

# PREDICTION OF DEFORMED AND ANNEALED MICROSTRUCTURES USING BAYESIAN NEURAL NETWORKS AND GAUSSIAN PROCESSES

C.A.L. Bailer-Jones<sup>\*,\*\*</sup>, T.J. Sabin<sup>\*</sup>, D.J.C. MacKay<sup>\*\*</sup>, P.J. Withers<sup>\*</sup>

<sup>\*</sup>Department of Materials Science and Metallurgy, University of Cambridge, Pembroke Street, Cambridge, CB2 3QZ, U.K.

<sup>\*\*</sup>Cavendish Laboratory, University of Cambridge, Madingley Road, Cambridge, CB3 0HE, U.K.

To be published in the proceedings of the Australasia-Pacific Forum on *Intelligent Processing and Manufacturing of Materials*, 1997.

## ABSTRACT

The forming of metals is important in many manufacturing industries. It has long been known that microstructure and texture affect the properties of a material, but to date limited progress has been made in predicting microstructural development during thermomechanical forming due to the complexity of the relationship between microstructure and local deformation conditions.

In this paper we investigate the utility of non-linear interpolation models, in particular Gaussian processes, to model the development of microstructure during thermomechanical processing of metals. We adopt a Bayesian approach which allows: (1) automatic control of the complexity of the non-linear model; (2) calculation of error bars describing the reliability of the model predictions; (3) automatic determination of the relevance of the various input variables. Although this method is not intelligent in that it does not attempt to provide a fundamental understanding of the underlying micromechanical deformation processes, it can lead to empirical relations that predict microstructure as a function of deformation and heat treatments. These can easily be incorporated into existing Finite Element forging design tools. Future work will examine the use of these models in reverse to guide the definition of deformation processes aimed at delivering the required microstructures.

In order to thoroughly train and test a Gaussian process or neural network model, a large amount of representative experimental data is required. Initial experimental work has focused on an Al-1%Mg alloy deformed in non-uniform cold compression followed by different annealing treatments to build up a set of microstructural data brought about by a range of processing conditions. The DEFORM Finite Element modelling package has been used to calculate the local effective strain as a function of position across the samples. This is correlated with measurements of grain areas to construct the data set with which to develop the model.

## THE METALLURGICAL PROBLEM

To optimize any component it is necessary to consider not only the alloy composition but also its microstructure. For example, in the aerospace industry, high-performance nickel alloys may go through

many hot forging and annealing stages, at great cost and over a period of several days, before the desired microstructure is obtained [1]. A reliable model for predicting microstructural evolution could greatly improve manufacturing efficiency.

The principal factors influencing microstructure during thermomechanical processing are recovery, recrystallisation and grain growth. These can be examined on the fundamental level of dislocation densities and subgrain forms and sizes [2] from which details of the recrystallisation are theoretically predictable [3]. However, such features are difficult and time-consuming to measure and lie beyond the scope of most industrial companies. A more feasible approach is to focus on more accessible processing parameters such as local temperature, strain, strain rate etc. and to use numerical models to relate their influence on microstructural features such as grain size and extent of recrystallisation [4][5].

## BAYESIAN PROBABILISTIC MODELLING OF DATA

In many cases, a prediction problem can be posed as an interpolation problem. The problem addressed in this paper is the prediction of the mean grain size ( $\langle A \rangle$ ) in a region of a cold deformed and annealed sample as a function of local strain ( $\varepsilon$ ), annealing temperature ( $T$ ) and annealing time ( $\tau$ ). One approach to a problem of this type is to calculate a physical relationship between the dependent variable and the measured variables from fundamental scientific principles [6]. Such a relationship will usually be governed by a number of parameters which have to be determined empirically [3][4]. This semi-empirical approach will generally only be reliable when applied to simple problems: Many real prediction problems will be too complex for such an approach to yield realistic or useful results [7].

