The Bias-Variance Tradeoff and the Randomized GACV

Grace Wahba, Xiwu Lin and Fangyu Gao
Dept of Statistics
Univ of Wisconsin
1210 W Dayton Street
Madison, WI 53706
wahba,xiwu,fgao@stat.wisc.edu

Dong Xiang
SAS Institute, Inc.
SAS Campus Drive
Cary, NC 27513
sasdxx@unx.sas.com

Ronald Klein, MD and Barbara Klein, MD
Dept of Ophthalmalogy
610 North Walnut Street
Madison, WI 53706
kleinr,kleinb@epi.ophth.wisc.edu

Abstract

We propose a new in-sample cross validation based method (randomized GACV) for choosing smoothing or bandwidth parameters that govern the bias-variance or fit-complexity tradeoff in 'soft' classification. Soft classification refers to a learning procedure which estimates the probability that an example with a given attribute vector is in class 1 *vs* class 0. The target for optimizing the the tradeoff is the Kullback-Liebler distance between the estimated probability distribution and the 'true' probability distribution, representing knowledge of an infinite population. The method uses a randomized estimate of the trace of a Hessian and mimics cross validation at the cost of a single relearning with perturbed outcome data.

1 INTRODUCTION

We propose and test a new in-sample cross-validation based method for optimizing the bias-variance tradeoff in 'soft classification' (Wahba *et al* 1994), called $ranGACV$ (randomized Generalized Approximate Cross Validation). Summarizing from Wahba *et al*(1994) we are given a training set consisting of n examples, where for each example we have a vector $t \in \mathcal{T}$ of attribute values, and an outcome y, which is either 0 or 1. Based on the training data it is desired to estimate the probability p of the outcome 1 for any new examples in the future. In 'soft' classification the estimate $\hat{p}(t)$ of $p(t)$ is of particular interest, and might be used by a physician to tell patients how they might modify their risk p by changing (some component of) t, for example, cholesterol as a risk factor for heart attack. Penalized likelihood estimates are obtained for p by assuming that the logit $f(t), t \in \mathcal{T}$, which satisfies $p(t) = e^{f(t)}/(1 + e^{f(t)})$ is in some space $\mathcal{H}$ of functions. Technically $\mathcal{H}$ is a reproducing kernel Hilbert space, but you don't need to know what that is to read on. Let the training set be $\{y_i, t_i, i = 1, \cdots, n\}$. Letting $f_i = f(t_i)$, the negative log likelihood $\mathcal{L}\{y_i, t_i, f_i\}$ of the observations, given f is

$$\mathcal{L}\{y_i, t_i, f_i\} = \sum_{i=1}^{n}[-y_i f_i + b(f_i)], \tag{1}$$

where $b(f) = log(1 + e^f)$. The penalized likelihood estimate of the function f is the solution to: Find $f \in \mathcal{H}$ to minimize $I_\lambda(f)$:

$$I_\lambda(f) = \sum_{i=1}^{n}[-y_i f_i + b(f_i)] + J_\lambda(f), \tag{2}$$

where $J_\lambda(f)$ is a quadratic penalty functional depending on parameter(s) $\lambda = (\lambda_1, ..., \lambda_q)$ which govern the so called bias-variance tradeoff. Equivalently the components of λ control the tradeoff between the complexity of f and the fit to the training data. In this paper we sketch the derivation of the $ranGACV$ method for choosing λ, and present some preliminary but favorable simulation results, demonstrating its efficacy. This method is designed for use with penalized likelihood estimates, but it is clear that it can be used with a variety of other methods which contain bias-variance parameters to be chosen, and for which minimizing the Kullback-Liebler (KL) distance is the target. In the work of which this is a part, we are concerned with λ having multiple components. Thus, it will be highly convenient to have an in-sample method for selecting λ, if one that is accurate and computationally convenient can be found.

Let p_λ be the the estimate and p be the 'true' but unknown probability function and let $p_i = p(t_i), p_{\lambda i} = p_\lambda(t_i)$. For in-sample tuning, our criteria for a good choice of λ is the KL distance $KL(p, p_\lambda) = \frac{1}{n}\sum_{i=1}^{n}[p_i log \frac{p_i}{p_{\lambda i}} + (1 - p_i) log \frac{(1-p_i)}{(1-p_{\lambda i})}]$. We may replace $KL(p, p_\lambda)$ by the comparative KL distance (CKL), which differs from KL by a quantity which does not depend on λ. Letting $f_{\lambda i} = f_\lambda(t_i)$, the CKL is given by

$$CKL(p, p_\lambda) \equiv CKL(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-p_i f_{\lambda i} + b(f_{\lambda i})]. \tag{3}$$

$CKL(\lambda)$ depends on the unknown p, and it is desired is to have a good estimate or proxy for it, which can then be minimized with respect to λ.

It is known (Wong 1992) that no exact unbiased estimate of $CKL(\lambda)$ exists in this case, so that only approximate methods are possible. A number of authors have tackled this problem, including Utans and Mọody(1993), Liu(1993), Gu(1992). The iterative UBR method of Gu(1992) is included in GRKPACK (Wang 1997), which implements general smoothing spline ANOVA penalized likelihood estimates with multiple smoothing parameters. It has been successfully used in a number of practical problems, see, for example, Wahba *et al* (1994,1995). The present work represents an approach in the spirit of GRKPACK but which employs several approximations, and may be used with any data set, no matter how large, provided that an algorithm for solving the penalized likelihood equations, either exactly or approximately, can be implemented.

