Spatial Operators for Evolving Dynamic Bayesian Networks from Spatio-temporal Data

Allan Tucker[1], Xiaohui Liu[1], and David Garway-Heath[2]

[1] Brunel Univeristy, Middlesex, UK
{allan.tucker,xiaohui.liu}@brunel.ac.uk
[2] Glaucoma Unit, Moorfield's Eye Hospital, London, UK
david.garway-heath@moorfields.nhs.uk

Abstract. Learning Bayesian networks from data has been studied extensively in the evolutionary algorithm communities [Larranaga96, Wong99]. We have previously explored extending some of these search methods to temporal Bayesian networks [Tucker01]. A characteristic of many datasets from medical to geographical data is the spatial arrangement of variables. In this paper we investigate a set of operators that have been designed to exploit the spatial nature of such data in order to learn dynamic Bayesian networks more efficiently. We test these operators on synthetic data generated from a Gaussian network where the architecture is based upon a Cartesian coordinate system, and real-world medical data taken from visual field tests of patients suffering from ocular hypertension.

1 Introduction

Bayesian Networks (BNs) are probabilistic models that can be used to combine expert knowledge and data. They also facilitate the discovery of complex relationships in large datasets. A BN consists of a *directed acyclic graph* consisting of links between nodes that represent variables in the domain. The links are directed from a *parent* node to a *child* node, and with each node there is an associated set of *conditional probability distributions*. Learning the structure of a BN from data [Cooper92] is a non-trivial problem due to the large number of candidate network structures and as a result there has been substantial research in developing efficient algorithms within the optimisation and evolutionary communities. For example, Evolutionary Programs (EP) [Bäck93] and Genetic Algorithms (GA) [Holland95] have been used to search for candidate structures [Larranaga96, Wong99]. The Dynamic Bayesian Network (DBN) is an extension of the BN that can model time series [Friedman98]. See figure 1 for an example of a DBN with N+1 variables spanning two *time slices*. Note that links in a DBN can be between nodes in the same time slice or from nodes in previous time slices. We have developed algorithms (evolutionary and non-evolutionary) for efficiently learning DBN structures [Tucker01].

Spatial Data-mining [Ester00] involves the application of specialist algorithms and operators (e.g. neighbourhood distance, topology and direction) to data that has a spatial nature. That is, variables in the domain can be considered to have neighbourhood relations with other variables. This can vary from simple Cartesian coordinates

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 2360–2371, 2003.

to more complex spatial neighbourhoods. See [Roddick99] for a bibliography on spatial and temporal data mining. Spatial data is common in geographical research [Cofiño02].

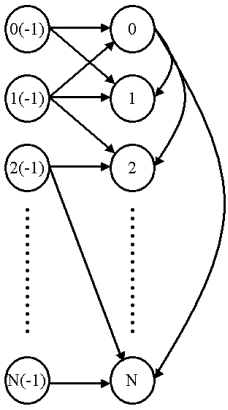

Fig. 1. A Typical DBN

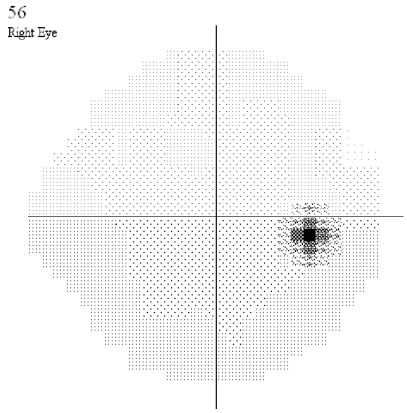

Fig. 2. A Visual Field Test

To our knowledge, BN learning has not been investigated with respect to spatial data. We define a Spatial Bayesian Network (SBN) to be a BN that represents data of a spatial nature and a Spatial Dynamic Bayesian Network (SDBN) to be a BN that represents spatio-temporal data. We introduce and test an evolutionary algorithm and a set of spatial operators for learning SDBNs from spatio-temporal data. These operators are tested on synthetic data and visual field data of patients suffering from ocular hypertension. The algorithm and its operators are compared with a standard BN search technique. We validate resulting networks in several ways including inspection of the spatial links, calculating the structural difference of the original network used to generate the synthetic data, and using clinical expert knowledge on the visual field data.