For many problems, then, a more flexible approach to prediction is required. The purely empirical method makes use of a set of *training data* to characterize the relationship between a number of inputs,  $\mathbf{x}$  (such as  $\varepsilon$ ,  $T$  and  $\tau$ ) and the corresponding output,  $t$ , which we are interested in predicting (such as  $\langle A \rangle$ ). (We shall consider predictions of only one variable.) The training data set,  $D$ , consists of a set of  $N$  inputs,  $\mathbf{X}_N (= \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\})$ , and the corresponding  $N$  outputs,  $\mathbf{t}_N$ . We are interested in interpolating these data using a model in order to be able to make predictions of  $t$  at values of  $\mathbf{x}$  which are not present in the training data. Generally, the measured values of  $t$  will contain noise,  $\nu$ , so model's prediction,  $y(\mathbf{x})$ , is related to the target output by  $t = y(\mathbf{x}) + \nu$ .

A common approach to this noisy interpolation problem is to parameterize the function,  $y(\mathbf{x}, \mathbf{w})$ , where  $\mathbf{w}$  is a set of parameters which are determined from the training data using methods such as least-squares minimisation of some cost function,  $E$ . This is the approach taken by feed-forward multi-layer perceptron neural networks, which provide a suitable framework for evaluating a non-linear interpolating function (interpolant) of a set of training data [8]. The parameters in these networks are represented by a set of weights: Training the neural network is the process of calculating the optimum weights by minimizing the cost function,  $E(\mathbf{w})$ .

Bayesian probabilistic data modelling is a robust and powerful approach to prediction problems and can be readily incorporated into the neural network approach [8]. Rather than giving a single 'optimum' prediction, Bayesian methods provide a probability distribution over the predicted value. This is often very important as it can be used to produce a characteristic error in the predictions which represents the uncertainty arising from interpolating noisy data. The probability of the data given the weights, the *likelihood*, can be written  $P(D|\mathbf{w}, \beta) \propto \exp^{-\beta E}$ .  $\beta$  is a so-called *hyperparameter* which parameterizes the probability distribution and is related to the noise variance,  $\nu$ , in the target outputs. The maximum likelihood approach to training a neural network (minimizing  $E$ ) is therefore equivalent to maximizing  $P(D|\mathbf{w}, \beta)$ . However, we normally include an explicit *prior* on the weights to specify our belief concerning the distribution of the weights in the absence of any data. This can be written  $P(\mathbf{w}|\alpha)$ , where  $\alpha$  is another hyperparameter. This prior is often used to give preference to smoother interpolating functions rather than rapidly varying ones which may over-fit the training data. This prior is particularly important when trying to model sparse data sets as it will generally improve the reliability of predictions. We then apply Bayes' theorem to the prior and the likelihood

to give the posterior probability distribution of the weights given the data

$$P(\mathbf{w}|D, \alpha, \beta) = \frac{P(D|\mathbf{w}, \beta)P(\mathbf{w}|\alpha)}{P(D)} . \quad (1)$$

It is this quantity which we should maximize when training the neural network [8]. (We can ignore the denominator,  $P(D)$ , when making predictions with a single model and data set,  $D$ .)

The Bayesian approach to prediction prescribes that we marginalize (i.e. sum) over uncertain parameters. We should, therefore, ideally integrate over all values of the weights rather than optimize them. We are interested in predicting a new value,  $t_{N+1}$ , given its corresponding input,  $\mathbf{x}_{N+1}$ , and the set of training data,  $D$ . In terms of probability distributions we are interested in finding  $P(t_{N+1}|\mathbf{x}_{N+1}, D, \alpha, \beta)$ . This is obtained by integrating over all possible values of the weights:

$$P(t_{N+1}|\mathbf{x}_{N+1}, D, \alpha, \beta) = \int P(t_{N+1}|\mathbf{x}_{N+1}, D, \mathbf{w}, \alpha, \beta)P(\mathbf{w}|D, \alpha, \beta)d\mathbf{w} . \quad (2)$$

The maximum of  $P(t_{N+1}|\mathbf{x}_{N+1}, D, \alpha, \beta)$  yields the most probable prediction for  $t_{N+1}$ . The integration can be performed by *Monte Carlo* methods or by making simplifying assumptions about the form of  $P(\mathbf{w}|D, \alpha, \beta)$ . This latter approach is often equivalent to making predictions at the optimum values of the weights found by maximizing  $P(\mathbf{w}|D, \alpha, \beta)$  in equation 1.

