Maximum a posteriori density estimation and the sparse grid combination technique

Matthias Wong[1] Markus Hegland[2]

(Received 31 October 2012; revised 19 July 2013)

Abstract

We study a novel method for maximum a posteriori (MAP) estimation of the probability density function of an arbitrary, independent and identically distributed $\mathbf{d}$-dimensional data set. We give an interpretation of the MAP algorithm in terms of regularised maximum likelihood. We also present numerical experiments using a sparse grid combination technique and the 'opticom' method. The numerical results demonstrate the viability of parallelisation for the combination technique.

Keywords: Data analysis

Contents

1 Introduction

The estimation of the probability density function (PDF) from a given data sample has always been important in statistical studies and applications [12]. In recent years, the maximum a posteriori (MAP) algorithm was developed for multi-dimensional probability density estimation [7]. The technique was further analysed from a theoretical point of view [4]. However, multi-dimensionality poses severe computational challenges. With these challenges in mind, the sparse grid combination technique was successfully employed in tackling multi-dimensional problems [3].

We study the MAP algorithm and explore the applicability of the combination technique to PDF estimation. In the following section we explore the connection between regularised maximum likelihood and the MAP algorithm. After

a short introduction to the sparse grid combination technique, we present our numerical results. Finally, we draw conclusions and discuss potential research directions.

We assume familiarity with the exponential family of distributions, including its *natural parameters* $\mathbf{c}$, *sufficient statistics* ϕ and *log partition term* $A(\mathbf{c})$ [10]. We make use of the two identities

$$\left.\frac{\partial A}{\partial c_j}\right|_{\mathbf{c}} = \mathbb{E}_{\mathbf{c}}\{\phi_j\},$$
$$\left.\frac{\partial^2 A}{\partial c_i \partial c_j}\right|_{\mathbf{c}} = \operatorname{cov}_{\mathbf{c}}\{\phi_i, \phi_j\}.$$

where c_j is the jth natural parameter and ϕ_j is the corresponding sufficient statistic. The maps $\mathbb{E}_{\mathbf{c}}$ and $\operatorname{cov}_{\mathbf{c}}$ are the expectation and covariance operators, respectively, corresponding to the PDF parameterised by $\mathbf{c}$.

2 The MAP algorithm for density estimation

Let $\Gamma := [0,1]^d$ and $\mathcal{C}(\Gamma)$ be the space of continuous real valued functions defined on Γ. We assume the data points $t_i \in \Gamma$ for $i = 1, \ldots, n$, are independent and identically distributed and have an underlying PDF

$$f(t \mid u) := \frac{e^{u(t)}}{\int_\Gamma e^{u(t)}\, d\mu(t)},$$

where $u \in H \subset \mathcal{C}(\Gamma)$ and H is a reproducing kernel Hilbert space continuously embedded in $\mathcal{C}(\Gamma)$. In others words, f resembles a member of the exponential family where the exponent is generalised to include a large class of functions. The log of the denominator in the above equation is

$$A(u) := \log \int_\Gamma e^{u(t)}\, d\mu(t),$$

which is analogous to the log partition term of the exponential family. Unlike the exponential family we do not assume the exponent to be a finite linear combination of some known sufficient statistics. However, if we choose a countable basis set for H, then the basis set is a *countable* number of sufficient statistics. This allows flexibility in PDF estimation to deal with situations where the data do not seem, a priori, to fit any known distribution.

The PDF estimation problem is reduced to minimising

$$j(u) := \frac{1}{2}\|u\|_H^2 + \log \int_T e^u \, d\mu - \frac{1}{n}\sum_{k=1}^{n} u(t_k) \,,$$

for functional $j : H \to \mathbb{R}$ [4]. The H-norm is implicitly parameterised by the number of data n and two statistical parameters α and β. In the functional $j(u)$, $\|u\|_H^2$ serves as a penalty term for regularity and well-posedness. The two remaining terms are analogous to the negative log likelihood functional used in maximum likelihood methods for the exponential family.