2 Methods

2.1 Datasets

We have generated spatio-temporal data using DBNs with Gaussian probability distributions [Geiger94]. This data contains 1000 time points of 64 variables spatially located on an 8x8 grid, where each variable was influenced by its immediate neighbours at the previous time point.

Visual Field (VF) data can be recorded using the Humphrey Perimeter [Haley87]. This involves a patient fixating a point in the centre of a dimly illuminated bowl. The perimeter shines brighter stimuli onto the bowl at various points, corresponding to points in the visual field. The stimuli are varied in intensity and the patient presses a

button when a stimulus has been observed. This technique determines the sensitivity to light of each point in the VF. The data that is used in this paper is from a study that contains tests recorded every few months, in patients with ocular hypertension, a major risk factor for glaucoma. The test measured 54 points on each eye (figure 2 shows an example of a patient's VF test). The dataset used in this paper involves 95 patients with 1809 measurements in all, concerning only the right eye. Two points in the VF correspond to the blind spot and should not contain any useful data. We have included these points to check for spurious relationships. We know of little research in using probabilistic models to understand VF data. Previously, a state space model has been used to classify glaucomatous patients [Anderssen98] and Bayesian statistics have been proposed to test VF data [Bengtsson97]. Both datasets were discretised into four states using equal bin sizes from the maximum value to the minimum value for each variable.

2.2 Algorithm

Candidate structures of a network, *bn*, given a dataset, *D*, are scored using the log-likelihood, calculated using the equation:

$$\log p(D \mid bn_D) = \log \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(F_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} F_{ijk}!,$$

where N is the number of variables in the domain, r_i denotes the number of states that a node x_i can take, q_i denotes the number of unique instantiations of the parents of node x_i, F_{ijk} is the number of cases in D, where x_i takes on its kth unique instantiation and, the parent set of i takes on its jth unique instantiation.

$$F_{ij} = \sum_{k=1}^{r_i} F_{ijk} \cdot$$

In order to increase efficiency of our algorithms, we have developed an evolutionary approach without the necessity of storing a population of candidate solutions. Rather, we consider each point in the spatial dataset to be an individual within the population of points. Therefore, the population, itself, is the candidate solution. We have looked at a similar method before for grouping algorithms [Liu01]. The algorithm also makes use of a simulated annealing type of selection criteria [Kirkpatrick83], where good operations are always carried forward, but sometimes less good ones are also accepted dependant upon a temperature parameter. A form of elitism [Grefenstette86] is employed to ensure that the final structure is the best discovered. This is to prevent the simulated annealing process from moving away from a better solution when the temperature is still high. We formally define the algorithm below where *maxfc* is the maximum number of calls to the scoring function, c is the 'cooling parameter', t_0 is the initial temperature, b is the branching factor of a network, and $R(a, b)$ is a uniform random number generator with limits, a and b.

```
Input    t0, b, maxfc, D
         fc = 0, t = t0
         Initialise bn to a SDBN with no links
         result = bn
         While fc ≤ maxfc do
            score = L(bn)
            For each operator do
               Apply operator to bn
               If bn is valid given b Then
                  newscore =  L(bn)
                  fc = fc + 1
                  dscore=newscore-oldscore
                  If newscore > score Then
                     result = bn
                  Else If R(0,1)< e^(dscore/t)  Then
                     Undo the operator
                  End If
               End If
            End For
            t = t × c
         End While
Output   result
```

2.3 Operators

We now introduce three spatial and three non-spatial operators. All involve manipulating links within the SDBN. For this paper we have only investigated links that span one time slice, although we have developed equivalent operators for links within the same time slice (these require a more strict ordering assumption to ensure the structure is directed acyclic which is an essential property of a BN structure).

