Parallel geostatistics for sparse and dense datasets

Ben Ingram[1] and Dan Cornford[1]

Neural Computing Research Group, Aston University, Aston Street, Birmingham. B4 7ET. B.R.Ingram@aston.ac.uk

1 Abstract

Very large spatially-referenced datasets, for example, those derived from satellite-based sensors which sample across the globe or large monitoring networks of individual sensors, are becoming increasingly common and more widely available for use in environmental decision making. In large or dense sensor networks, huge quantities of data can be collected over small time periods. In many applications the generation of maps, or predictions at specific locations, from the data in (near) real-time is crucial. Geostatistical operations such as interpolation are vital in this map-generation process and in emergency situations, the resulting predictions need to be available almost instantly, so that decision makers can make informed decisions and define risk and evacuation zones. It is also helpful when analysing data in less time critical applications, for example when interacting directly with the data for exploratory analysis, that the algorithms are responsive within a reasonable time frame.

Performing geostatistical analysis on such large spatial datasets can present a number of problems, particularly in the case where maximum likelihood. Although the storage requirements only scale linearly with the number of observations in the dataset, the computational complexity in terms of memory and speed, scale quadratically and cubically respectively. Most modern commodity hardware has at least 2 processor cores if not more. Other mechanisms for allowing parallel computation such as Grid based systems are also becoming increasingly commonly available. However, currently there seems to be little interest in exploiting this extra processing power within the context of geostatistics.

In this paper we review the existing parallel approaches for geostatistics. By recognising that different natural parallelisms exist and can be exploited depending on whether the dataset is sparsely or densely sampled with respect to the range of variation, we introduce two contrasting novel implementations of parallel algorithms based on approximating the data likelihood extending

the methods of Vecchia [1988] and Tresp [2000]. Using parallel maximum likelihood variogram estimation and parallel prediction algorithms we show that computational time can be significantly reduced. We demonstrate this with both sparsely sampled data and densely sampled data on a variety of architectures ranging from the common dual core processor, found in many modern desktop computers, to large multi-node super computers. To highlight the strengths and weaknesses of the different methods we employ synthetic data sets and go on to show how the methods allow maximum likelihood based inference on the exhaustive Walker Lake data set.

2 Introduction

The problem of large datasets was once considered a solved issue Schabenberger and Gotway [2005]. By using method–of–moments variograms and moving window kriging, all but the very massive dataset are computationally tractable. In recent years the popularity of and interest in maximum likelihood based algorithms has grown. Problems are encountered computationally with likelihood based methods when more than a few thousand observations are encountered.

Parallel geostatistics is not a new topic and has been considered previously by Pedelty et al. [2003], Gebhardt [2003] and Kerry and Hawick [1998]. The basis of these existing techniques is to perform moving window kriging by assigning a prediction area to each processor and predict at the locations for a given area. The authors neglect to discuss parameter estimation in a parallel context. Practically, computing the variogram using a method–of–moments estimator provides few challenges when compared to the computational complexity of prediction since computing the variogram is a $O(n^2)$ process.

The motivation for this study lies in a shift in computer microprocessor design, where uniprocessor microprocessors are being replaced by multi–processor or multi–core architectures. Although this new design does not always result in a speed–up for many geostatistical algorithms. Software typically needs to be written to utilise such architectures. If the software is built upon existing libraries such as BLAS[1], LAPACK[2] and ATLAS[3], then these libraries can be replaced with parallel equivalents. The current version of LAPACK comes with configuration options to create multi–threaded versions where the number of threads can be specified. One warning however is that some users have noted decreased computational speeds due to synchronisation and communication between different threads.

In this paper we discuss data parallelism approaches for performing geostatistics. In contrast to task parallelism, data parallelism relies on splitting

[1] `http://www.netlib.org/blas/`
[2] `http://www.netlib.org/lapack/`
[3] `http://math-atlas.sourceforge.net/`

the data into a number of smaller clusters and performing calculations across a number of processors.

We utilise the Message Passing Interface (MPI) for intra–node communication since this is the *de facto* standard for parallel programming. Implementations of MPI can be found for most architectures from the largest massively parallel super–computers to a standard desktop with a dual–core processor. During the development of this software we used the Matlab compatible application Octave[4] and MPITB [Fernández et al., 2004] which is a MPI implementation compatiable with Matlab and Octave. Octave does not have the licensing restrictions that Matlab has so is ideal for using on a multiple processor machine.

