi_k(x)\}$ — фиксированный линейно независимые функции, $c_0, c_1, \cdots, c_n$ — не определенные пока коэффициенты. Из условия (2) получаем систему из $n+1$ уравнений относительно коэффициентов $\{c_k\}$:

$$\sum_{k=0}^{N} c_k \Phi_k(x_i) = y_i, i = 0, 1, \cdots, n.$$

Предположим, что система функций $\Phi_k(x)$ такова, что при любом выборе узлов $a < x_1 < \cdots < x_i < \cdots < x_n = b$ отличен от нуля определитель системы:

$$\Delta(\Phi) = \begin{vmatrix} \Phi_0(x_0) & \Phi_1(x_0) & \cdots & \Phi_n(x_0) \\ \Phi_0(x_1) & \Phi_1(x_1) & \cdots & \Phi_n(x_1) \\ \cdots & \cdots & \cdots & \cdots \\ \Phi_0(x_n) & \Phi_1(x_n) & \cdots & \Phi_n(x_n) \end{vmatrix}.$$

Тогда по заданным $y_i(i = 1, \cdots, n)$ однозначно определяются коэффициенты $c_k(k = 1, \cdots, n)$.

Интерполяция кубическими сплайнами является частным случаем кусочно-полиномиальной интерполяции. В этом специальном случае между любыми двумя соседними узлами функция интерполируется кубическим полиномом, его коэффициенты на каждом интервале определяются из условий сопряжения в узлах:

$$f_i = y_i, f'(x_i - 0) = f'(x_i + 0), f''(x_i - 0) = f''(x_i + 0), i = 1, 2, \cdots, n - 1.$$

Кроме того, на границе при $x = x_0$ и $x = x_n$ ставятся условия

$$f''(x_0) = 0, f''(x_n) = 0. \tag{3}$$

Будем искать кубический полином в виде

$$f(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, x_{i-1} \leq \xi \leq \xi_i. \tag{4}$$

Из условия $f_i = y_i$ имеем

$$f(x_{i-1}) = a_i = y_{i-1}, f(x_i) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, h_i = x_i - x_{i-1}, i = 1, 2, \cdots, n-1. \quad (5)$$

Вычислим производные:

$$f'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2, f''(x) = 2c_i + 6d_i(x - x_{i-1}), x_{i-1} \le \xi \le x_i,$$

и потребуем их непрерывности при $x = x_i$:

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2, c_{i+1} = c_i + 3d_i h_i, i = 1, 2, \cdots, n-1. \quad (6)$$

Общее число неизвестных коэффициентов, очевидно, равно $4n$, число уравнений (5) и (6) равно $4n-2$. Недостающие два уравнения получаем из условия (3) при $x = x_0$ и $x = x_n$:

$$c_1 = 0, c_n + 3d_n h_n = 0.$$

Выражение из (6) $d_i = \frac{c_{i+1} - c_i}{3h_i}$; подставляя это выражение в (5) и исключая $a_i = y_{i-1}$, получим

$$b_i = \left[\frac{y_i - y_{i-1}}{h_i}\right] - \frac{1}{3} h_i (c_{i+1} + 2c_i), i = 1, 2, \cdots, n-1, b_n = \left[\frac{y_n - y_{n-1}}{h_n}\right] - \frac{2}{3} h_n c_n. \quad (7)$$

Подставив теперь выражения для b_i, b_{i+1} и d_i в первую формулу (6), после несложных преобразований получаем для определения c_i разностное уравнение второго порядка

$$h_i c_i + 2(h_i + h_{i+1}) c_{i+1} + h_{i+1} c_{i+2} = 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_{i+1}}\right), i = 1, 2, \cdots, n-1. \quad (8)$$