2.4 Non-spatial Operators

We have chosen three non-spatial operators as these represent common operators used in optimisation techniques such as hill climbing and simulated annealing.

1) *Add* - A link with random parent and child is added to the network.
2) *Take*– Randomly remove a single existing link.
3) *Mutate* - Randomly change the parent of an existing link.

2.5 Spatial Operators

For the scope of this paper, we assume that the points in a spatial dataset are located according to Cartesian coordinates. Therefore, each point in a dataset with coordinates (x,y) has a first order *neighbourhood* which includes all nodes with coordinates (i,j) for $x\text{-}1 \le i \le x\text{+}1$ and $y\text{-}1 \le j \le y\text{+}1$. The spatial operators that we have developed exploit the Cartesian spatial nature of a dataset, whereby links are added to a node

based upon the proximity of the parent to the child, and mutations are made from an existing parent's position to the positions of other nearby parents. A crossover operator has also been developed that swaps the relative positions of node's parents. Figure 3 shows examples of parent coordinates, relative to the child node, when applying the operators. Unfilled circles represent child nodes, filled circles represent parents of the child.

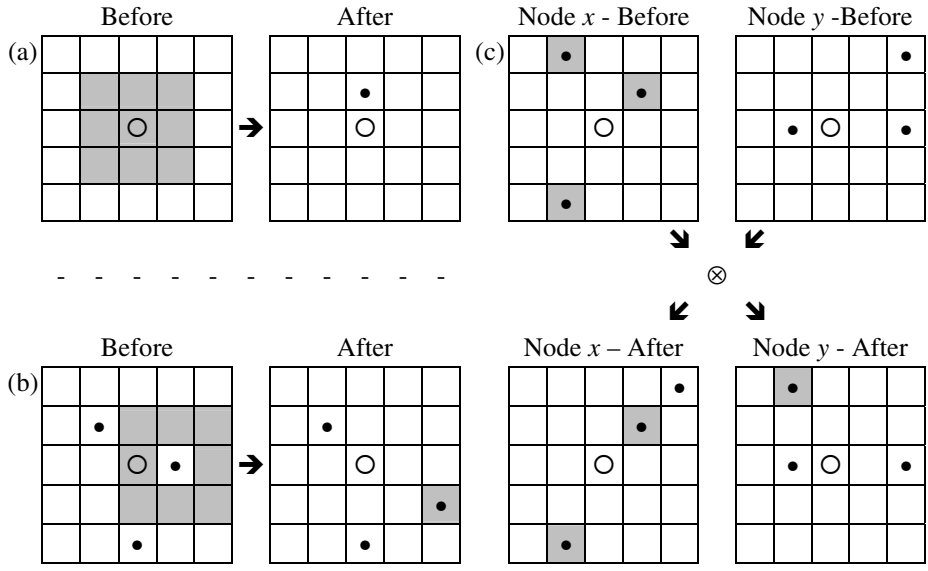

Fig. 3. Spatial Operator Examples: (a) Add (b) Mutate and (c) Crossover**.**

1) *Spatial Add* - Randomly add a link to a node, such that its parent is one of its first order neighbours (shaded in figure 3a). Note that the coordinates of a parent can be the same as its child because we are dealing with links that span a time-slice.

2) *Spatial Mutate* - Randomly replace the parent of an existing link with one of its first order neighbours (for example, the shaded region in figure 3b represents possible new positions for the parent immediately to the right of its child).

3) *Spatial Crossover* - Randomly select two nodes. For each parent of a node, x, there is a 50% chance of moving it to become a parent of the other node, y, so that its relative position to x becomes updated relative to y. See figure 3c where the shaded cells with filled circles show the parents of node x, before and after crossover, and the unshaded cells with filled circles show the parents of node y.

For all spatial operators, if a parent that is selected for a node is outside the boundary of the coordinate system, then it is deemed invalid.

