Optimal Grid Quantization

Mingzhou Song
Department of Computer Science
Queens College of CUNY
Flushing, NY 11367

Robert M. Haralick
Computer Science Doctoral Program
Graduate Center of CUNY
New York, NY 10016

Abstract

Optimal quantization, a non-parametric technique for pattern recognition, determines a compact and efficient density representation of data by optimizing a global quantizer performance measure, which is a weighted combination of average log likelihood, entropy and correct classification probability. In multi-dimensions, we obtain the quantization grid using genetic algorithms. Smoothing is an important aspect as it affects the generalization ability of the quantizer. We propose a fast k neighborhood smoothing algorithm. Optimal quantization is much more efficient than other non-parametric methods. For not very well separated Gaussian mixture models, it produces much better results than the EM algorithm, which fails to converge to the true parameters of the underlying density.

1. Introduction

Non-parametric methods do not have large modelling biases inherent in parametric models. Optimal quantization is to alleviate the on-line computation and storage burden of the standard non-parametric techniques, by finding the most effective non-parametric representation of the data, for given CPU cycles, memory and the targeted performance. It requires intensive off-line training. Ideally, the representation should be compact. Some choices of representation yield very fast on-line algorithms whose time and space complexities are not directly related to the sample size. Optimal quantization is scalable to trade between resources and the performance. Other standard non-parametric models do not scale up or down and in many situations become prohibitive to apply.

The multivariate histogram is an example of quantization for density estimation. [14] has established the consistency results of multivariate histogram density estimates and [16] proposed general partition schemes based on their results. Entropy and likelihood [8, 10, 11] have been used as quantizer measures. Splines or local polynomials can be used as representation of the density of a cell, but they are computational inefficient in multi-dimensions.

Quite many activities [5, 6, 13, 4, 17] concentrate on discretization of multi-class 1-D data, partly because it serves as the building blocks for tree classifier construction. In multi-dimensions, the statistically equivalent blocks of [7] is an early tree structured partitioning scheme. The CART [2] algorithm also uses the tree-structured classifier. Grid based partition schemes are studied, e.g., multivariate ASH [19], STING algorithm in [20], OptiGrid algorithm by [12], STUCCO algorithm of [1], adaptive grids of [15] and maximum marginal entropy quantization by [3]. Most multivariate discretization approaches are greedy. In [10, 11], the grid is acquired randomly. In [19], the grid lines are equally spaced. In [1], the grid is improved by merging adjacent intervals by a hypothesis test. The adaptive grid [15] merges dense cells.

WARPing [9] and averaging shifted histograms [18] are techniques that smooth density estimates of the cells. Otherwise there are relatively few smoothing methods. We would like to have all cells with a non-zero density estimate, but there is no available methods for such including ASH. In addition, smoothing is important in maintaining consistency of density estimates and deserves further study.

We design a genetic algorithm to find a grid globally, to be described in Section 2. Section 3 extends the kernel smoothing idea and uses a fast smoothing method to obtain final density estimates. Section 4 compares the performance of the optimal grid quantization and the EM algorithm for Gaussian mixture data. Finally, Section 5 concludes our study.

2. Grid finding by genetic algorithms

Equal spacing grids are not efficient statistically. Variable spacing grids can dramatically improve the statistical efficiency while having low computational complexity. The grid we consider is shown in Fig. 1.

In genetic algorithms, we use fitness proportionate selection: the chance of an individual being selected is in propor-

1051-4651/02 $17.00 © 2002 IEEE

Figure 1. A grid pattern.

tion to its fitness. The fitness function of a grid G on given data is

$$\varphi(G) = [\exp(J(G))]^{W_J}[\exp(H(G))]^{W_H}[P_c(G)]^{W_c}$$

$J(G)$ is the average data likelihood which is a probability. $H(G)$ is the entropy of the grid. $P_c(G)$ is the correct classification probability. $\varphi(G)$ is a weighted geometric combination of the above components. It is non-negative as it is a product of exponentials.

Alg. 1 Find-Grid-GA outlines the grid finding algorithm. $\mathcal{X}_N$ and $\mathcal{Y}_N$ are a sequence of data and their classes, respectively. It starts with an initial population of N_p random grids. The population evolves itself by going through the selection-reproduction-selection cycle in the **for** loop from line 2 to 16. In every cycle, or generation, N_p children are reproduced in the **for** loop from line 7 to 15. Every execution of the loop produces two children C_1 and C_2 by parents G_1 and G_2. The parents are randomly selected (line 8 and 9) from the population and the chance is in proportion to their fitness values. A cross-over site d_r is randomly decided for the parent chromosomes or grids G_1 and G_2, and it happens with a probability of P_r. Once the cross-over is finished, two children C_1 and C_2 are produced (line 11). For each of the two children grids, some of their decision boundaries are randomly mutated (line 12 to 13). Then these two children are inserted to the next generation (line 14). The fittest grid is kept as G^*. G^* is returned after a certain number of generations have been evolved.