2 THE GACV ESTIMATE

In the general penalized likelihood problem the minimizer $f_\lambda(\cdot)$ of (2) has a representation

$$f_\lambda(t) = \sum_{\nu=1}^{M} d_\nu \phi_\nu(t) + \sum_{i=1}^{n} c_i Q_\lambda(t_i, t) \tag{4}$$

where the ϕ_ν span the null space of J_λ, $Q_\lambda(s,t)$ is a reproducing kernel (positive definite function) for the penalized part of $\mathcal{H}$, and $c = (c_1, \cdots, c_n)'$ satisfies M linear conditions, so that there are (at most) n free parameters in f_λ. Typically the unpenalized functions ϕ_ν are low degree polynomials. Examples of $Q(t_i, \cdot)$ include radial basis functions and various kinds of splines; minor modifications include sigmoidal basis functions, tree basis functions and so on. See, for example Wahba(1990,1995), Girosi, Jones and Poggio(1995). If $f_\lambda(\cdot)$ is of the form (4) then $J_\lambda(f_\lambda)$ is a quadratic form in c. Substituting (4) into (2) results in I_λ a convex functional in c and d, and c and d are obtained numerically via a Newton Raphson iteration, subject to the conditions on c. For large n, the second sum on the right of (4) may be replaced by $\sum_{k=1}^{K} c_{i_k} Q_\lambda(t_{i_k}, t)$, where the t_{i_k} are chosen via one of several principled methods.

To obtain the $GACV$ we begin with the ordinary leaving-out-one cross validation function $CV(\lambda)$ for the CKL:

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} [-y_i f_{\lambda i}^{[-i]} + b(f_{\lambda i})], \tag{5}$$

where $f_\lambda^{[-i]}$ the solution to the variational problem of (2) with the ith data point left out and $f_{\lambda i}^{[-i]}$ is the value of $f_\lambda^{[-i]}$ at t_i. Although $f_\lambda(\cdot)$ is computed by solving for c and d the $GACV$ is derived in terms of the values $(f_1, \cdots, f_n)'$ of f at the t_i. Where there is no confusion between functions $f(\cdot)$ and vectors $(f_1, \cdots, f_n)'$ of values of f at $t_1, \cdots, t_n$, we let $f = (f_1, \cdots, , f_n)'$. For any $f(\cdot)$ of the form (4), $J_\lambda(f)$ also has a representation as a non-negative definite quadratic form in $(f_1, \cdots, f_n)'$. Letting Σ_λ be twice the matrix of this quadratic form we can rewrite (2) as

$$I_\lambda(f, y) = \sum_{i=1}^{n} [-y_i f_i + b(f_i)] + \frac{1}{2} f' \Sigma_\lambda f. \tag{6}$$

Let $W = W(f)$ be the $n \times n$ diagonal matrix with $\sigma_{ii} \equiv p_i(1 - p_i)$ in the iith position. Using the fact that σ_{ii} is the second derivative of $b(f_i)$, we have that $H = [W + \Sigma_\lambda]^{-1}$ is the inverse Hessian of the variational problem (6). In Xiang and Wahba (1996), several Taylor series approximations, along with a generalization of the leaving-out-one lemma (see Wahba 1990) are applied to (5) to obtain an approximate cross validation function $ACV(\lambda)$, which is a second order approximation to $CV(\lambda)$. Letting h_{ii} be the iith entry of H, the result is

$$CV(\lambda) \approx ACV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} [-y_i f_{\lambda i} + b(f_{\lambda i})] + \frac{1}{n} \sum_{i=1}^{n} \frac{h_{ii} y_i (y_i - p_{\lambda i})}{[1 - h_{ii}\sigma_{ii}]} . \tag{7}$$

Then the $GACV$ is obtained from the ACV by replacing h_{ii} by $\frac{1}{n}\sum_{i=1}^{n} h_{ii} \equiv \frac{1}{n} tr(H)$ and replacing $1 - h_{ii}\sigma_{ii}$ by $\frac{1}{n} tr[I - (W^{1/2} H W^{1/2})]$, giving

$$GACV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} [-y_i f_{\lambda i} + b(f_{\lambda i})] + \frac{tr(H)}{n} \frac{\sum_{i=1}^{n} y_i (y_i - p_{\lambda i})}{tr[I - (W^{1/2} H W^{1/2})]} , \tag{8}$$

where W is evaluated at f_λ. Numerical results based on an exact calculation of (8) appear in Xiang and Wahba (1996). The exact calculation is limited to small n however.

3 THE RANDOMIZED GACV ESTIMATE

Given any 'black box' which, given λ, and a training set $\{y_i, t_i\}$ produces $f_\lambda(\cdot)$ as the minimizer of (2), and thence $f_\lambda = (f_{\lambda 1}, \cdots, f_{\lambda n})'$, we can produce randomized estimates of trH and $tr[I - W^{1/2}HW^{1/2}]$ without having any explicit calculations of these matrices. This is done by running the 'black box' on perturbed data $\{y_i + \delta_i, t_i\}$. For the y_i Gaussian, randomized trace estimates of the Hessian of the variational problem (the 'influence matrix') have been studied extensively and shown to be essentially as good as exact calculations for large n, see for example Girard(1998). Randomized trace estimates are based on the fact that if A is any square matrix and δ is a zero mean random n-vector with independent components with variance σ_δ^2, then $E\delta' A\delta = \frac{1}{\sigma_\delta^2} trA$. See Gong et al(1998) and references cited there for experimental results with multiple regularization parameters. Returning to the 0-1 data case, it is easy to see that the minimizer $f_\lambda(\cdot)$ of I_λ is continuous in y, not withstanding the fact that in our training set the y_i take on only values 0 or 1. Letting $f_\lambda^y = (f_{\lambda 1}, \cdots, f_{\lambda n})'$ be the minimizer of (6) given $y = (y_1, \cdots, y_n)'$, and $f_\lambda^{y+\delta}$ be the minimizer given data $y+\delta = (y_1+\delta_1, \cdots, y_n+\delta_n)'$ (the t_i remain fixed), Xiang and Wahba (1997) show, again using Taylor series expansions, that $f_\lambda^{y+\delta} - f_\lambda^y \sim [W(f_\lambda^y) + \Sigma_\lambda]^{-1}\delta$. This suggests that $\frac{1}{\sigma_\delta^2}\delta'(f_\lambda^{y+\delta} - f_\lambda^y)$ provides an estimate of $tr[W(f_\lambda^y) + \Sigma_\lambda]^{-1}$. However, if we take the solution f_λ^y to the nonlinear system for the original data y as the initial value for a Newton-Raphson calculation of $f_\lambda^{y+\delta}$ things become even simpler. Applying a one step Newton-Raphson iteration gives

$$f_\lambda^{y+\delta,1} = f_\lambda^y - [\frac{\partial^2 I_\lambda}{\partial f' \partial f}(f_\lambda^y, y+\delta)]^{-1} \frac{\partial I_\lambda}{\partial f}(f_\lambda^y, y+\delta). \tag{9}$$