3 Methodologies

We discuss and implement two methods in this paper. The first method we consider is that of Vecchia [1988] which can be used to approximate the likelihood function. The second method, the Bayesian Committee Machine (BCM) is used for prediction using all of the data. We compare the results with traditional geostatistics that such as method–of–moments variograms and moving window kriging.

3.1 Vecchia's approximation

Vecchia [1988] approximation is based on the multiplicative theorem which states for any number of N events: $z_1, \ldots, z_N$ the following relationship holds:

$$p\left(z_1 \cap z_2 \cap \ldots \cap z_N\right) = p\left(z_1\right) \cdot p\left(z_2|z_1\right) \cdot \ldots \cdot p\left(z_N|z_1, z_2, \ldots, z_{N-1}\right) \quad (1)$$

where $p\left(z_a|z_b\right)$ is the conditional probability of z_a given z_b Pardo-Igúzquiza and Dowd [1997].

In the case of a multivariate probability density function, the following relationship is obtained:

$$p\left(Z\left(\mathbf{x}\right)\right) = \prod_{i=1}^{N} p\left(Z\left(\mathbf{x}_i\right)|Z\left(\mathbf{x}_1\right), \ldots, Z\left(\mathbf{x}_{i-1}\right)\right) \quad (2)$$

One then assumes that some of the information in the dataset is redundant and hence instead of conditioning on the whole dataset the observations are conditioned on smaller subsets of size $m < (i-1)$ where i is the current observation of the dataset. This gives the following relationship:

$$p\left(Z\left(\mathbf{x}_i\right)|Z\left(\mathbf{x}_1\right), \ldots, Z\left(\mathbf{x}_{i-1}\right)\right) \cong p\left(Z\left(\mathbf{x}_i\right)|Z\left(\mathbf{x}_1\right), \ldots, Z\left(\mathbf{x}_m\right)\right) \quad (3)$$

[4] `http://www.octave.org`

where the approximation becomes almost exact as m approaches the number of observations in the dataset.

Assuming that data is a zero mean multivariate Gaussian, the conditional probability $p\left(Z\left(\mathbf{x}_i\right)|Z\left(\mathbf{x}_j\right)\right)$ where $j = 1, \ldots, m$ is also Gaussian for any observation i and any conditioning subset size m and is given by

$$\mathcal{N}\left(\Sigma_{ij}\Sigma_{jj}^{-1}Z\left(\mathbf{x}_j\right), \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji}\right) \tag{4}$$

where Σ_{ij} refers to the covariance between the observation i and the observations in the conditioning subset j. The following give the mean:

$$\Sigma_{i|j} = \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji} \tag{5}$$

and covariance:

$$\mu_{i|j} = \Sigma_{ij}\Sigma_{jj}^{-1}Z\left(\mathbf{x}_j\right) \tag{6}$$

conditioned on a subset of j observations where Σ_{jj} is a $j \times j$ covariance matrix between the points of vector y_j, Σ_{ij} is a vector of covariances between the ith observation and m points of the vector y_j and y_j are m observations at locations chosen for each subset.

This leads to the following log likelihood approximation:

$$\mathcal{L}\left(\theta\right) = -\frac{N}{2}\log\left(2\pi\right) - \frac{1}{2}\sum_{i=1}^{n}\log\left|\mathbf{\Sigma}\left(\boldsymbol{\theta}\right)_{i|j}\right| - \frac{1}{2}\sum_{i=1}^{n}Z\left(\mathbf{x}\right)^{T}\mathbf{\Sigma}\left(\boldsymbol{\theta}\right)_{i|j}^{-1}Z\left(\mathbf{x}\right) \tag{7}$$

which instead of depending the inverse of a covariance matrix $\mathbf{\Sigma}\left(\boldsymbol{\theta}\right)^{-1}$ of size N, depends on i covariance matrices of maximum size m. Hence the smaller size m the more computationally efficient the algorithm is but at the expensive of yielding a poorer approximation to the *true* probability density function.