Note that we are really interested in  $P(t_{N+1}|\mathbf{x}_{N+1}, D)$  rather than  $P(t_{N+1}|\mathbf{x}_{N+1}, D, \alpha, \beta)$ . This is obtained from equation 2 by also integrating over the hyperparameters  $\alpha$  and  $\beta$  (although it is often adequate to optimize  $\alpha$  and  $\beta$ ). These hyperparameters are important because they control the complexity of the model. They are distinct from *parameters* (i.e. the network weights) which parameterize the input–output mapping. One of the advantages of the Bayesian approach to data modelling is that it automatically embodies complexity control by means of these hyperparameters [9].

## GAUSSIAN PROCESSES FOR MODELLING

From the Bayesian perspective, we are interested only in  $P(t_{N+1}|\mathbf{x}_{N+1}, D)$ : we are not interested in the network weights themselves. Given that we should integrate over all weights, a preferable model is one which does not have such weights at all. The Gaussian process can be considered as a neural network in which we have integrated over all possible values of the weights.

The Gaussian process approach to the prediction problem assumes that the joint probability distribution of any  $N$  output values,  $\mathbf{t}_N$ , is an  $N$ -dimensional Gaussian [10][11][12]

$$P(\mathbf{t}_N|\mathbf{X}_N, \Theta) = \frac{1}{Z} \exp\left(-\frac{1}{2}(\mathbf{t}_N - \boldsymbol{\mu})^T \mathbf{C}_N^{-1}(\mathbf{t}_N - \boldsymbol{\mu})\right) . \quad (3)$$

For example, this applies to the  $N$  values in the training data set,  $D$ , defined on the previous page. This distribution is completely determined by the mean,  $\boldsymbol{\mu}$ , and the covariance matrix,  $\mathbf{C}_N$ . The elements of  $\mathbf{C}_N$  are given by the covariance function,  $C_{ij} = C(\mathbf{x}_i, \mathbf{x}_j, \Theta)$ , where  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are any two inputs and  $\Theta$  is a set of hyperparameters. The form of  $C$  is important and will be discussed shortly.

Let  $\mathbf{t}_{N+1}$  be the vector composed of the output values in the training data,  $\mathbf{t}_N$ , and the point we wish to predict,  $t_{N+1}$ . As equation 3 can be applied to any value of  $N$ , it also describes the probability distribution of  $\mathbf{t}_{N+1}$ , which is an  $N+1$ -dimensional Gaussian  $P(\mathbf{t}_{N+1}|\mathbf{x}_{N+1}, \mathbf{X}_N, \Theta)$ . The predictive probability distribution for  $t_{N+1}$  is just  $P(\mathbf{t}_{N+1}|\mathbf{x}_{N+1}, \mathbf{X}_N, \Theta)/P(\mathbf{t}_N|\mathbf{X}_N, \Theta)$ , which is the one-dimensional Gaussian,  $P(t_{N+1}|\mathbf{x}_{N+1}, D, \Theta)$ . The mean and standard deviation of this Gaussian distribution can be evaluated analytically in terms of the new input value,  $\mathbf{x}_{N+1}$ , the training data,  $D$ , and the hyperparameters,  $\Theta$ . Note that this Gaussian refers to the *probability distribution* over the