3 Experiments

The experiments involved running the stochastic algorithm described earlier on the synthetic and VF datasets. One experiment involved using only non-spatial operators, one involved only spatial operators and one using all operators. All operators were applied in the order that they appear listed in the previous section. Due to the stochastic nature of the algorithm, we ran each experiment ten times and recorded the average learning curve. We also investigated Cooper's greedy search algorithm, K2, [Cooper92] when applied to the two datasets. To find out the performance of the individual operators, we also recorded the average success rate of each operator in the 'all operators' experiments. A success is recorded when an operator results in improved fitness.

In order to determine the quality of the final structures, we use the Structural Difference (SD) metric on the synthetic data. This is calculated from summing the missing links with the spurious links in the discovered structure, compared to the original network used to generate the data. Expert knowledge is used for the VF data, based upon the anatomical correspondence of VF locations to particular nerve fibre bundles and the proximity of the nerve fibre bundles to each other in terms of their position of entry (angular location) at the Optic Nerve (ON) margin [Garway00].

3.1 Parameters

In this experiments we set t_0=5, b=3, c=0.9999 and $maxfc$=35000. These values were chosen as they generated the most efficient results for all methods.

4 Results

4.1 Synthetic Data

Figure 4a shows the learning curves for each method when applied to the synthetic data. It can be seen that the simple K2 algorithm is not as efficient as any of the stochastic methods. What is more, the log likelihood of the final solution is not as high as the other methods. The spatial operators alone seem to perform the most efficiently. This could be due to the more limited type of relationships that exist. All are based upon the first order Cartesian neighbourhood. It should be noted that on the synthetic data that if all methods, including K2, are run for long enough they eventually reach solutions very close to one another (again perhaps due to the simplicity of the relationships within the network).

Figure 4b shows the mean success of individual operators during an experiment, figure 4b shows that the most successful were *Spatial Add* (SpatAdd), *Take*, and *Spatial Mutation* (SpatMut). The worst seems to be *Spatial Crossover* (SpatCross).

We have calculated the structural difference of the discovered structures from the original structures used to generate the data. Table 1 shows that on average the closest network to resemble the original structure that generated the data is surprisingly that from using K2. This is somewhat unexpected, but could be due to the simplistic na-

ture of the synthetic data (where all dependencies are generated from a linear Gaussian process given each variable's first order neighbours). The next best structure is that from the spatial operators alone. The next closest is found when using spatial and non-spatial operators, and the worst when using only non-spatial operators.