Although Vecchia [1988] notes that the orderings of the data makes a difference to the approximation, this is not considered a significant issue and it is not dealt with. A number of years after this approximation method was proposed, Stein et al. [2004] suggested a number of improvements to the algorithm. Firstly it is suggested that the approximation gives better results when the observations are ordered so as to give clustered data. Secondly, by not only conditioning on observations near, but also on some observations far away, the approximation is further improved.

Since the approximate maximum likelihood approach has reduced the calculation to a sum of a number of independent calculations, a parallel implementation follows trivially. A further desirable feature is that all the data need not be sent to each process in the parallel system. How much data sent to each process depends on m, the size of the conditioning data. Particularly accurate approximations to the likelihood can be achieved with large m.

One serial implementation of the approach of Vecchia [1988] was presented by Pardo-Igúzquiza and Dowd [1997]. Since the computation of the conditioning subsets is an *embarrassingly parallel* problem, it can be easily parallelised. Figure 1 shows a simple pseudo code parallel algorithm.

```
1. Master to broadcast covariance parameters to each process
   (MPI_Bcast)
2. Master to scatter training data to each process
   (MPI_Scatter)
3. Each node to calculate likelihood of each supplied
   observations conditioned on a subset of the data
4. Master to collect log likelihoods and sum (MPI_Reduce)
```

Fig. 1. Pseudo code for Vecchia approximation

3.2 Bayesian Committee Machine

The Bayesian Committee Machine was proposed by Tresp [2000] as an alternative method for reducing the computational complexity of prediction and has been frequently applied in the context of machine learning. Using this model, the data is split up into submodels or committees and weighted by the inverse variance or precision at the prediction location. The BCM has an equivalence to kriging with a number of additional assumptions.

One important feature to note about the BCM is that it is a *transductive* method rather than an *inductive* method. The term *transductive* means that the method computes a model dependent on a user specified set of prediction locations [Schwaighofer and Tresp, 2003]. In this way, knowledge about the covariance between the prediction locations is exploited in the approximation.

It has been shown by Schwaighofer and Tresp [2003] that the BCM method is equivalent to assuming a low–rank covariance matrix where the exact block diagonal structure of the full covariance is retained. As with many low–rank matrix approximations or their equivalents the concept of knots, pseudo inputs or active points are used. For example, assuming a dataset of observations, $(x_i, y_i) : i = 1, \ldots, n$, where a subset of the locations $(x_j) : j = 1, \ldots, m$ are selected and termed the active set. The low–rank covariance, $\hat{\Sigma}$ or approximation to the full covariance matrix Σ is given by:

$$\hat{\Sigma} = c(d)C^{-1}c(d)' \tag{8}$$

where $c(\cdot)$ is the approximate covariance function and C is the covariance matrix between the locations selected for inclusion in the active set.

The BCM assumes that the prediction locations compose the active set which we will denote as Σ_{pred}. The apparent limitation of having to compute the covariance matrix (and the inverse) of the prediction locations is not too restrictive since smaller prediction covariance matrices can be created and the BCM equations can be repeatedly calculated without a growth of the algorithm complexity.

The predictive distribution equations are calculated as:

$$\hat{Z}_{bcm} = \mathbf{\Sigma}_{bcm} \sum_{w=1}^{W} \tilde{\Sigma}_w^{-1} \hat{Z}_w \tag{9}$$

where $\mathbf{\Sigma}_{bcm}$ is the predictive covariance is given by:

$$\mathbf{\Sigma}_{bcm}^{-1} = -(W-1)\,\Sigma_{pred}^{-1} \sum_{c=1}^{W} \tilde{\Sigma}_{w}^{-1} \tag{10}$$

where W is the number of committees used and Σ_{pred} is covariance matrix between the prediction locations [Tresp, 2001]. An interesting observation is that the BCM predictive mean is constructed from a weighted sum of the individual committee members predictive means:

$$\hat{Z}_w = c(d)_w^T \mathbf{\Sigma}_w^{-1} \mathbf{Z}_w \tag{11}$$

where the matrix $\mathbf{\Sigma}_w$ is the covariance matrix of the observations assigned to committee w. The covariance between the prediction locations and the locations assigned to the committee is denoted by $c(d)_w$. The prediction locations are conditioned on the observed data assigned to a committee w by:

$$\tilde{\Sigma} = \mathbf{\Sigma}_{pred} - c(d)_w^T \mathbf{\Sigma}_w^{-1} c(d)_w. \tag{12}$$