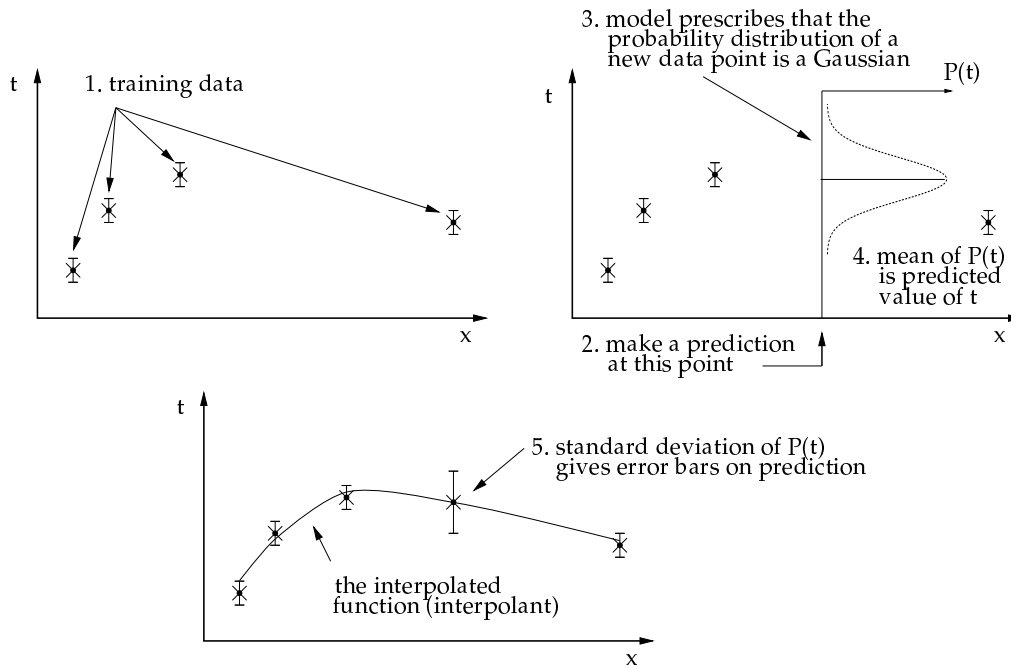


Figure 1: Schematic description of how a Gaussian process interpolates a function to make predictions. We consider here the simple case of the parameter of interest ( $t$ ) having only one dependent variable ( $x$ ). Given a set of  $N$  training data points, the model provides a predictive Gaussian probability distribution over  $t$  at a new point  $x_{N+1}$ . The mean and standard deviation of this Gaussian are evaluated from  $x_{N+1}$ , the  $N$  training data points and a set of hyperparameters. These hyperparameters (which are evaluated using the training data) control the smoothness of the interpolant. Note that we do *not* assume that the function  $t(x)$  is a sequence of Gaussians.

predicted output: we do not assume that the interpolating *function* is a Gaussian. Figure 1 summarizes schematically the prediction process with a Gaussian process with  $N = 4$ .

The form of the covariance function,  $C$ , is specified by our assumptions regarding the form of the interpolant. We will generally be interested in producing a smooth interpolant. This can be achieved by including a term in  $C$  which gives a larger positive contribution to  $C_{ij}$  the closer together two inputs are: A larger value of  $C_{ij}$  means that  $t_i$  and  $t_j$  are more closely correlated, i.e. they are more likely to have similar values. The degree of correlation achieved by a given proximity of the inputs (i.e. the smoothness of the interpolant) is dictated by the *length scale* hyperparameters which parameterize this term in  $C$ . There is one of these hyperparameters for each input dimension. In addition to controlling the smoothness of the interpolant, the relative sizes of the length scales are a measure of the relevance of each input dimension in determining the output. Thus we could assess (for example) whether local strain is more relevant than annealing temperature in determining recrystallized grain size. The noise in the data is represented in the covariance function by another term with another hyperparameter. These hyperparameters are evaluated from the training data by maximizing  $P(\Theta|D)$ , the probability of the hyperparameters given the training data. We would typically include explicit priors on the hyperparameters to express our prior knowledge of the noise in the data and the smoothness of the underlying function. Once determined, we can evaluate  $P(t_{N+1}|x_{N+1}, D, \Theta)$  to give predictions (with errors) at any new input value,  $x_{N+1}$ . Figure 2 shows the application of a Gaussian process to the interpolation of data drawn from a noisy sine function with increasing amounts of data.