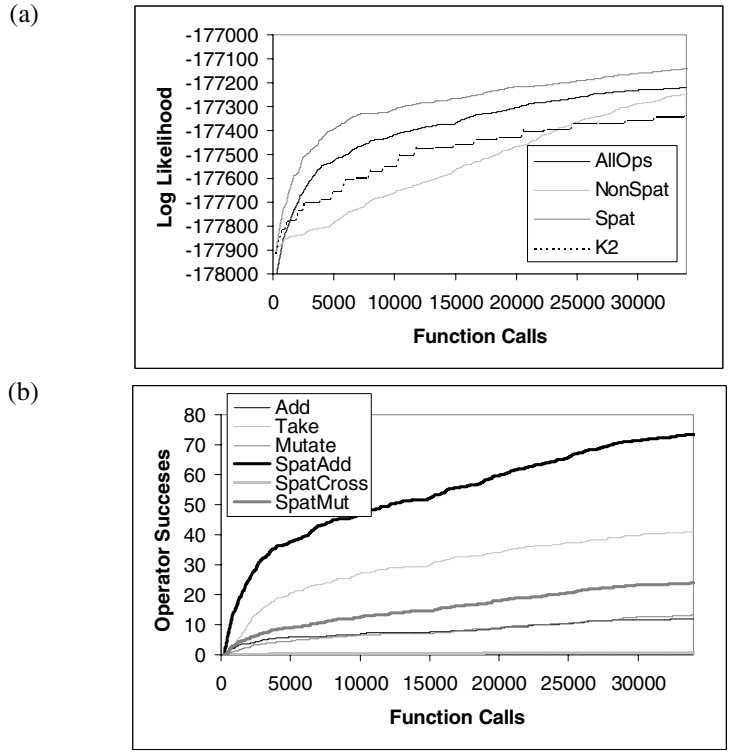

Fig. 4. (a) Mean Learning Curves and (b) Operator Successes for Synthetic Data

Table 1. Quality of Networks Learnt from Synthetic Data

	K2	Non-Spat	Spat	All
Structural Difference	119.0	142.0	122.3	129.2

Figure 5 shows the spatial nature of the final networks for the synthetic dataset. All networks appear to have strong spatial characteristics. The K2 result appears to have more links that lie on the border compared to the stochastic methods. This could be due to a bias in the search or an inability to remove links once they are discovered.

4.2 Visual Field Data

Figure 6a shows the learning curves for all methods on the VF data. It appears that the best operators are a combination of spatial and non-spatial. This results in more efficient learning curves compared to the experiments involving just spatial operators,

which in turn generate more efficient learning curves than the non-spatial operators alone. However, figure 6a implies that if the non-spatial operator experiments are run for enough iterations then they eventually overtake the spatial operator curves (probably due to the over-restrictive nature of using these operators only). The VF data, unlike the synthetic data, appears to have a mixture of spatial and non-spatial relationships. Figure 6b shows the average success of individual operators during an experiment on the VF data. The most successful were *Spatial Add*, *Take*, and *Spatial Mutation*. The worst seems to be *Spatial Crossover*, though there is a steady success rate. In fact, successes involving *Spatial Crossover* generally result in higher fitness improvements than other operators.

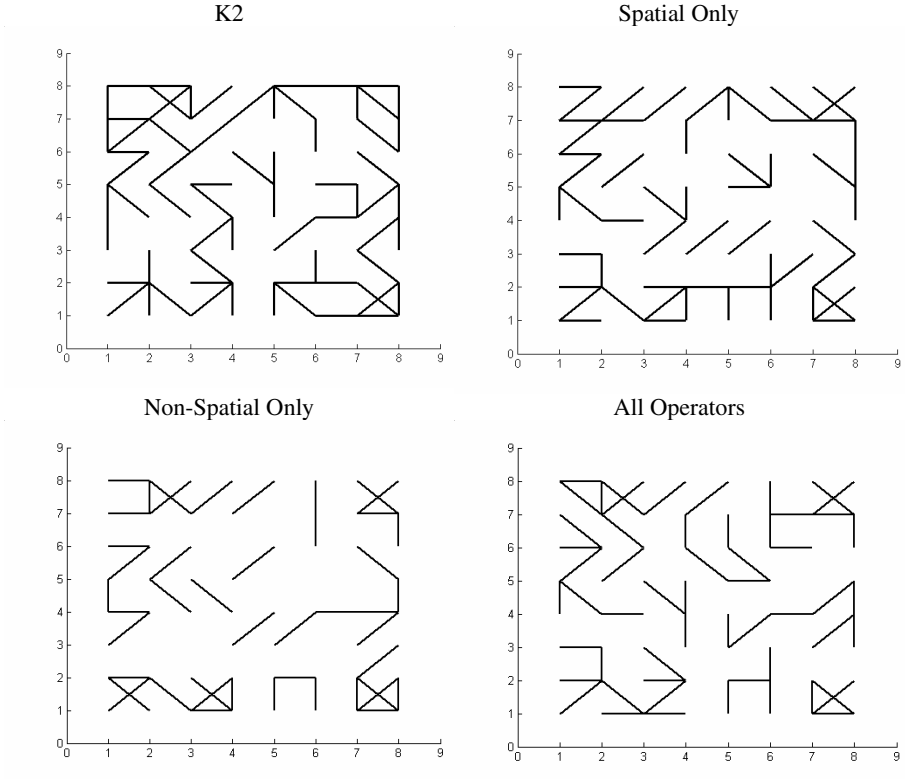

Fig. 5. Networks Learnt from Synthetic Data

We use the average Optic Nerve (ON) distance between each link's parent and child to validate the resulting networks. Table 2 shows that a similar ordering is found to that on the SD with the synthetic data but with K2 performing somewhat worse. The spatial operators generate networks that have least ON distance, followed by a combination of spatial and non-spatial, then just the non-spatial and finally the structure with largest ON distance is that generated using K2. This implies that for more complex real-world problems, a simple greedy search such as K2 is not suitable. We