Another observation is that the weights are obtained by the inverse predictive covariance (or predictive precision) at the prediction location. Effectively the BCM scales the contribution of each committee based on how confident it is about the prediction from each committee. Substituting the individual committee members predictive means and variances gives full expressions for the full predictive mean:

$$\hat{Z}_{bcm} = \mathbf{\Sigma}_{bcm} \sum_{w=1}^{W} \left(\mathbf{\Sigma}_{pred} - c(d)_w^T \mathbf{\Sigma}_w^{-1} c(d)_w \right)^{-1} c(d)_w^T \mathbf{\Sigma}_w^{-1} \mathbf{Z}_w \tag{13}$$

and predictive variance:

$$\mathbf{\Sigma}_{bcm} = \left(-(W-1)\,\Sigma_{pred}^{-1} \sum_{w=1}^{W} \left(\mathbf{\Sigma}_{pred} - c(d)_w^T \mathbf{\Sigma}_w^{-1} c(d)_w \right)^{-1} \right)^{-1}. \tag{14}$$

Equations 13 and 14 indicate that there are a number of matrix inversions needed for this calculation. Some of these matrix inversions can be performed independently of other calculations and hence in parallel. The iterations in the sum calculation are completely independent of each other. By assigning these iterations to other processors in a parallel system it is proposed that speed–ups can be achieved since the main bottleneck in this algorithm (and many other geostatistical algorithms) is the matrix inversion.

For the BCM parallel implementation, the individual committee predictive mean and predictive variance will be performed on separate processors. The calculations of the predictive mean and predictive variance require the inverse of a matrix of the same size as the number of observed data assigned to each committee. A further inversion is needed to calculate the inverse of the predictive variance which is a matrix of the same size as the number of prediction locations. The basic algorithm for a parallel BCM is given in Figure 2.

```
1. Master to broadcast committee parameters to each process
   (MPI_Bcast)
2. Master to broadcast test data locations to each process
   (MPI_Bcast)
3. Master to scatter training data to each process
   (MPI_Scatter)
4. Each node to calculate the contribution of assigned
   committee
5. Master to collect the mean and variance at the test
   locations from each process and sum results (MPI_Reduce)
```

Fig. 2. Pseudo code for parallel Bayesian Committee Machine.

3.3 Moving window kriging

One approach to performing kriging with large datasets was introduced by David [1976]. A specified search radius from the prediction location is used to select a local neighbourhood of observations to use in the kriging system. This neighbourhood moves according to the location which is being predicted. An alternative approach for selecting the neighbourhood is to select a predetermined number of near observations for each prediction location. As noted by Davis and Culbane [1984], these methods produce spurious behaviour in some of the estimates and hence should be used with caution, this is apparent as observations are added or removed from the moving window. Ad–hoc methods of subsetting the data were formalised by the moving-window approach of Haas [1995], although the local covariance functions fitted within the window may yield incompatible covariances at larger spatial lags. Cressie [1993] states that for datasets that are large, the general feeling is that kriging is impossible and ad–hoc local kriging neighbourhoods are typically used. Isaaks and Srivastava [1989] devote a whole chapter to choosing an effective search strategy. Implementations of kriging tend to use this approach for performing kriging efficiently. Here we use the moving window kriging approach as a means of benchmarking the BCM.

4 Experimental setup

4.1 Datasets

To test these methods we simulate two large spatial datasets, each with 40,000 observations on a grid of 200 $\times$ 200 points. To do this we use the Turning Bands method of simulation [Emery and Lantuéjoul, 2006] since large datasets can be simulated without prohibitive running times. Figure 3 shows the two simulated fields. Both datasets were simulated with an exponential covariance function. The first dataset was simulated with an effective range of 15m

and the second dataset has an effective range of 150m. We sample 20,000 observations from each dataset using simple random sampling. We use this for learning the model parameters and prediction. We use the remaining 20,000 observations for cross–validation to test our model.