Since $\frac{\partial I_\lambda}{\partial f}(f_\lambda^y, y+\delta) = -\delta + \frac{\partial I_\lambda}{\partial f}(f_\lambda^y, y) = -\delta$, and $[\frac{\partial^2 I_\lambda}{\partial f'\partial f}(f_\lambda^y, y+\delta)]^{-1} = [\frac{\partial^2 I_\lambda}{\partial f'\partial f}(f_\lambda^y, y)]^{-1}$, we have $f_\lambda^{y+\delta,1} = f_\lambda^y + [\frac{\partial^2 I_\lambda}{\partial f'\partial f}(f_\lambda^y, y)]^{-1}\delta$ so that $f_\lambda^{y+\delta,1} - f_\lambda^y = [W(f_\lambda^y) + \Sigma_\lambda]^{-1}\delta$. The result is the following $ranGACV$ function:

$$ranGACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f_{\lambda i} + b(f_{\lambda i})] + \frac{\delta'(f_\lambda^{y+\delta,1} - f_\lambda^y)}{n} \frac{\sum_{i=1}^n y_i(y_i - p_{\lambda i})}{[\delta'\delta - \delta' W(f_\lambda^y)(f_\lambda^{y+\delta,1} - f_\lambda^y)]}. \tag{10}$$

To reduce the variance in the term after the '+' in (10), we may draw R independent replicate vectors $\delta_1, \cdots, \delta_R$, and replace the term after the '+' in (10)by $\frac{1}{R}\sum_{r=1}^R \frac{\delta_r'(f_\lambda^{y+\delta_r,1} - f_\lambda^y)}{n} \frac{\sum_{i=1}^n y_i(y_i - p_{\lambda i})}{[\delta_r'\delta_r - \delta_r' W(f_\lambda^y)(f_\lambda^{y+\delta_r,1} - f_\lambda^y)]}$ to obtain an R-replicated $ranGACV(\lambda)$ function.

4 NUMERICAL RESULTS

In this section we present simulation results which are representative of more extensive simulations to appear elsewhere. In each case, $K << n$ was chosen by a sequential clustering algorithm. In that case, the t_i were grouped into K clusters and one member of each cluster selected at random. The model is fit. Then the number of clusters is doubled and the model is fit again. This procedure continues until the fit does not change. In the randomized trace estimates the random variates were Gaussian. Penalty functionals were (multivariate generalizations of) the cubic spline penalty functional $\lambda \int_0^1 (f''(x))^2$, and smoothing spline ANOVA models were fit.

4.1 EXPERIMENT 1. SINGLE SMOOTHING PARAMETER

In this experiment $t \in [0,1], f(t) = 2sin(10t)$, $t_i = (i - .5)/500, i = 1, \cdots, 500$. A random number generator produced 'observations' $y_i = 1$ with probability $p_i = e^{f_i}/(1 + e^{f_i})$, to get the training set. Q_λ is given in Wahba(1990) for this cubic spline case, $K = 50$. Since the true p is known, the true CKL can be computed. Fig. 1(a) gives a plot of $CKL(\lambda)$ and 10 replicates of $ranGACV(\lambda)$. In each replicate R was taken as 1, and δ was generated anew as a Gaussian random vector with $\sigma_\delta = .001$. Extensive simulations with different σ_δ showed that the results were insensitive to σ_δ from 1.0 to 10^{-6}. The minimizer of CKL is at the filled-in circle and the 10 minimizers of the 10 replicates of $ranGACV$ are the open circles. Any one of these 10 provides a rather good estimate of the λ that goes with the filled-in circle. Fig. 1(b) gives the same experiment, except that this time $R = 5$. It can be seen that the minimizers $ranGACV$ become even more reliable estimates of the minimizer of CKL, and the CKL at all of the $ranGACV$ estimates are actually quite close to its minimum value.

4.2 EXPERIMENT 2. ADDITIVE MODEL WITH $\lambda = (\lambda_1, \lambda_2)$

Here $t \in [0,1] \otimes [0,1]$. $n = 500$ values of t_i were generated randomly according to a uniform distribution on the unit square and the y_i were generated according to $p_i = e^{f_i}/(1 + e^{f_i})$ with $t = (x_1, x_2)$ and $f(t) = 5\sin 2\pi x_1 - 3sin2\pi x_2$. An additive model as a special case of the smoothing spline ANOVA model (see Wahba *et al*, 1995), of the form $f(t) = \mu + f_1(x_1) + f_2(x_2)$ with cubic spline penalties on f_1 and f_2 were used. $K = 50, \sigma_\delta = .001, R = 5$. Figure 1(c) gives a plot of $CKL(\lambda_1, \lambda_2)$ and Figure 1(d) gives a plot of $ranGACV(\lambda_1, \lambda_2)$. The open circles mark the minimizer of $ranGACV$ in both plots and the filled in circle marks the minimizer of CKL. The inefficiency, as measured by $CKL(\hat{\lambda})/min_\lambda CKL(\lambda)$ is 1.01. Inefficiencies near 1 are typical of our other similar simulations.