С краевыми условиями

$$c_1 = 0, c_{n+1} = 0. \quad (9)$$

Условие $c_{n+1} = 0$ эквивалентно условию $c_n + 3d_n h_n$ =0 и уравнению $c_{i+1} = c_i + d_i h_i$. Разностное уравнение (8) с условиями (9) можно решить методом прогонки, представив в виде системы линейных алгебраических уравнений вида $A * x = F$, где вектор x соответствует вектору $\{c_i\}$, вектор F поэлементно равен правой части уравнения (8), а матрица A имеет следующий вид:

$$A = \begin{pmatrix} C_1 & B_1 & 0 & 0 & \cdots & 0 & 0 \\ A_2 & C_2 & B_2 & 0 & \cdots & 0 & 0 \\ 0 & A_3 & C_3 & B_3 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & B_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & A_n & C_n \end{pmatrix}, \quad (10)$$

где $A_i = h_i, i = 2, \cdots, n, B_i = h_{i+1}, i = 1, \cdots, n-1$ и $C_i = 2(h_i + h_{i+1}), i = 1, \cdots, n.$

4. Метрики сравнений изображений

В ходе работы встал вопрос о методе сравнения качества восстановления изображений. Были выбраны насколько метрик, основанные на разнице пикселей.

4.1. Метрика Минковского (MM)

Норма разницы между изображениями может быть посчитана путем взятия Минковского среднего разниц пикселей сначала пространственно, а затем хроматически (т.е. по зонам):

$$\epsilon^\gamma = \frac{1}{K}\sum_{k=1}^{K}\left\{\frac{1}{N^2}\sum_{i,j=0}^{N-1}\left|C_k(i,j) - \hat{C}_k(i,j)\right|^\gamma\right\}^{1/\gamma},$$

где $\hat{C}_k(i,j)$ - значение цветосоставляющей пикселя восстановленного изображения.

Для значения $\gamma = 2$ мы получаем формулу для вычесления среднеквадратической ошибки, которая описана ниже.

В ходе работы использовалось значение $\gamma = \infty$:

$$\epsilon^\infty = \max_{i,j}\sum_{k=1}^{K}\frac{1}{K}\left|C_k(i,j) - \hat{C}_k(i,j)\right| = \max_{i,j}||\mathbf{C}(i,j) - \hat{\mathbf{C}}(i,j)||.$$

4.2. Среднеквадратическая ошибка (MSE)

Данная метрика является частным случаем метрики Минковского:

$$\epsilon = \frac{1}{N^2}\sum_{i,j=0}^{N-1}\left(C(i,j) - \hat{C}(i,j)\right)^2.$$

4.3. Метрика разницы с соседями (DON)

Данная метрика толерантна к сдвигам пикселей и обычно используется в оченке качества сжатия видеоряда.

$$\epsilon = \sqrt{\frac{1}{2(N-w)^2}\sum_{i,j=w/2}^{N-w/2}\left[A^2 + B^2\right]},$$

$$A = \min_{l,m\in w_{i,j}}\left\{d(\mathbf{C}(i,j), \hat{\mathbf{C}}(i,m))\right\},$$

$$B = \min_{l,m\in w_{i,j}}\left\{d(\hat{\mathbf{C}}(i,j), \mathbf{C}(l,m))\right\},$$

где $d(\bullet,\bullet)$ – некая функция расстояния. В данной работе рассматривается Евклидово расстояние.

Нетрудно заметить, что данная метрика при $w = 1$ сводится к среднеквадратической ошибке, описанной выше.

4.4. Метрика многомерного расстояния (MDM)

Данная метрика основывается на том, что большинство изображений хранятся в больших разрешениях (2000 пикселей в каждом измерении и более):

$$\epsilon = \sum_{r=1}^{R} \left(\frac{1}{2^r} \frac{1}{2^{2r-3}} \sum_{i,j=1}^{2^{r-1}} \left[(g_{ij}^R - \hat{g}_{ij}^R)^2 + (g_{ij}^B - \hat{g}_{ij}^B)^2 + (g_{ij}^G - \hat{g}_{ij}^G)^2 \right]^{1/2} \right),$$

где, для примера, g_{ij}^R - среднее значение серой составляющей ij-того блока красной составляющей, а r - относительное разрешение метрики.

5. Сравнение методов восстановления

В ходе работы было принято правило, что существует некая карта утраченных пикселов. Это было сделано для упрощения алгоритмов. На практике определение таковых потерь не является сложным, однако в контексте рассматриваемой темы, данный вопрос не является критичным.