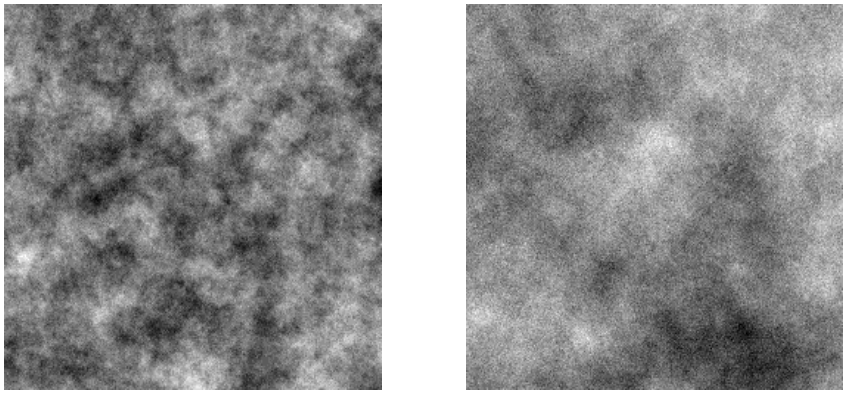

Fig. 3. Plots of simulated datasets with exponential covariance and (left) short range parameter (right) long range parameter.

4.2 Software

For the experiments in this paper we used an 8 node tightly coupled parallel system where we compared the performance using 1, 2, 4 and 8 processors. We choose LAM/MPI[5] implementation of the MPI standard.[6] ATLAS was compiled with the thread support disabled. The software was written using the Matlab language and executed in Octave although we hope to release a standalone implementation soon.

5 Results

The results from the experiments are split into two tables. Table 1 shows the Mean Average Error (MAE) of using the Vecchia method to estimate the covariance parameters and then we predict at the cross–validation locations. The first thing to notice is how the prediction results do not change depending on the number of processors used which is to be expected. As the number of

[5] http://www.lam-mpi.org/
[6] http://www-unix.mcs.anl.gov/mpi/

observations used in the conditioning subset, the MAE decreases with both datasets. The effect seems less pronounced with Dataset 2 however. The timings are taken after 10 iterations of the conjugate gradient minimisation of the approximate likelihood. For the BCM algorithm we compare this directly

Table 1. Performance of parallel maximum likelihood using Vecchia's method

Processors	Subset Size	Time (s)	MAE Dataset 1	MAE Dataset 2
1	50	50.84	34.44	36.23
2	50	30.75	34.44	36.23
4	50	20.74	34.44	36.23
8	50	16.44	34.44	36.23
1	100	240.12	31.71	34.83
2	100	130.47	31.71	34.83
4	100	70.33	31.71	34.83
8	100	40.34	31.71	34.83
1	200	1290.38	30.43	34.01
2	200	640.69	30.43	34.01
4	200	340.91	30.43	34.01
8	200	180.34	30.43	34.01

to MWK (Moving Window Kriging). By increasing the size of each committee, it can be seen that the computational complexity increases. However, increasing the number of processors reduces the computational burden. As to be expected and as the previous results, the MAE does not change depending on the number of processors used. In this experiment, by increasing the committee size seems to reduce the MAE for Dataset 2 more markedly than with Dataset 1.

The results for MWK show that for Dataset 1, where the range parameter is short with respect to the overall scale of the dataset, that MWK out performs the BCM in terms of prediction accuracy, although the computational speed is significantly slower. This can reduced of course by applying MWK to a parallel processor computer. With Dataset 2 where the range parameter is long with respect to the overall scale of the data, the effect seems less severe. The BCM seems to perform equally as well as MWK in terms of prediction accuracy, however in terms of prediction speed, the BCM is many more times more efficient.

6 Conclusions

In this paper we have considered two methods for applying parallel geostatistics. Firstly we looked at approximating the likelihood using a well known technique in geostatistics [Stein et al., 2004]. We showed how this was particularly effective when the range parameter was short when compared with the

Table 2. Performance of parallel BCM and moving window kriging

Processors	Subset Size	Time (s)	MAE Dataset 1	MAE Dataset 2
1	250	1.84	27.92	22.15
2	250	2.05	27.92	22.15
4	250	1.92	27.92	22.15
8	250	2.14	27.92	22.15
MWK	250	224.23	22.10	22.19
1	500	3.67	27.83	21.62
2	500	3.27	27.83	21.62
4	500	2.84	27.83	21.62
8	500	2.34	27.83	21.62
MWK	500	573.74	21.95	21.22
1	1000	11.90	27.53	20.76
2	1000	6.75	27.53	20.76
4	1000	3.98	27.53	20.76
8	1000	2.85	27.53	20.76
MWK	1000	1473.34	22.01	20.90

overall scale of the area of interest. When applied to Dataset 2, with a long range parameter, the performance was poorer. Increasing the number of processors reduced prediction time. Using two processors does not exactly half the calculation time due to overheads of distributing the data to the other processor. In terms of the computational complexity of this algorithm, the distribution of data will cause a short delay (depending on the architecture of the system.