We can use a variational Newton–Galerkin algorithm to solve the above infinite dimensional minimisation problem [4]. However, in practice, one solves a finite dimensional problem with fixed basis $\phi_1, \ldots, \phi_m$. We write $u := \sum_{i=1}^{m} c_i \phi_i$ for some coefficients $c_1, \ldots, c_m$. Explicitly,

$$f(t \mid \mathbf{c}) := \exp\left(\sum_{i=1}^{m} c_i \phi_i(t) - A(\mathbf{c})\right).$$

This has the same form as a member of the exponential family. In this form, the MAP PDF estimation problem suggested by Griebel and Hegland [4] is equivalent to solving a maximum likelihood problem with a regularising term $\|u\|_H^2$. We interpret the basis functions ϕ_i as the sufficient statistics. This is a statistical interpretation of the algorithm.

Since we are minimising over $\mathbb{R}^m$, we apply standard minimisation techniques. We find that using Newton's method recovers the same algorithm as the MAP algorithm.

2.1 Solving the finite dimensional problem

Consider a finite dimensional subspace $V_h \subset H$ with basis functions $\phi_1, \dots, \phi_m$. Finding $\arg\min j(v)$ on V_h is rephrased as finding $c_1, \dots, c_m$ such that $j\left(\sum_{i=1}^m c_i \phi_i\right)$ is minimal. The minimum $u_h = \arg\min_{v \in V_h} j(v)$ is uniquely characterised by a zero derivative on V_h [4].

Proposition 1. *If $\phi_1, \dots, \phi_m$ are basis functions of V_h and if $u_n^c := \sum_{i=1}^m c_i \phi_i$, then the coefficients $c_1, \dots, c_m$ which minimise*

$$j(u_n^c) := \frac{1}{2}\|u_n^c\|_H^2 + \log \int e^{u_n^c} \, d\mu - \frac{1}{n} \sum_{j=1}^n u_n^c(t_j)$$

are the solutions of

$$\sum_{j=1}^m c_j (\phi_j, \phi_i)_H + \mathbb{E}_c\{\phi_i\} - \frac{1}{n} \sum_{j=1}^n \phi_i(t_j) = 0 \quad \text{for all } i = 1, \dots, m. \tag{1}$$

Proof: We define $J : \mathbb{R}^m \to \mathbb{R}$ by

$$J(c_1, \dots, c_m) := j\left(\sum_{i=1}^m c_i \phi_i\right),$$

and differentiate J with respect to c_i to obtain $\nabla J(\mathbf{c})$. We then set $\nabla J(\mathbf{c}) = 0$ for equation (1). The derivative is straightforward except for the log partition term. For this we use the identity

$$\left.\frac{\partial A}{\partial c_i}\right|_{\mathbf{c}} = \mathbb{E}_c\{\phi_i\}.$$

♠

To find the roots of equation (1) we apply Newton's method. In applying Newton's method we rely on the following proposition.

Proposition 2. *Fix* $\phi_1, \dots, \phi_m$ *and* $\mathbf{c}$. *Denote* $u_n^c = \sum_{i=1}^m c_i \phi_i$. *The Jacobian matrix of the* m*-vector function* $F : \mathbb{R}^m \to \mathbb{R}^m$ *with entries*

$$F_i(\mathbf{c}) := (u_n^c, \phi_i)_H + \mathbb{E}_{\mathbf{c}}\{\phi_i\} - \frac{1}{n}\sum_{k=1}^{n} \phi_i(t_k) \quad i = 1, \dots,$$

is $DF(\mathbf{c})$ *with elements*

$$DF_{i,j}(\mathbf{c}) := (\phi_i, \phi_j)_H + \mathrm{cov}_{\mathbf{c}}\{\phi_i, \phi_j\}.$$

Proof: The derivative is straightforward. For the second term we apply the identity

$$\left.\frac{\partial^2 A}{\partial c_i \partial c_j}\right|_{\mathbf{c}} = \frac{\partial}{\partial c_j}\mathbb{E}_{\mathbf{c}}\{\phi_i\} = \mathrm{cov}_{\mathbf{c}}\{\phi_i, \phi_j\}.$$

The third term vanishes since it does not depend on c_j. This yields the Jacobian $DF(\mathbf{c})$. ♠

The algorithm starts with an initial vector $\mathbf{c}$, determines a Newton step $\Delta\mathbf{c}$ using

$$DF(\mathbf{c})\Delta\mathbf{c} = -F(\mathbf{c}),$$

and updates $\mathbf{c}$ with

$$\mathbf{c}_{\text{update}} := \mathbf{c} + \lambda_{\mathbf{c}}\Delta\mathbf{c},$$

for some $\lambda_{\mathbf{c}}$. The scaling $\lambda_{\mathbf{c}}$ is the step size control which ensures convergence. It is computed using the Armijo criterion described by Kelley [9].