Algorithm 1. Find-Grid-GA($\mathcal{X}_N$, $\mathcal{Y}_N$, P_r, P_u)

1: $N_p \leftarrow$ population size, $N_g \leftarrow$ number of generations
   $\mathcal{P}^0 \leftarrow$ a population of N_P random grids;
2: **for** $j \leftarrow 0$ to $N_g - 1$ **do**
3: **if** $\varphi(G^*) < \max_{G \in \mathcal{P}^j} \varphi(G)$ **then**
4: $G^* \leftarrow \operatorname{argmax}_{G \in \mathcal{P}^j} \varphi(G)$;
5: **end if**
6: $\mathcal{P}^{j+1} \leftarrow \phi$;
7: **for** $i \leftarrow 0$ to $N_p - 1$ with increment of 2 **do**
8: Randomly select a grid G_1 from $\mathcal{P}^j$ with a probability proportion to fitness value;
9: Randomly select a grid G_2 from $\mathcal{P}^j$ with a probability proportion to fitness value;
10: Randomly decide a dimension d_r as the cross-over site;
11: Exchange the decision boundaries of dimensions 1 to d_r between C_1 and C_2 with probability P_r;
12: Mutation of C_1: randomly adjust each decision boundary with probability P_u;
13: Mutation of C_2: randomly adjust each decision boundary with probability P_u;
14: $\mathcal{P}^{j+1} \leftarrow \mathcal{P}^{j+1} \cup \{C_1, C_2\}$;
15: **end for**
16: **end for**
17: **if** $\varphi(G^*) < \max_{G \in \mathcal{P}^{N_g}} \varphi(G)$ **then**
18: $G^* \leftarrow \operatorname{argmax}_{G \in \mathcal{P}^{N_g}} \varphi(G)$;
19: **end if**
20: **return** G^*;

3. k neighborhood cell smoothing

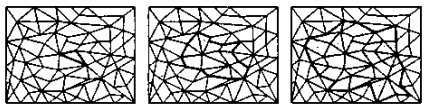
Over-memorization is a serious problem when the emptiness issue is not properly handled, where the quantizer works extraordinarily well on the training sample, but fails on any unseen samples. Smoothing is an effort to reduce the variance of the density estimates.

Let $V_k(q)$ be the volume of a minimum neighborhood containing at least k points. We do not require the shape of a neighborhood be a ball in the metric space chosen. Let k_q be the actual number of points in the neighborhood. A smoothed probability density estimate of cell q is

$$p(q) = \frac{\rho(q)}{\sum_r \rho(r) V(r)} = \frac{k_q}{V_k(q) \sum_r \frac{k_r}{V_k(r)} V(r)} \qquad (1)$$

Searching for the exact k-th nearest neighbor is costly. We introduce the k neighborhood smoothing algorithm. We call cell a and b *neighbor cells* if they share at least a partial boundary. The *radius* 0 *neighborhood* of cell q is the cell itself, designated by $\mathcal{N}(q,0)$. The *radius* R ($R \in \mathbb{Z}^+$) *neighborhood* of cell q, $\mathcal{N}(q,R)$, is the union of the radius $R-1$ neighborhood $\mathcal{N}(q,R-1)$, and the set of all the neighbor cells of $\mathcal{N}(q,R-1)$. k *neighborhood* of a cell is its smallest radius R neighborhood containing at least k points. Fig. 2 shows an example. In Fig. 2(a), the cell of interest is the one in gray, which is also its own radius 0 neighborhood. Fig 2(b) and (c) draw its radius 1 and 2 neighborhoods.

Alg. 2 Radius-Smoothing is based on the k neighborhood concept. The algorithm searches for a minimum radius R neighborhood of current cell with at least k data points. Then the density of the k neighborhood is assigned to the cell as its density estimate. M is the total mass on the density support. M can be considered an adjusted data count by smoothing and is related to N. For the cells containing less than k data points, the initial guess of R is the radius of the k neighborhood of the previous cell. To make this initial guess more realistic, we shall go through the cells in an or-



(a) $R = 0$ (b) $R = 1$ (c) $R = 2$

Figure 2. Radius R neighborhood of a cell.

der that every pair of adjacent cells are neighbor cells. k_q is the actual total number of data points in the current radius R neighborhood. Based on the data count in the current radius R neighborhood, we either increase R until there are at least k data points in the neighborhood, or decrease R until the $R-1$ neighborhood contains less than k data points.