Были реализованы два метода восстановления потерь и 4 метода сравнения восстановленных изображений с эталонным. Для упрощения вычислений метрики вычислялись только по карте потерь.

Было проведено 100 испытаний на каждом из 20 изображений различных типов:

1. Высококачественные фотографии NASA.
2. Высококачественные фотографии пейзажей.
3. Фотографии групп людей.
4. Различные искусственные изображения - логотипы, векторная графика, фрактальные изображения, которые были преобразованы в формат BMP.

Данные типы изображений были выбраны из соображений наиболее полного охвата возможных входных наборов данных.

Для каждого испытания выбирался случайный потерянный блок размером в 64Кб. Причем, начало блока не было кратно ни 64Кб (как номер блока от начала файла), ни 4 байт (как номер потерянного пикселя относительно начала). После выбора границ блока составлялась карта потерянных данных, основываясь на структуре файла и на тех пикселях, любая составляющая которых входила в потерянный блок. Для приближения к реальным условиям пиксель, в котором хотя бы одна составляющая потеряна, признавался потерянным не полностью и обрабатывался отдельно, без помощи цветовой модели YUV.

Результаты, полученные в ходе испытаний, представлены в таблице 1 (результат в отдельной группе взят как среднее арифметическое всех испытаний этой группы).

Таблица разбита на 4 группы по 2 столбца в каждой. Каждая группа отображает результаты соответствующей метрики: MM – метрика Минковского; MSE

Таблица 1. Результаты испытаний

Метрика	MM		MSE		DON		MDM	
Метод	Класт.	Спл.	Класт.	Спл.	Класт.	Спл.	Класт.	Спл.
Высококачественные фотографии NASA	246,51	12,14	6345,20	276,58	876,45	23,48	758,45	90,23
Высококачественные фотографии пейзажей	645,02	34,58	9437,69	234,76	589,32	64,75	923,47	65,89
Фотографии групп людей	234,76	59,23	4753,29	347,60	289,34	75,98	234,76	59,82
Искусственные изображения	96,45	88,23	87,65	62,37	85,23	56,56	96,27	98,43

– среднеквадратическая ошибка; DON – метрика разницы с соседями; MDM – метрика многомерного расстояния.

Из таблицы с результатами видно, что кластеризация данных с пропусками по всем статьям проигрывает интерполяции бикубическими сплайнами, кроме случаев искусственных изображений. Это можно объяснить тем, что в искусственных изображениях доля линейных составляющих велика, в отличие от изображений реального мира.

Выводы

В ходе работы были реализованы два метода восстановления потерянной графической информации – кластеризация матриц с пропусками (линейная одномерная модель) и интерполяция бикубическими сплайнами. Было проведено сравнение этих методов метриками оценки качества восстановления, основанных на разнице значений пикселов.

Результаты испытания методов показали, что кластеризация матриц с пропусками не является достаточно качественным методом для восстановления фотографий. Однако для восстановления изображений, созданных искусственно, данный метод вполне подходит и показывает результаты восстановления, не уступающие в достаточной мере методу интерполяции бикубическими сплайнами.

Результаты показали, что метод восстановления изображений, основанный на кластеризации матриц, имеет право на жизнь. Однако, для достаточно точного восстановления фотографий, необходимо применять модель, отличную от линейной одномерной.

Литература

1. Горбань, А.Н. Итерационный метод главных компонент для таблиц с пробелами / А.Н. Горбань, С.В. Макаров, А.А. Россиев // Третий сибирский конгресс по прикладной и индустриальной математике (ИНПРИМ-98), 22-27 июня 1998. – Тезисы докладов. Ч. 5. – Новосибирск: Изд-во Института математики СО РАН, 1998.
2. Самарский, А.А. Введение в численные методы / А.А. Самарский. – М.: Наука, 1982.
3. Avcibas, I. Image Quality Statistics and Their Use in Steganalysis and Compression / Ismail Avcibas, Bulent Sankur, Lale Akarun, Emin Anarim, Nasir Memon, Yucel Yemez. – 2001.