The above discussion yields the algorithm needed to find $u = \arg\min J(v)$ in the space V_h. Using

$$f(t \mid u) = \frac{e^u}{\int e^u \, d\mu},$$
$$\mathrm{cov}\{v, w\} = \mathbb{E}\{vw\} - \mathbb{E}\{v\}\mathbb{E}\{w\},$$

it is readily seen that our equations are identical to those derived through abstract analysis by Griebel and Hegland [4]. Our results give an alternative interpretation of their algorithm—it is simply solving a regularised maximum likelihood problem using Newton's method.

3 Numerical experience with the combination technique

In this section we are concerned with the ability of the sparse grid combination technique to improve computational performance of the MAP algorithm, compared to a direct full grid discretisation. The sparse grid discretisation and its combination technique were introduced to deal with multi-dimensional problems [13, 5]. After a brief overview of the technique, we discuss our experience. More detailed discussions are available elsewhere [3, 5].

3.1 The sparse grid combination technique

Let $(l_1, \ldots, l_d)$ denote the resolution levels of a uniform grid so, for example, $(2, 4)$ denotes a $2^2 \times 2^4$ grid with $(2^2 + 1) \times (2^4 + 1)$ nodes. In solving a numerical problem, we can only approximate the true (infinite dimensional) solution $u \in H$ using finite dimensional discretisation. The finer one discretises, the more satisfactorily one *may* be able to approximate the true solution. In this context one would prefer, say, the solution on a grid defined by resolutions $(6, 6)$ rather than $(5, 3)$. This reasoning breaks down in high dimensional situations. Computing the solution on the grid $(8, 8, 8, 8, 8)$ may no longer be as desirable as on, say, $(2, 3, 3, 2, 1)$, because the computational resources needed to handle the finer grid becomes exorbitant. As a rule, we are limited to solving problems on grids such as $(2, 3, 3, 2, 1)$ and other low resolution grids such as $(2, 6, 1, 1, 1)$.

Can we construct a better approximation through a linear combination of lower resolution solutions? That is, for a desired resolution level n in all dimensions, can we find lower resolution solutions $u_1, \ldots, u_m$ and combine them to obtain a new solution $u_n^c := \sum_{i=1}^{m} c_i u_i$?

For example, the user specifies a desired level, say $n = 3$, and the combination technique defines a set of component grids and their corresponding coefficients. For $n = 3$ in two dimensions, the combination technique specifies levels $(3,1)$, $(2,2)$, $(1,3)$, $(2,1)$ and $(1,2)$ with coefficients $1, 1, 1, -1, -1$ (see Figure 1). We solve the problem on these grids, obtain solutions $u_1, \ldots, u_5$, and combine them for a better solution

$$u_3^c := u_1 + u_2 + u_3 - u_4 - u_5 = \sum_{i=1}^{5} c_i u_i \,.$$

The combination scheme, illustrated by Figure 1, approximates the more expensive full grid solution $(3,3)$, indicated by F in the figure. This is the classical combination technique.

If $u_{i,j}$ is the solution on the grid (i,j), then the formula for the two dimensional combined solution with level n is

$$u_n^c := \sum_{i+j=n+1} u_{i,j} - \sum_{i+j=n} u_{i,j} \,. \tag{2}$$

The grid scheme and the coefficients are defined by the inclusion/exclusion principle [6]. Diagrams illustrating this concept are presented elsewhere [5, 3].

The degrees of freedom of the d-dimensional combined solution are $\mathcal{O}(n^{d-1}2^n)$, compared to $\mathcal{O}(2^{nd})$ for the full grid solution. The component solutions are calculated independently. Therefore, the combination technique allows a cheap and fully parallel way of approximating the full grid solution using component solutions. Although error analysis was derived for some problems [11, 1, 5], the effectiveness of the technique is not yet fully understood—it often works but there are times when it does not.

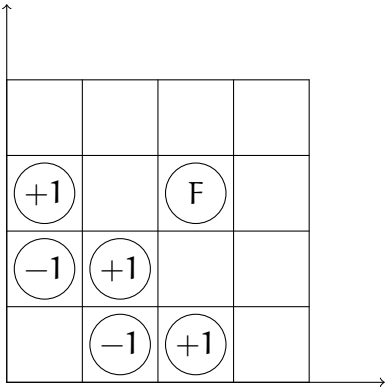

Figure 1: A combination technique illustrated in the grid of grids—the ijth cell corresponds to a grid with discretisation (i, j).