4.3 EXPERIMENT 3. COMPARISON OF ranGACV AND UBR

This experiment used a model similar to the model fit by GRKPACK for the risk of progression of diabetic retinopathy given $t = (x_1, x_2, x_3)$ = (duration, glycosylated hemoglobin, body mass index) in Wahba *et al*(1995) as 'truth'. A training set of 669 examples was generated according to that model, which had the structure $f(x_1, x_2, x_3) = \mu + f_1(x_1) + f_2(x_2) + f_3(x_3) + f_{1,3}(x_1, x_3)$. This (synthetic) training set was fit by GRKPACK and also using $K = 50$ basis functions with $ranGACV$. Here there are $p = 6$ smoothing parameters (there are 3 smoothing parameters in f_{13}) and the $ranGACV$ function was searched by a downhill simplex method to find its minimizer. Since the 'truth' is known, the CKL for $\hat{\lambda}$ and for the GRKPACK fit using the iterative UBR method were computed. This was repeated 100 times, and the 100 pairs of CKL values appears in Figure 1(e). It can be seen that the UBR and $ranGACV$ give similar CKL values about 90% of the time, while the $ranGACV$ has lower CKL for most of the remaining cases.

4.4 DATA ANALYSIS: AN APPLICATION

Figure 1(f) represents part of the results of a study of association at baseline of pigmentary abnormalities with various risk factors in 2585 women between the ages of 43 and 86 in the Beaver Dam Eye Study, R. Klein *et al*(1995). The attributes are: x_1 = age, x_2 =body mass index, x_3 = systolic blood pressure, x_4 = cholesterol. x_5 and x_6 are indicator variables for taking hormones, and history of drinking. The smoothing spline ANOVA model fitted was $f(t) = \mu + d_1x_1 + d_2x_2 + f_3(x_3) + f_4(x_4) + f_{34}(x_3, x_4) + d_5I(x_5) + d_6I(x_6)$, where I is the indicator function. Figure 1(e) represents a cross section of the fit for $x_5 = no, x_6 = no$,

x_2, x_3 fixed at their medians and x_1 fixed at the 75th percentile. The dotted lines are the Bayesian confidence intervals, see Wahba *et al*(1995). There is a suggestion of a borderline inverse association of cholesterol. The reason for this association is uncertain. More details will appear elsewhere.

Principled soft classification procedures can now be implemented in much larger data sets than previously possible, and the $ranGACV$ should be applicable in general learning.

References

Girard, D. (1998), 'Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression', *Ann. Statist.* **126**, 315–334.

Girosi, F., Jones, M. & Poggio, T. (1995), 'Regularization theory and neural networks architectures', *Neural Computation* **7**, 219–269.

Gong, J., Wahba, G., Johnson, D. & Tribbia, J. (1998), 'Adaptive tuning of numerical weather prediction models: simultaneous estimation of weighting, smoothing and physical parameters', *Monthly Weather Review* **125**, 210–231.

Gu, C. (1992), 'Penalized likelihood regression: a Bayesian analysis', *Statistica Sinica* **2**, 255–264.

Klein, R., Klein, B. & Moss, S. (1995), 'Age-related eye disease and survival. the Beaver Dam Eye Study', *Arch Ophthalmol* **113**, 1995.

Liu, Y. (1993), Unbiased estimate of generalization error and model selection in neural network, manuscript, Department of Physics, Institute of Brain and Neural Systems, Brown University.

Utans, J. & Moody, J. (1993), Selecting neural network architectures via the prediction risk: application to corporate bond rating prediction, *in* 'Proc. First Int'l Conf. on Artificial Intelligence Applications on Wall Street', IEEE Computer Society Press.

Wahba, G. (1990), *Spline Models for Observational Data*, SIAM. CBMS-NSF Regional Conference Series in Applied Mathematics, v. 59.

Wahba, G. (1995), Generalization and regularization in nonlinear learning systems, *in* M. Arbib, ed., 'Handbook of Brain Theory and Neural Networks', MIT Press, pp. 426–430.

Wahba, G., Wang, Y., Gu, C., Klein, R. & Klein, B. (1994), Structured machine learning for 'soft' classification with smoothing spline ANOVA and stacked tuning, testing and evaluation, *in* J. Cowan, G. Tesauro & J. Alspector, eds, 'Advances in Neural Information Processing Systems 6', Morgan Kauffman, pp. 415–422.

Wahba, G., Wang, Y., Gu, C., Klein, R. & Klein, B. (1995), 'Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy', *Ann. Statist.* **23**, 1865–1895.

Wang, Y. (1997), 'GRKPACK: Fitting smoothing spline analysis of variance models to data from exponential families', *Commun. Statist. Sim. Comp.* **26**, 765–782.

Wong, W. (1992), Estimation of the loss of an estimate, Technical Report 356, Dept. of Statistics, University of Chicago, Chicago, Il.

Xiang, D. & Wahba, G. (1996), 'A generalized approximate cross validation for smoothing splines with non-Gaussian data', *Statistica Sinica* **6**, 675–692, preprint TR 930 available via `www.stat.wisc.edu/~wahba` – > TRLIST.

Xiang, D. & Wahba, G. (1997), Approximate smoothing spline methods for large data sets in the binary case, Technical Report 982, Department of Statistics, University of Wisconsin, Madison WI. To appear in the Proceedings of the 1997 ASA Joint Statistical Meetings, Biometrics Section, pp 94-98 (1998). Also in TRLIST as above.