have also used knowledge of how *nerve fibre bundles* are arranged on the visual field [Garway00]. Figure 7 shows how these bundles are positioned with respect to the visual field. This means that it is likely that points in the visual field that share a nerve fibre bundle are likely to be related. Grey shaded points in figure 7 correspond to the blind spot. Table 2 shows the mean number of links that are contained within the same bundle as a percentage of all links, for each search method. A similar ordering to the mean ON distance is observed: a higher percentage of links in the same bundle implies a shorter average ON distance. The highest percentage of links contained within the same bundle are from networks generated from only spatial operators, the next best being all operators, followed by non-spatial operators and the worst being those generated using K2.

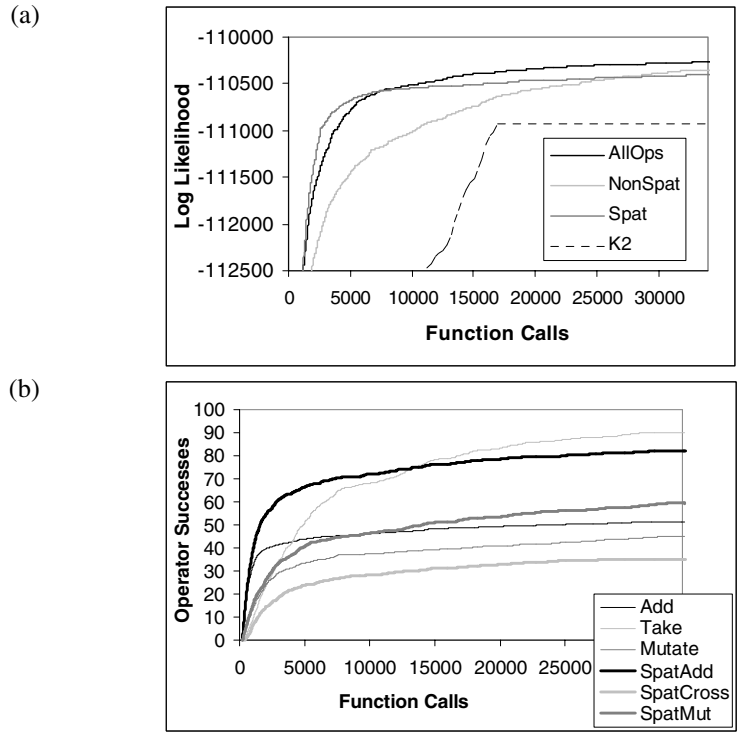

Fig. 6. (a) Mean Learning Curves and (b) Operator Successes for VF Data

Table 2. Quality of Networks learnt from VF Data

	% Links in same Bundle	Mean ON Distance
K2	62.963	41.056
Non-Spat	70.863	29.477
Spat	78.325	19.225
All	73.333	25.138