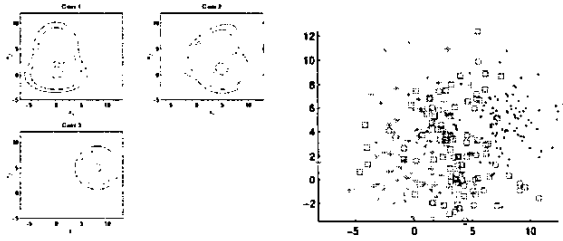
Algorithm 2. Radius-Smoothing(Q, k)

$M \leftarrow 0, R \leftarrow -1$;
for each cell q **do**
  if $N(q) = k$ **then**
    $R \leftarrow 0, k_q \leftarrow N(q)$;
  else
    if $R < 0$ **then**
      $R \leftarrow 0, k_q \leftarrow N(q)$;
    else
      $\mathcal{A} \leftarrow \mathcal{N}(q^-, R) \cap \mathcal{N}(q, R)$;
      $k_q \leftarrow k_q - |\mathcal{N}(q^-, R) - \mathcal{A}| + |\mathcal{N}(q, R) - \mathcal{A}|$;
    end if
    while $k_q > k$ and $R > 0$ **do**
      $k_q \leftarrow k_q - |\mathcal{N}(q, R) - \mathcal{N}(q, R-1)|, R \leftarrow R - 1$;
    end while
    while $k_q < k$ and $R < R_{\max}$ **do**
      $k_q \leftarrow k_q + |\mathcal{N}(q, R+1) - \mathcal{N}(q, R)|, R \leftarrow R + 1$;
    end while
  end if
  $\rho(q) \leftarrow \frac{k_q}{V(\mathcal{N}(q,R))}, M \leftarrow M + \rho(q)V(q), q^- \leftarrow q$
end for
for each cell q **do**
  $p(q) \leftarrow \frac{\rho(q)}{M}$;
end for

The k neighborhood search of a cell checks at most all L cells. Thus the total time of smoothing is $O(L^2)$. However, because we (1) use the radius of the previous k neighborhood as an initial guess instead of starting from $R = 0$ and (2) find the data count in the current neighborhood by adjusting data count in previous neighborhood, we expect a constant time in finding the k neighborhood of a cell. Hence, we would have total expected running time of $O(L)$ for doing all the cells. The optimal control parameter k^* is cross-validated, such that it maximizes the average fitness value.

4. Experimental comparison with EM algorithm

In our experiment, we have three classes, each with a different Gaussian mixtures p.d.f., shown in Fig 3(a). Fig 3(b) shows the scatter plot of a simulated data set. The data set we will use is a sample of 3,000,000 with weighted class assignment.



(a) Class p.d.f. contour plots. (b) Sample scatter plot.

Figure 3. True class p.d.f. and simulated data.

For the EM algorithm, we set the maximum number of mixture components for each class to 2. The actual number is cross-validated. Fig. 4 presents the contour plots of the estimated class p.d.f.

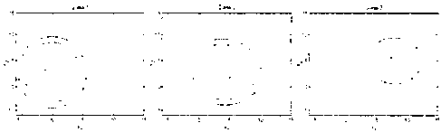


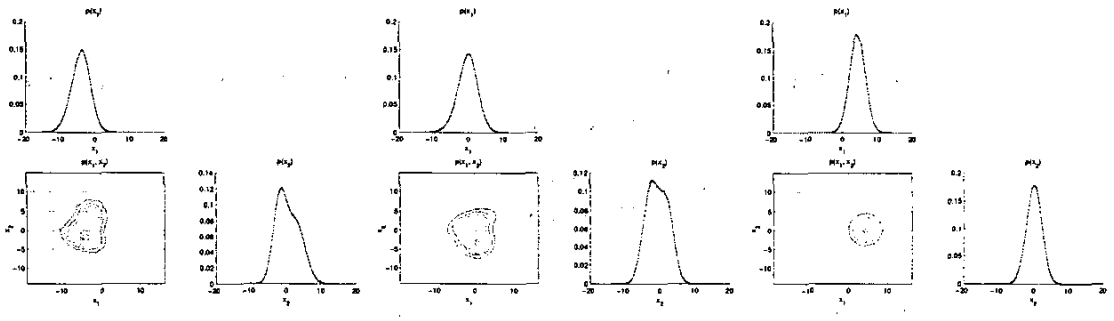
Figure 4. EM algorithm: estimated p.d.f.