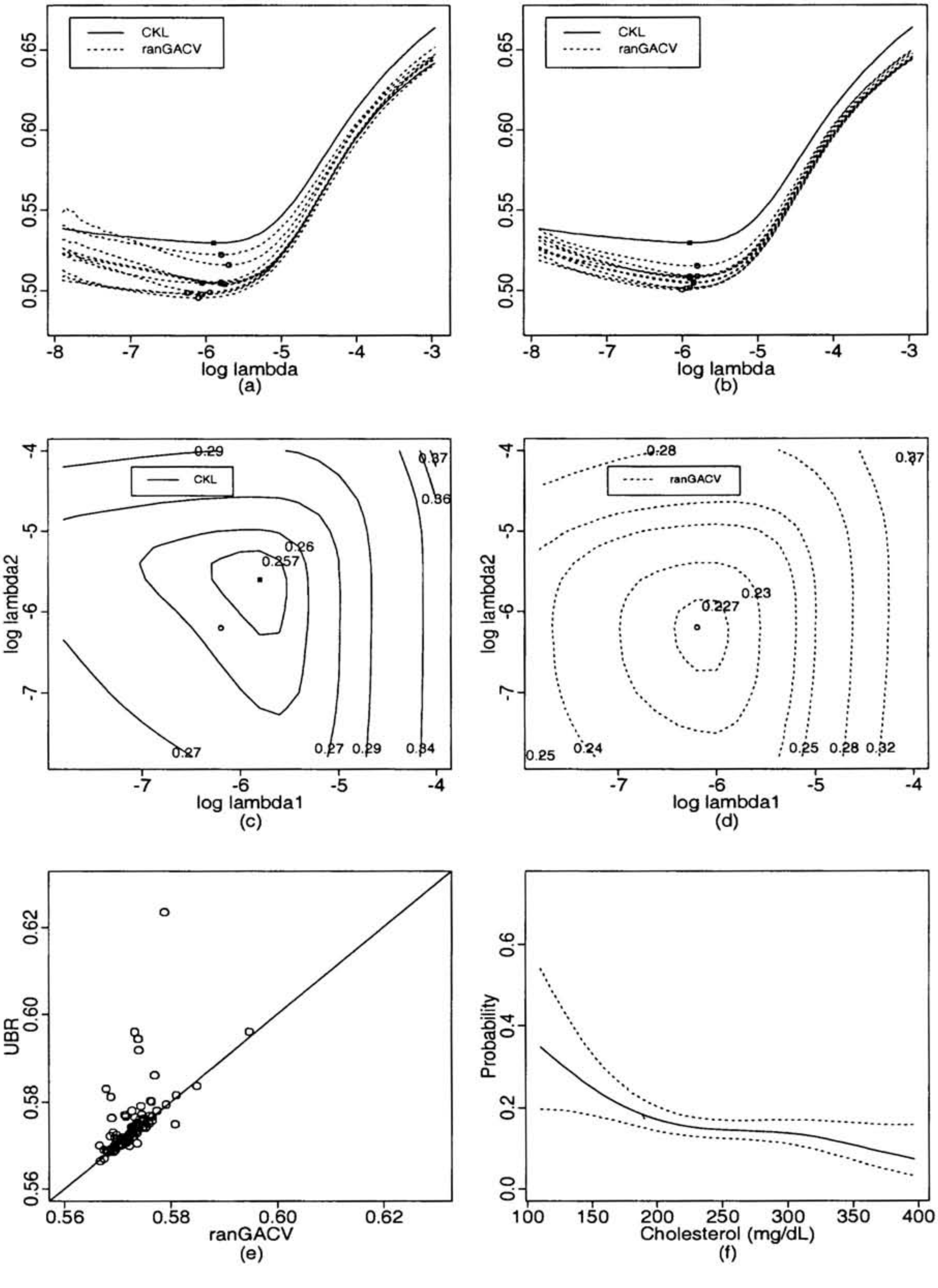


Figure 1: (a) and (b): Single smoothing parameter comparison of $ranGACV$ and CKL. (c) and (d): Two smoothing parameter comparison of $ranGACV$ and CKL. (e): Comparison of $ranGACV$ and UBR. (f): Probability estimate from Beaver Dam Study

Graph Matching for Shape Retrieval

Benoit Huet, Andrew D.J. Cross and Edwin R. Hancock*
Department of Computer Science, University of York
York, Y01 5DD, UK

Abstract

This paper describes a Bayesian graph matching algorithm for data-mining from large structural data-bases. The matching algorithm uses edge-consistency and node attribute similarity to determine the *a posteriori* probability of a query graph for each of the candidate matches in the data-base. The node feature-vectors are constructed by computing normalised histograms of pairwise geometric attributes. Attribute similarity is assessed by computing the Bhattacharyya distance between the histograms. Recognition is realised by selecting the candidate from the data-base which has the largest *a posteriori* probability. We illustrate the recognition technique on a data-base containing 2500 line patterns extracted from real-world imagery. Here the recognition technique is shown to significantly outperform a number of algorithm alternatives.

1 Introduction

Since Barrow and Popplestone [1] first suggested that relational structures could be used to represent and interpret 2D scenes, there has been considerable interest in the machine vision literature in developing practical graph-matching algorithms [8, 3, 10]. The main computational issues are how to compare relational descriptions when there is significant structural corruption [8, 10] and how to search for the best match [3]. Despite resulting in significant improvements in the available methodology for graph-matching, there has been little progress in applying the resulting algorithms to large-scale object recognition problems. Most of the algorithms developed in the literature are evaluated for the relatively simple problem of matching a model-graph against a scene known to contain the relevant structure. A more realistic problem is that of taking a large number (maybe thousands) of scenes and retrieving the ones that best match the model. Although this problem is key to data-mining from large libraries of visual information, it has invariably been approached using low-level feature comparison techniques. Very little effort [7, 4] has been devoted to matching

*corresponding author erh@cs.york.ac.uk

higher-level structural primitives such as lines, curves or regions. Moreover, because of the perceived fragility of the graph matching process, there has been even less effort directed at attempting to retrieve shapes using relational information.

Here we aim to fill this gap in the literature by using graph-matching as a means of retrieving the shape from a large data-based that most closely resembles a query shape. Although the indexation images in large data-bases is a problem of current topicality in the computer vision literature [5, 6, 9], the work presented in this paper is more ambitious. Firstly, we adopt a structural abstraction of the shape recognition problem and match using attributed relational graphs. Each shape in our data-base is a pattern of line-segments. The structural abstraction is a nearest neighbour graph for the centre-points of the line-segments. In addition, we exploit attribute information for the line patterns. Here the geometric arrangement of the line-segments is encapsulated using a histogram of Euclidean invariant pairwise (binary) attributes. For each line-segment in turn we construct a normalised histogram of relative angle and length with the remaining line-segments in the pattern. These histograms capture the global geometric context of each line-segment. Moreover, we interpret the pairwise geometric histograms as measurement densities for the line-segments which we compare using the Bhattacharyya distance.