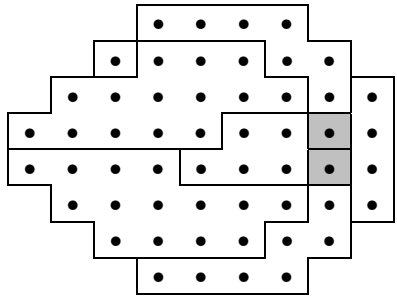

Fig. 7. Nerve Fibre Bundles

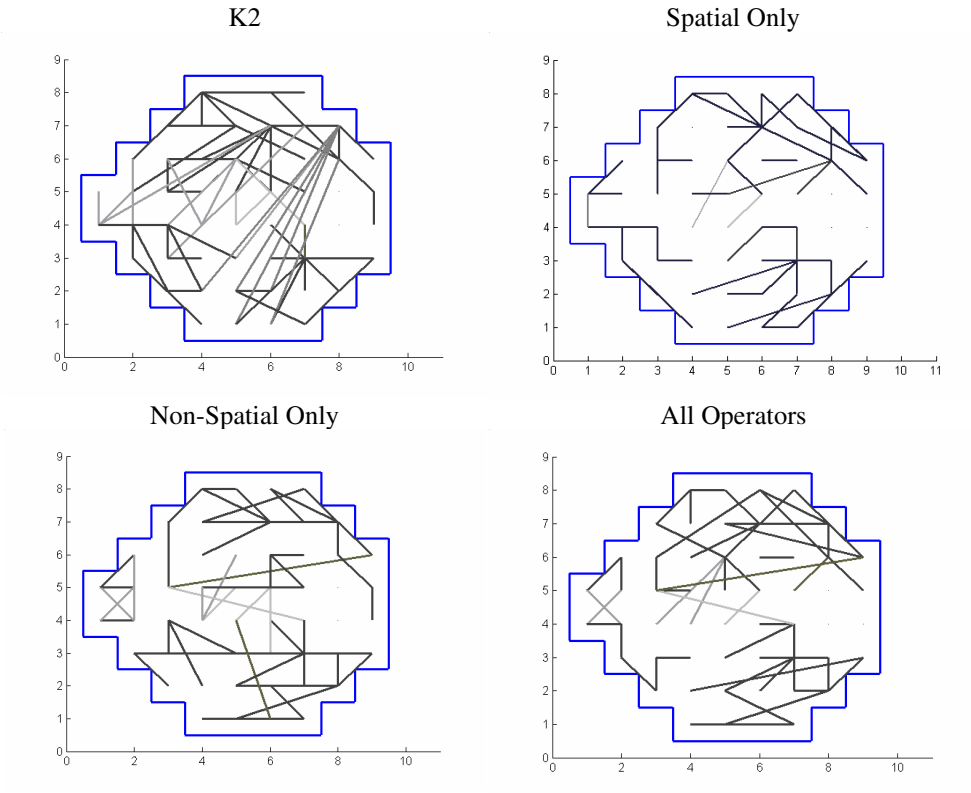

Fig. 8. Networks Learnt from VF Data

Figure 8 shows the spatial nature of the final networks for the different search methods on the VF data. ON distance is represented by the shading of the links (the larger the distance the lighter the link). Notice that the stochastic methods all have similar looking networks with strong spatial features. This is likely to be because these networks are the result of a large number of iterations and the methods have converged close to the optimum (figure 6a). The K2 result displays fewer spatial

characteristics and many links with high ON distance (in lighter grey), including links across the horizontal midline that are anatomically unlikely. None of the networks have included links involving the blind spot, as was expected. For the stochastic methods, there are still some links that have high ON distance (links in lighter grey). Most of these links appear to be between VF points that are very close to one another and cross the central horizontal axis. Some nerve fibres interdigitate across the horizontal raphe in the nasal visual field, but this does not account for some of the more central links crossing the horizontal meridian. These could be a result of bias in the spatial operators, but they still exist in the non-spatial experiments. This implies errors in measurement of the visual field where the VF is shifted slightly or the disease process acts symmetrically in the visual field in anatomically unlinked locations. These biases and errors will be explored further.

5 Conclusions and Future Work

In this paper, we have developed evolutionary and non-evolutionary operators, for learning Bayesian network structures from spatial time series data. These operators have been tested on synthetic data and real-world visual field data. We have compared the efficiency of the operators as well as a well-known straw-man learning algorithm. Results using the proposed operators were very encouraging, particularly on the real-world data, where high quality networks were found more efficiently when spatial operators were used. Various methods were used to measure network quality, including structural difference and expert knowledge on the synthetic and visual field data, respectively. In the future, we will use spatial information such as optic nerve distance and geographical direction to guide search rather than simple Cartesian coordinates. It would be interesting to see how temporal lag search [Tucker01] interacts with spatial operators, and their performance on other datasets such as rainfall prediction of cities [Cofiño02]. We intend to explore how the operators compare to other methods for learning the network structures such as Estimation of Distribution Algorithms [Larranaga01] and methods based upon neural network optimisation [Kahng95]. We also plan to extend the modelling of the visual field to include data from both eyes, as well as clinical information.