The second technique we looked as was the BCM. This was shown to be equivalent to a low–rank covariance matrix approximation with the exact diagonal structure of the true covariance matrix retained. Low–rank methods are particularly useful when the range parameter of the dataset is long when compared with the overall scale of the dataset. Hence it is to be expected that the BCM performs better on Dataset 2. The BCM provides an effective alternative to moving window kriging when large datasets are encountered. For the BCM experiments the covariance function parameters were determined *a–priori*. Another advantage of using the BCM method is that all the data in the dataset is used for prediction rather than a subset. We are aware of an unpublished work that provides an approximation to the BCM likelihood using a Laplace propagation technique. This will be implemented in future versions.

The methods presented here are effective when applied to a specific geostatistical problems. They enable principled geostatistics to be applied to large datasets.

Acknowledgements

This work is funded by the European Commission, under the Sixth Framework Programme, by the Contract N. 033811 with the DG INFSO, action Line IST-2005-2.5.12 ICT for Environmental Risk Management. The views expressed herein are those of the authors and are not necessarily those of the European Commission.

References

Noel A.C. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, New York, 1993.

M. David. The practice of kriging. *Advanced Geostatistics in the Mining Industry*, 31:461, 1976.

M. W. Davis and P. G. Culbane. Contouring very large data sets using kriging. *Geostatistics for Natural Resources Characterization*, 2:599–619, 1984.

X. Emery and C. Lantuéjoul. TBSIM: A computer program for conditional simulation of three-dimensional Gaussian random fields via the turning bands method. *Computers and Geosciences*, 32(10):1615–1628, 2006.

J. Fernández, M. Anguita, S. Mota, A. Cañas, E. Ortigosa, and F. J. Rojas. MPI Toolbox for Octave. In *Proceedings of 6th International Conference on High Performance Computing for Computational Science*, Valencia, Spain, 2004. Springer-Verlag Berlin Heidelberg.

A. Gebhardt. PVM kriging with R. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna*, 2003.

T. C. Haas. Local prediction of a spatio-temporal process with an application to wet sulfate deposition. *Journal of the American Statistical Association*, 90(432): 1189–199, 1995.

Edward H. Isaaks and R. Mohan Srivastava. *An introduction to applied geostatistics*. Oxford University Press, USA., 1989.

K. E. Kerry and K. A. Hawick. Kriging Interpolation on High-Performance Computers. In *Proceedings of the International Conference and Exhibition on High-Performance Computing and Networking*, pages 429–438. Springer-Verlag Berlin Heidelberg, 1998.

E. Pardo-Igúzquiza and P.A. Dowd. AMLE3D: A computer program for the inference of spatial covariance parameters by approximate maximum likelihood estimation. *Computers and Geosciences*, 23(7):793–805(13), 1997.

J. A. Pedelty, J. L. Schnase, and J. A. Smith. High Performance Geostatistical Modeling of Biospheric Resources in the Cerro Grande Wildfire Site, Los Alamos, New Mexico and Rocky Mountain National Park, Colorado. NASA Goddard Space Flight Center, Code 930, 2003.

O. Schabenberger and C.A. Gotway. *Statistical Methods For Spatial Data Analysis*. CRC Press, Boca Raton, FL., 2005.

A. Schwaighofer and V. Tresp. Transductive and inductive methods for approximate Gaussian process regression. *Advances in Neural Information Processing Systems*, 15:953–960, 2003.

M.L. Stein, Z. Chi, and L.J. Welty. Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2):275–296, 2004.

V. Tresp. A Bayesian Committee Machine. *Neural Computation*, 12(11):2719–2741, 2000.

V. Tresp. *Committee machines, in Handbook for Neural Network Signal Processing*, chapter 5, pages 1–18. CRC Press, Boca Raton, FL., 2001.

A. V. Vecchia. Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society. Series B. Methodological*, 50(2):297–312, 1988.