Once we have established the pattern representation, we realise object recognition using a Bayesian graph-matching algorithm. This is a two-step process. Firstly, we establish correspondence matches between the individual tokens in the query pattern and each of the patterns in the data-base. The correspondences matches are sought so as to maximise the *a posteriori* measurement probability. Once the MAP correspondence matches have been established, then the second step in our recognition architecture involves selecting the line-pattern from the data-base which has maximum matching probability.

2 MAP Framework

Formally our recognition problem is posed as follows. Each ARG in the database is a triple, $G = (V_G, E_G, A_G)$, where V_G is the set of vertices (nodes), E_G is the edge set ($E_G \subset V_G \times V_G$), and A_G is the set of node attributes. In our experimental example, the nodes represent line-structures segmented from 2D images. The edges are established by computing the N-nearest neighbour graph for the line-centres. Each node $j \in V_G$ is characterised by a vector of attributes, $\underline{x}_j$ and hence $A_G = \{\underline{x}_j | j \in V_G\}$. In the work reported here the attribute-vector is represents the contents of a normalised pairwise attribute histogram.

The data-base of line-patterns is represented by the set of ARG's $\mathcal{D} = \{G\}$. The goal is to retrieve from the data-base $\mathcal{D}$, the individual ARG that most closely resembles a query pattern $Q = (V_Q, E_Q, A_Q)$. We pose the retrieval process as one of associating with the query the graph from the data-base that has the largest *a posteriori* probability. In other words, the class identity of the graph which most closely corresponds to the query is

$$\omega_Q = \arg\max_{G' \in \mathcal{D}} P(G'|Q)$$

However, since we wish to make a detailed structural comparison of the graphs, rather than comparing their overall statistical properties, we must first establish a set of best-match correspondences between each ARG in the data-base and the query Q. The set of correspondences between the query Q and the ARG G is a relation $f_G : V_G \mapsto V_Q$ over the vertex sets of the two graphs. The mapping function consists of a set of Cartesian pairings between the nodes of the two graphs, i.e. $f_G = \{(a,\alpha); a \in V_G, \alpha \in V_Q\} \subseteq V_G \times V_Q$. Although this may appear to be a brute force method, it must be stressed that we view this process of correspondence matching as the final step in the filtering of the line-patterns. We provide more details of practical implementation in the experimental section of this paper.

With the correspondences to hand we can re-state our maximum *a posteriori* probability recognition objective as a two step process. For each graph G in turn, we locate the maximum *a posteriori* probability mapping function f_G onto the query Q. The second step is to perform recognition by selecting the graph whose mapping function results in the largest matching probability. These two steps are succinctly captured by the following statement of the recognition condition

$$\omega_Q = \arg \max_{G' \in \mathcal{D}} \max_{f_{G'}} P(f_{G'}|G', Q)$$

This global MAP condition is developed into a useful local update formula by applying the Bayes formula to the *a posteriori* matching probability. The simplification is as follows

$$P(f_G|G,Q) = \frac{p(A_G, A_Q|f_G) P(f_G|V_G, E_G, V_Q, E_Q) P(V_G, E_G) P(V_Q, E_Q)}{P(G)P(Q)}$$

The terms on the right-hand side of the Bayes formula convey the following meaning. The conditional measurement density $p(A_G, A_Q|f_G)$ models the measurement similarity of the node-sets of the two graphs. The conditional probability $P(f_G|E_G, E_Q)$ models the structural similarity of the two graphs under the current set of correspondence matches. The assumptions used in developing our simplification of the *a posteriori* matching probability are as follows. Firstly, we assume that the joint measurements are conditionally independent of the structure of the two graphs provided that the set of correspondences is known, i.e. $P(A_G, A_Q|f_G, E_G, V_G, E_Q, V_Q) = P(A_G, A_Q|f_G)$. Secondly, we assume that there is conditional independence of the two graphs in the absence of correspondences. In other words, $P(V_G, E_G, V_Q, E_Q) = P(V_Q, E_Q)P(V_G, E_G)$ and $P(G,Q) = P(G)P(Q)$. Finally, the graph priors $P(V_G, E_G)$, $P(V_Q, E_Q)$ $P(G)$ and $P(Q)$ are taken as uniform and are eliminated from the decision making process.

To continue our development, we first focus on the conditional measurement density, $p(A_G, A_Q|f_G)$ which models the process of comparing attribute similarity on the nodes of the two graphs. Assuming statistical independence of node attributes, the conditional measurement density $p(A_G, A_Q|f_G)$, can be factorised over the Cartesian pairs $(a, \alpha) \in V_G \times V_Q$ which constitute the the correspondence match f_G in the following manner

$$p(A_G, A_Q|f_G) = \prod_{(a,\alpha) \in f_G} p(\underline{\mathrm{x}}_a, \underline{\mathrm{x}}_\alpha | f_G(a) = \alpha)$$

As a result the correspondence matches may be optimised using a simple node-by-node discrete relaxation procedure. The rule for updating the match assigned to the node a of the graph G is

$$f_G(a) = \arg \max_{\alpha \in V_Q \cup \{\Phi\}} p(\underline{\mathrm{x}}_a, \underline{\mathrm{x}}_\alpha)|f(a) = \alpha) P(f_G|E_G, E_Q)$$

In order to model the structural consistency of the set of assigned matches,we turn to the framework recently reported by Finch, Wilson and Hancock [2}. This work provides a framework for computing graph-matching energies using the weighted Hamming distance between matched cliques. Since we are dealing with a large-scale object recognition system, we would like to minimise the computational overheads associated with establishing correspondence matches. For this reason, rather than working with graph neighbourhoods or cliques, we chose to work with the relational units of the smallest practical size. In other words we satisfy ourself with measuring consistency at the edge level. For edge-units, the structural matching probability $P(f_G|V_G, E_G, V_Q, E_Q)$ is computed from the formula

$$\ln P(f_G|V_G, E_G, V_G, E_Q) = \sum_{(a,b)\in E_G} \sum_{(\alpha,\beta)\in E_Q} \{\ln(1-P_e)s_{a,\alpha}s_{b,\beta} + \ln P_e(1-s_{a,\alpha}s_{b,\beta})\}$$

where P_e is the probability of an error appearing on one of the edges of the matched structure. The $s_{a,\alpha}$ are assignment variables which are used to represent the current state of match and convey the following meaning

$$s_{a,\alpha} = \begin{cases} 1 & \text{if } f_G(a) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