Acknowledgements. We would like to thank Nick Strouthidis for his help with collating the visual field dataset and Stephen Swift for his general help and advice. In addition, we would like to thank the EPSRC and BBSRC for funding this research.

References

Anderssen, K.E., Jeppesen, V., Classifying Visual Field Data, Technical Report, MSc Thesis, Aalborg University, Denmark, (1998).

Bengtsson, B., Olsson, J., Heijl, A., Rootzen. H., A new generation of algorithms for computerized threshold perimetry, SITA. Acta Ophthalmologica Scandinavica 75, (1997), 368–375.

Cofiño, A.S., Cano, R., Sordo C., Gutiérrez, J.M., Bayesian Networks for Probabilistic Weather Prediction, Proc. of the 15th European Conference on Artificial Intelligence. IOS Press, (2002).

Cooper, G.F., Herskovitz, E., A Bayesian Method for the Induction of Probabilistic Networks from Data, Machine Learning 9, (1992), 309–347.

Ester, M., Frommelt, A., Kriegel, H.-P., Sander, J., Spatial Data mining: Database Primitives, Algorithms and Efficient DBMS Support, Special Issue on Integration of Data Mining with Database Technology, Data Mining and Knowledge Discovery, an International Journal, Kluwer Academic Publishers 4, Nos. 2/3, (2000).

Friedman, N., Learning the Structure of Dynamic Probabilistic Networks, Proc. of the 14th Annual Conference on Uncertainty in AI, (1998), 139–147.

Garway-Heath, D.F., Fitzke, F., Hitchings, R.A., Mapping the Visual Field to the Optic Disc, Opthalmology 2000, 107, (2000), 1809–1815.

Geiger, D., Heckerman, D., Learning Gaussian Networks, Proc. of the 10th Conference in Uncertainty in Artificial Intelligence, (1994), 235–243.

Grefenstette, J.J., Optimization of Control Parameters for Genetic Algorithms, IEEE Transactions on Systems, Man & Cybernetics 16, No. 1, (1986), 122–128.

Haley, M.J. (ed.), The Field Analyzer Primer, Allergan Humphrey, San Leandro, California, (1987).

Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press, (1995).

Kahng, A.B., Moon, B.R., Toward More Powerful Recombinations, Proc. of the 6th International Conference on Genetic Algorithms, Morgan Kauffman, (1995), 96–103.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., Optimization by Simulated Annealing, Science 220, No. 4598, (1983), 671–80.

Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R., Kuijpers, C., Structure Learning of Bayesian Networks using GAs, IEEE Transactions on Pattern Analysis and Machine Intelligence 18, No.9, (1996), 912–926.

Larrañaga, P., Lozano, J.A. (eds.), Estimation of Distribution Algorithms, Kluwer, (2001).

Liu, X., Swift, S., Tucker, A., Using Evolutionary Algorithms to Tackle Large Scale Grouping Problems, GECCO (2001), 454–460.

Roddick, J.F., Spiliopoulou, M., A Bibliography of Temporal, Spatial and Spatio-Temporal Data mining Research, ACM SIGKDD Explorations, Vol. 1, No. 1, (1999), 34–38.

Bäck T., Schwefel H.-P., An Overview of Evolutionary Algorithms for Parameter Optimization, Evolutionary Computation 1, No. 1, (1993) 1–23

Tucker, A., Liu, X., Ogden-Swift, A., Evolutionary Learning of Dynamic Probabilistic Models with Large Time Lags, International Journal of Intelligent Systems 16, No. 5, (2001), 621–646.

Wong, M., Lam, W., Leung, S., Using Evolutionary Programming and Minimum Description Length Principle for Data Mining of Bayesian Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No.2, (1999), 174–178.