3.2 Experimental results

We are interested in the effectiveness of the combination technique in approximating a full grid solution when applied to our PDF estimation algorithm. For this reason we measure the relative error using

$$e(u_n^c) := \frac{j(u_n^c) - j(u_{n,n})}{j(u_{n,n})},$$

where $u_{n,n}$ denotes a solution solved on the full grid (n, n). This measures how much the combined solution differs from the full grid solution.

For our experiments, we used a desktop computer with two i5 Intel Cores at 2.5 GHz, coupled with 8 GB of RAM. For the combination technique we exploit the inherent parallelisms and use all four threads.

Table 1: Two dimensional bimodal dataset with the combination technique

	functional $\mathfrak{j}(\mathfrak{u})$			% error $\mathfrak{e}(\mathfrak{u})$	
level	full	component	combisol	component	combisol
2	−1.3012	−1.0601	−1.2936	19	0.58
3	−1.3836	−1.3012	−1.3749	6.0	0.62
4	−1.3941	−1.3434	−1.3850	3.6	0.65
5	−1.4026	−1.3836	−1.3953	1.4	0.51
6	−1.4132	−1.3888	−1.4057	1.7	0.52

3.2.1 Two dimensional Gaussian dataset

The first dataset is a two dimensional bimodal dataset. Five thousand points are sampled from two interposed Gaussians with different means. The 'component' column, in Table 1 under 'functional $\mathfrak{j}(\mathfrak{u})$', contains the best component solution calculated for the combination technique. The 'combisol' column contains the combination technique solution, that is, the solution obtained by combining the component solutions.

From Table 1, we see the combination of the component solutions is between three and thirty times better than even the best component solution. Moreover, the combined solution approximates the full grid solution very well. From Table 2 we see the combination technique led to a huge saving in computational time which grew exponentially with level. The results speak strongly in favour of the combination technique. In the next subsection we push the technique even further.

3.2.2 Three dimensional correlated dataset

In the three dimensional case, we sample fifty thousand points from a three dimensional, highly correlated dataset. The results from the combination technique are presented in Table 3. In this case, the combination technique is

Table 2: Two dimensional bimodal dataset with the combination technique

	time (milliseconds)	
level	full	combisol
2	0.92	0.68
3	5.2	1.7
4	32	5.3
5	320	17
6	16000	60

Table 3: Three dimensional dataset using the combination technique

	functional $j(u)$			% error $e(u)$	
level	full	component	combisol	component	combisol
2	−3.1282	−2.7001	−2.7253	14	13
3	−3.9000	−2.9208	−2.9048	25	26
4	−4.1674	−3.1318	−3.4801	25	16
5	−4.2391	−3.4038	−3.6490	20	13

not so successful, especially for the level three case—combining the solutions actually *reduced* the accuracy.

3.3 The opticom method

The 'opticom' method was proposed to improve the combination technique by choosing optimal coefficients depending on the problem [8]. It was used effectively where the combination technique failed [2].

In our context, we choose a combination of component solutions which minimises $j(u)$. In other words, after finding component solutions $u_1, \dots, u_m$, we find $\arg\min_{v \in V_c} j(v)$ where

$$V_c := \operatorname{span}\{u_1, \dots, u_m\}.$$

Table 4: Three dimensional dataset using opticom

	functional $j(u)$			% error $e(u)$	
level	full	component	opticom	component	opticom
2	−3.1282	−2.7253	−2.8991	13	7.3
3	−3.9000	−2.9048	−3.2974	26	5.5
4	−4.1674	−3.4801	−3.6411	16	13
5	−4.2391	−3.6490	−3.8898	13	8.2

This problem is exactly the same as the finite dimensional minimisation problem already discussed, only with the basis ϕ_i replaced by u_i. The implementation of this problem therefore makes use of much of the existing code and framework. One uses an object-oriented approach, then the class structures are almost identical. Moreover, the new system to be solved is relatively small in size.