3 Histogram-based consistency

We now furnish some details of the shape retrieval task used in our experimental evaluation of the recognition method. In particular, we focus on the problem of recognising 2D line patterns in a manner which is invariant to rotation, translation and scale. The raw information available for each line segment are its orientation (angle with respect to the horizontal axis) and its length (see figure 1). To illustrate how the Euclidean invariant pairwise feature attributes are computed, suppose that we denote the line segments associated with the nodes indexed a and b by the vectors $\underline{v}_a$ and $\underline{v}_b$ respectively. The vectors are directed away from their point of intersection. The pairwise relative angle attribute is given by

$$\theta_{a,b} = \arccos\left[\frac{\underline{v}_a \cdot \underline{v}_b}{|\underline{v}_a||\underline{v}_b|}\right]$$

From the relative angle we compute the directed relative angle. This involves giving the relative angle a positive sign if the direction of the angle from the baseline $\underline{v}_a$ to its partner $\underline{v}_b$ is clockwise and a negative sign if it is counter-clockwise. This allows us to extend the range of angles describing pairs of segments from $[0,\pi]$ to $[-\pi,\pi]$.

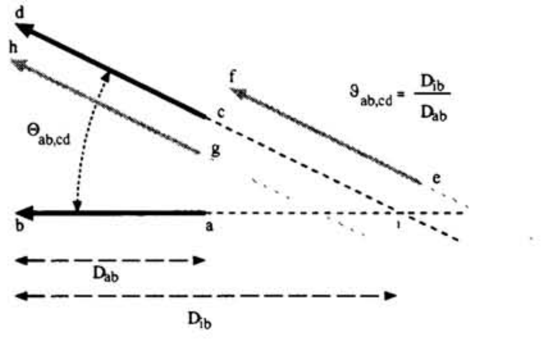


Figure 1: Geometry for shape representation

The directed relative position $\vartheta_{a,b}$ is represented by the normalised length ratio between the oriented baseline vector $\underline{v}_a$ and the vector $\underline{v}\prime$ joining the end (b) of the baseline segment (ab) to the intersection of the segment pair (cd).

$$\vartheta_{a,b} = \frac{1}{\frac{1}{2} + \frac{D_{ib}}{D_{ab}}}$$

The physical range of this attribute is $(0,1]$. A relative position of 0 indicates that the two segments are parallel, while a relative position of 1 indicates that the two segments intersect at the middle point of the baseline.

The Euclidean invariant angle and position attributes $\theta_{a,b}$ and $\vartheta_{a,b}$ are binned in a histogram. Suppose that $S_a(\mu,\nu) = \{(a,b)|\theta_{a,b} \in A_\mu \wedge \vartheta_{a,b} \in R_\nu \wedge b \in V_D\}$ is the set of nodes whose pairwise geometric attributes with the node a are spanned by the range of directed relative angles A_μ and the relative position attribute range R_ν. The contents of the histogram bin spanning the two attribute ranges is given by $H_a(\mu,\nu) = |S_a(\mu,\nu)|$. Each histogram contains n_A relative angle bins and n_R length ratio bins. The normalised geometric histogram bin-entries are computed as follows

$$h_a(\mu,\nu) = \frac{H_a(\mu,\nu)}{\sum_{\mu'=1}^{n_A} \sum_{\nu'=1}^{n_R} H_a(\mu,\nu)}$$

The probability of match between the pattern-vectors is computed using the Bhattacharyya distance between the normalised histograms.

$$P(f(a)=\alpha|\underline{\mathrm{x}}_a, \underline{\mathrm{x}}_\alpha) = \frac{\sum_{\mu=1}^{n_A} \sum_{\nu=1}^{n_R} h_a(\mu,\nu) h_\alpha(\mu,\nu)}{\sum_{j' \in Q} \sum_{\mu'=1}^{n_A} \sum_{\nu'=1}^{n_R} h_a(\mu,\nu) h_\alpha(\mu,\nu)} = \exp[-B_{a,\alpha}]$$

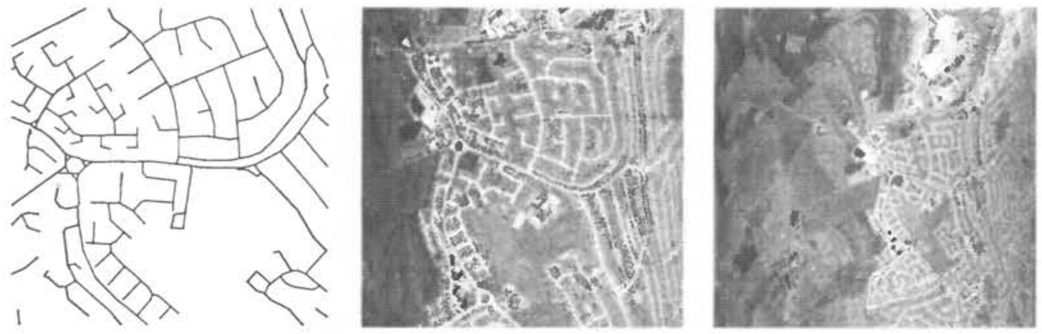
With this modelling ingredient, the condition for recognition is

$$\omega_Q = \arg \min_{G' \in \mathcal{D}} \sum_{(a,b) \in E'_G} \sum_{(\alpha,\beta) \in E_Q} \left\{ -B_{a,\alpha} - B_{b,\beta} + \ln(1-P_e) s_{a,\alpha} s_{b,\beta} + \ln P_e (1 - s_{a,\alpha} s_{b,\beta}) \right\}$$

4 Experiments

The aim in this section is to evaluate the graph-based recognition scheme on a data-base of real-world line-patterns. We have conducted our recognition experiments with a data-base of 2500 line-patterns each containing over a hundred lines. The line-patterns have been obtained by applying line/edge detection algorithms to the raw grey-scale images followed by polygonisation. For each line-pattern in the data-base, we construct the six-nearest neighbour graph. The feature extraction process together with other details of the data used in our study are described in recent papers where we have focussed on the issues of histogram representation [4] and the optimal choice of the relational structure for the purposes of recognition. In order to prune the set of line-patterns for detailed graph-matching we select about 10% of the data-base using a two-step process. This consists of first refining the data-base using a global histogram of pairwise attributes [4]. The top quartile of matches selected in this way are then further refined using a variant of the Haussdorff distance to select the set of pairwise attributes that best match against the query.