For the optimal grid quantization, Fig. 5 presents density estimates and their marginal densities using approximately 65536 quantization cells. Data are pre-rotated for efficient use of the given quantization levels. The relative quantization levels in a dimension is in proportion to the marginal histogram entropy in that dimension.

It is evident that EM did not converge to the right parameters of the densities. The grid has quantization effect locally, but it captures the underlying true densities much better than the EM algorithm.

5. Conclusions

Optimal grid quantization substantially avoids the time and space inefficiency of standard non-parametric methods,



(a) Class 1. (b) Class 2. (c) Class 3.

Figure 5. Optimal grid quantization: estimated class p.d.f.'s with 65536 quantization cells.

while maintaining their benefit. The improvement in efficiency is a result of the adaptivity of optimal quantization. Resources are only spent in important regions. In addition, better results can always be achieved with more quantization levels, which provides a very natural way of balancing resources and performance. For not very well separated Gaussian mixture models, where the convergence of the EM algorithm fails or is very slow, optimal grid quantization has produced much better results.

While we have demonstrated the theoretical feasibility of optimal quantization, we are in the process of evaluating it on more real data sets. In real-world applications, when parametric models can not be assumed or the convergence fails, optimal quantization is a definite solution to produce efficient and consistent density representations of data.

References

[1] S. D. Bay. Multivariate discretization for set mining. *Knowledge and Information Systems*, 3(4):491–512, 2001.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks/Cole, CA, 1984.

[3] T. Chau. Marginal maximum entropy partitioning yields asymptotically consistent probability density functions. *IEEE Trans. PAMI*, 23(4):414–417, Apr. 2001.

[4] T. Elomaa and J. Rousu. General and efficient multisplitting of numerical attributes. *Machine Learning*, 36:201–44, 1999.

[5] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. 13th Int'l Joint Conf. AI*, pages 1022–1029, 1993.

[6] T. Fulton, S. Kasif, and S. L. Salzberg. Efficient algorithms for finding multi-way splits for decision trees. In *Proc. 12th Int'l Conf. Machine Learning*, pages 244–251, 1995.

[7] M. P. Gessaman. A consistent nonparametric multivariate density estimator based on statistically equivalent blocks. *The Annals of Mathematical Statistics*, 41:1344–46, 1970.

[8] R. M. Haralick. The table look-up rule. *Communications in Statistics – Theory and Methods*, A5(12):1163–91, 1976.

[9] W. Härdle and D. W. Scott. Smoothing by weighted averaging of rounded points. *Computational Statistics*, 7:97–128, 1992.

[10] L. B. Hearne and E. J. Wegman. Maximum entropy density estimation using random tessellations. In *Computing Science and Statistics*, volume 24, pages 483–7, 1992.

[11] L. B. Hearne and E. J. Wegman. Fast multidimensional density estimation based on random-width bins. In *Computing Science and Statistics*, volume 26, pages 150–5, 1994.

[12] A. Hinneburg and D. A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB'99, Edinburgh, Scotland, UK*, pages 506–517. Morgan Kaufmann, 1999.

[13] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *Proc. 2nd Int'l Conf. Knowledge Discovery & Data Mining*, pages 114–9, 1996.

[14] G. Lugosi and A. B. Nobel. Consistency of data-driven histogram methods for density estimation and classification. *Annals of Statistics*, 24:687–706, 1996.

[15] H. Nagesh, S. Goil, and A. Choudhary. Adaptive grids for clustering massive data sets. In V. Kumar and R. Grossman, editors, *First SIAM Int'l. Conf. Data Mining*, pages 506–517. SIAM, 2001.

[16] A. B. Nobel. Histogram regression estimation using data-dependent partitions. *Annals of Statistics*, 24(3):1084–1105, 1996.

[17] J. Rousu. *Efficient Range Partitioning in Classification Learning*. Ph.D. Dissertation, Department of Computer Science, University of Helsinki, Finland, January 2001.

[18] D. W. Scott. *Multivariate Density Estimation – Theory, Practice and Visualization*. John Wiley & Sons, 1992.

[19] D. W. Scott and G. Whittaker. Multivariate applications of the ASH in regression. *Communications in Statistics A: Theory and Methods*, 25:2521–30, 1996.

[20] W. Wang, J. Yang, and R. R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proc. 23rd VLDB Conf.*, pages 186–195, 1997.