3.3.1 Three dimensional dataset with opticom

We now apply opticom to the three dimensional dataset. From Table 4, as predicted by the theory, opticom always leads to an improvement over the combined solution. Most significantly, it corrected the level three combined solution. This is consistent with the existing results for opticom [2]. Moreover, since we are choosing optimal coefficients to minimise $j(u)$, opticom is a way of getting the most out of the combination technique by performing a small calculation, after calculating all components, to find the best coefficients.

From Table 5 we see that opticom is initially slower than the full grid discretisation. It grows at a rate similar to the combination technique and for large levels the time taken for opticom is negligible compared to a full grid discretisation. The cost of opticom is mostly spent on the quadrature for e^u. Tackling the quadrature is still a subject of research.

Table 5: Three dimensional dataset using opticom

	time (milliseconds)		
level	full	combisol	opticom
2	26	9.1	300
3	720	46	1500
4	48000	160	3500
5	3200000	640	12000

4 Conclusion and outlook

We examined the MAP algorithm for density estimation. We provided a statistical interpretation of the algorithm in terms of regularised maximum likelihood for the exponential family. It is a very general method and makes few assumptions about the underlying distributions. The algorithm is at most linearly affected by the size of the dataset, making the algorithm feasible for the huge datasets in modern applications.

Our numerical results suggest the applicability of the combination technique to the MAP algorithm. The main advantage of the combination technique is the saving of computational resources and an added level of parallelism. We were able to further improve the combination technique by the opticom modification. Using opticom, we obtained the best combination of the component solutions. At present, the theoretical conditions for the effectiveness of the combination technique are not well understood. There is a need to investigate the exact conditions for the method's effectiveness.

We were limited to two and three dimensions because the quadrature of e^u becomes unmanageable in higher dimensions. Using the Clenshaw–Curtis rule, we found that a resolution level of 13 was necessary for a desirable accuracy. Addressing the quadrature is still a matter open for research. However, we are optimistic about applying the combination technique and opticom with the MAP algorithm for large scale data mining once the quadrature has been tackled.

References

[1] H. J. Bungartz, M. Griebel, D. Röschke and C. Zenger. Pointwise convergence of the combination technique for the Laplace equation. *East-West J. Numer. Math*, 2:21–45 (1994). http://zbmath.org/?q=an:00653220 C515

[2] J. Garcke. Regression with the optimised combination technique. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 321–328 (2006). doi:10.1145/1143844.1143885 C518, C519

[3] J. Garcke. Sparse grid tutorial. Technical report (2011). http://page.math.tu-berlin.de/~garcke/paper/sparseGridTutorial.pdf C509, C514, C515

[4] M. Griebel and M. Hegland. A finite element method for density estimation with Gaussian process priors. *SIAM J. Numer. Anal.*, 47:4759–4792 (2010). doi:10.1137/080736478 C509, C511, C512, C514

[5] M. Griebel, M. Schneider and C. Zenger. A combination technique for the solution of sparse grid problems. In *Iterative methods in linear algebra (Brussels, 1991)*, pages 263–281. North-Holland, Amsterdam (1992). C514, C515

[6] M. Hegland. Adaptive sparse grids. *ANZIAM J.*, 44:C335–C353 (2003). http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/685 C515

[7] M. Hegland. Approximate maximum a posteriori with Gaussian process priors. *Constr. Approx.*, 26:205–224 (2007). doi:10.1007/s00365-006-0661-4 C509

[8] M. Hegland, J. Garcke, and V. Challis. The combination technique and some generalisations. *Linear Algebra Appl.*, 420:249–275 (2007). doi:10.1016/j.laa.2006.07.014 C518

[9] C. T. Kelley. *Solving nonlinear equations with Newton's method.* Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2003). C513

[10] H. Kobayashi, B.L. Mark, and W. Turin. *Probability, Random Processes, and Statistical Analysis: Applications to Communications, Signal Processing, Queueing Theory and Mathematical Finance.* Cambridge University Press (2012). C510

[11] C. Pflaum and A. Zhou. Error analysis of the combination technique. *Numerische Mathematik*, 84:327–350 (1999). doi:10.1007/s002110050474 C515

[12] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons (2004). C509

[13] C. Zenger. Sparse grids. *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar*, 31 (1990). C514

Author addresses

1. **Matthias Wong**, Mathematical Sciences Institute, Australian National University, Canberra, Australia
mailto:matthias.wong@anu.edu.au

2. **Markus Hegland**, Mathematical Sciences Institute, Australian National University, Canberra, Australia
mailto:markus.hegland@anu.edu.au