The recognition task is posed as one of recovering the line-pattern which most closely resembles a digital map. The original images from which our line-patterns have been obtained are from a number of diverse sources. However, a subset of the images are aerial infra-red line-scan views of southern England. Two of these infra-red images correspond to different views of the area covered by the digital map. These views are obtained when the line-scan device is flying at different altitudes. The line-scan device used to obtain the aerial images introduces severe barrel distortions and hence the map and aerial images are not simply related via a Euclidean or affine transformation. The remaining line-patterns in the data-base have been extracted from trademarks and logos. It is important to stress that although the raw images are obtained from different sources, there is nothing salient about their associated line-pattern representations that allows us to distinguish them from one-another.



(a) Digital Map (b) Target 1 (c) Target 2

Figure 2: Images from the data-base

Moreover, since it is derived from a digital map rather than one of the images in the data-base, the query is not identical to any of the line-patterns in the model library.

We aim to assess the importance of different attributes representation on the retrieval process. To this end, we compare node-based and the histogram-based attribute representation. We also consider the effect of taking the relative angle and relative position attributes both singly and in tandem. The final aspect of the comparison is to consider the effects of using the attributes purely for initialisation purposes and also in a persistent way during the iteration of the matching process. To this end we consider the following variants of our algorithm.

- **Non-Persistent Attributes:** Here we ignore the attribute information provided by the node-histograms after the first iteration and attempt to maximise the structural congruence of the graphs.
- **Local attributes:** Here we use only the single node attributes rather than an attribute histogram to model the *a posteriori* matching probabilities.

Graph Matching Strategy	Retrieval Accuracy	Iterations per recall
Rel. Position Attribute (Initialisation only)	39%	5.2
Rel. Angle Attribute (Initialisation only)	45%	4.75
Rel. Angle + Position Attributes (Initialisation only)	58%	4.27
1D Rel. Position Histogram (Initialisation only)	42%	4.7
1D Rel. Angle Histogram (Initialisation only)	59%	4.2
2D Histogram (Initialisation only)	68%	3.9
Rel. Position Attribute (Persistent)	63%	3.96
Rel. Angle Attribute (Persistent)	89%	3.59
Rel. Angle + Position Attributes (Persistent)	98%	3.31
1D Rel. Position Histogram (Persistent)	66%	3.46
1D Rel. Angle Histogram (Persistent)	92%	3.23
2D Histogram (Persistent)	100%	3.12

Table 1: Recognition performance of various recognition strategies averaged over 26 queries in a database of 260 line-patterns

In Table 1 we present the recognition performance for each of the recognition strategies in turn. The table lists the recall performance together with the average number of iterations per recall for each of the recognition strategies in turn. The main features to note from this table are as follows. Firstly, the iterative recall using the full histogram representation outperforms each of the remaining recognition methods in terms of both accuracy and computational overheads. Secondly, it is interesting to compare the effect of using the histogram in the initialisation-only and iteration persistent modes. In the latter case the recall performance is some 32% better than in the former case. In the non-persistent mode the best recognition accuracy that can be obtained is 68%. Moreover, the recall is typically achieved in only 3.12 iterations as opposed to 3.9 (average over 26 queries on a database of 260 images). Finally, the histogram representation provides better performance, and more significantly, much faster recall than the single-attribute similarity measure. When the attributes are used singly, rather than in tandem, then it is the relative angle that appears to be the most powerful.

5 Conclusions

We have presented a practical graph-matching algorithm for data-mining in large structural libraries. The main conclusion to be drawn from this study is that the combined use of structural and histogram information improves both recognition performance and recall speed. There are a number of ways in which the ideas presented in this paper can be extended. Firstly, we intend to explore more a perceptually meaningful representation of the line patterns, using grouping principals derived from Gestalt psychology. Secondly, we are exploring the possibility of formulating the filtering of line-patterns prior to graph matching using Bayes decision trees.

References

[1] H. Barrow and R. Popplestone. Relational descriptions in picture processing. *Machine Intelligence*, 5:377–396, 1971.

[2] A. Finch, R. Wilson, and E. Hancock. Softening discrete relaxation. *Advances in NIPS 9, Edited by M. Mozer, M. Jordan and T. Petsche, MIT Press*, pages 438–444, 1997.

[3] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE PAMI*, 18:377–388, 1996.

[4] B. Huet and E. Hancock. Relational histograms for shape indexing. *IEEE ICCV*, pages 563–569, 1998.

[5] W. Niblack *et al.*. The QBIC project: Querying images by content using color, texture and shape. *Image and Vision Storage and Retrieval*, 173–187, 1993.

[6] A. P. Pentland, R. W. Picard, and S. Scarloff. Photobook: tools for content-based manipulation of image databases. *Storage and Retrieval for Image and Video Database II*, pages 34–47, February 1994.

[7] K. Sengupta and K. Boyer. Organising large structural databases. *IEEE PAMI*, 17(4):321–332, 1995.

[8] L. Shapiro and R. Haralick. A metric for comparing relational descriptions. *IEEE PAMI*, 7(1):90-94, 1985.

[9] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[10] R. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE PAMI*, 19(6):634–648, June 1997.