## PREDICTING MICROSTRUCTURE

We have applied a Gaussian process to the problem of predicting the area of recrystallized grains in

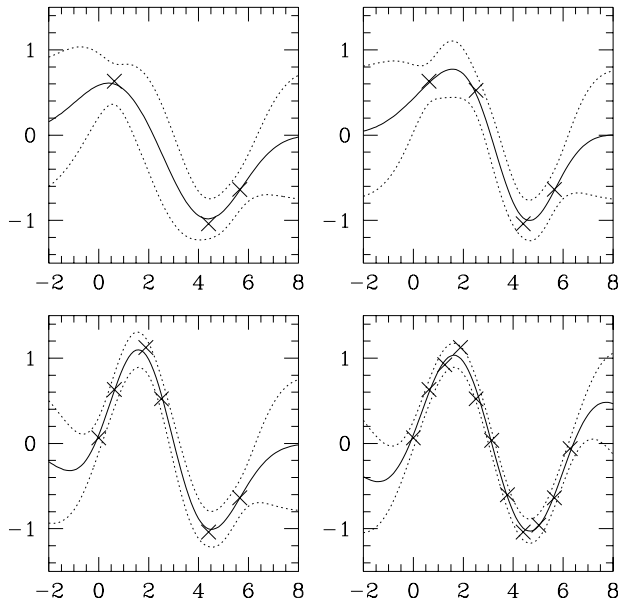


Figure 2: A Gaussian process interpolation of noisy data drawn from a sine function. The noise is Gaussian with standard deviation = 0.1. The four plots show the results of using a Gaussian process to interpolate data sets with  $N = 3, 4, 6, 11$  data points. The solid line is the interpolated function; the dashed lines are the corresponding  $1\sigma$  error bars. The accuracy with which the interpolant approximates a sine function improves as more data are added to the training data set. In regions more distant from the data the interpolant is less-well confined, and this is reflected by the larger error bars predicted by the model. This is particularly true outside the limits of the data sets where the model is attempting to extrapolate. Note that the particular covariance function used here makes the interpolant tend towards zero in regions well away from the data.

a deformed and subsequently annealed Al-1%Mg alloy as a function of local strain ( $\varepsilon$ ), annealing temperature ( $T$ ) and annealing time ( $\tau$ ). Our training data were obtained from the plane strain compression of two workpieces (20% and 40% size reduction) which were sectioned to produce a total of 20 samples [13]. These were then annealed at 325°C, 350°C or 375°C for 2, 5, 10, 30 or 60 mins. The DEFORM Finite Element package was used to calculate the local strain at different regions in these samples. This yielded a set of 57 measurements of  $\varepsilon$ ,  $T$  and  $\tau$  to act as the inputs in the training data set. The corresponding outputs—the mean grain areas,  $\langle A \rangle$ —were evaluated by sectioning the samples and measuring areas using the Kontron Elektronik KS400 package. This data set was then used to train (infer the most probable hyperparameters of) a Gaussian process model. The length scale hyperparameters indicated that the three inputs were of roughly equal relevance, as expected. The model was then used to predict the dependence of  $\langle A \rangle$  on each of  $\varepsilon$ ,  $T$  and  $\tau$  whilst holding the other two constant. These predictions are shown in Figure 3b–d. As a test of the quality of the model we trained it on 3/4 of the data and then used it to predict the  $\langle A \rangle$  values of the remaining 1/4 of the data on which it was not trained. Figure 3a shows predicted area vs. measured area for this test data. A second Gaussian process model was used to predict the extent of recrystallisation as a function of the same input parameters ( $\varepsilon$ ,  $T$ ,  $\tau$ ), and the results combined with predicted grain areas in Figure 4.

## DISCUSSION

The results presented in this paper demonstrate the feasibility of using a Gaussian process model to predict microstructural development. Figure 3a demonstrates that the model has generalized well, i.e.

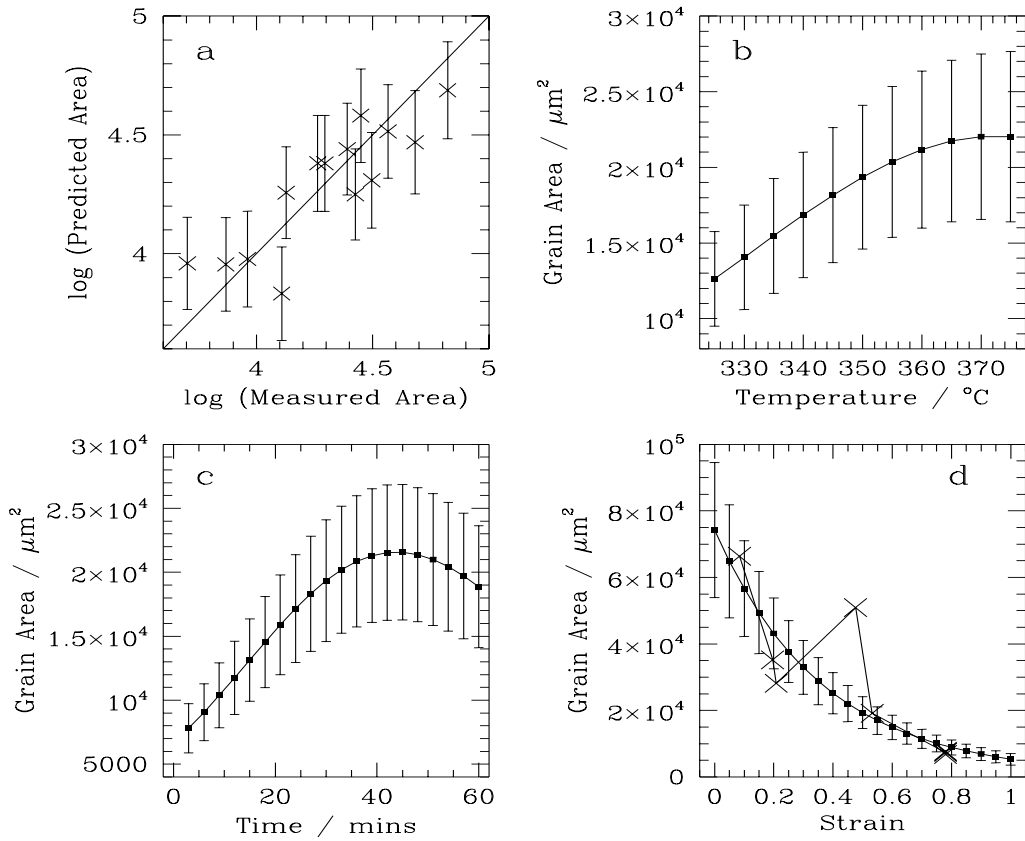


Figure 3: Gaussian process prediction of recrystallized grain area in an aluminium alloy. All errors are  $1\sigma$  errors. (a) The accuracy of the Gaussian process model was assessed by training it on 3/4 of the data and using the model to predict the remaining 1/4. The over-plotted line is the  $y = x$  line, although no model can make exact predictions due to the noise in the measured values of  $\langle A \rangle$ . Because  $\langle A \rangle$  covers a large dynamic range, the model was developed using  $\log \langle A \rangle$ . In (b)–(d) the model was trained on the full data set and used to predict  $\langle A \rangle$  as a function of each of the three input parameters with the other two held constant. When not being varied, the inputs were held constant at:  $T = 350^\circ\text{C}$ ;  $d = 30$  mins;  $\varepsilon = 0.5$ . The crosses in (d) are points from the training set: It is important to realise that the entire training set is used to make the predictions and not just the data points shown.

has identified underlying trends in the training data. These trends are shown explicitly in Figures 3b–d, and are in broad agreement with those identified by others [1]. An exception to this is the slight fall-off in  $\langle A \rangle$  at extended annealing times. However, the uncertainty predicted by the model is relatively large in this region—due to the sparseness of the training data at high  $\tau$ —and the model is not inconsistent with a levelling-off at high temperatures. A more rigorous test of the model’s capabilities is to use it to predict microstructure at different processing conditions, e.g. with different deformation geometries.

The average size of the error bars on the predictions in Figure 3a is  $\delta(\log \langle A \rangle) = 0.18$ , which is equivalent to an error in  $\langle A \rangle$  of  $\sim 40\%$ . These error bars are calculated by the model and represent the uncertainty in its predictions. The sources of this uncertainty which will be discussed shortly. In contrast to these error bars, we can measure the actual scatter (RMS error) of the predicted values of the grain areas about their measured values: this is 0.16. That these two values are similar can be seen graphically from the fact that most of the error bars on the predictions overlap with the  $y = x$  line in Figure 3a. This demonstrates that the errors predicted by the model are commensurate with the true scatter of its predictions about their measured values. A model which predicted inappropriately sized error bars would be of limited practical value, as would one which predicted no errors at all.

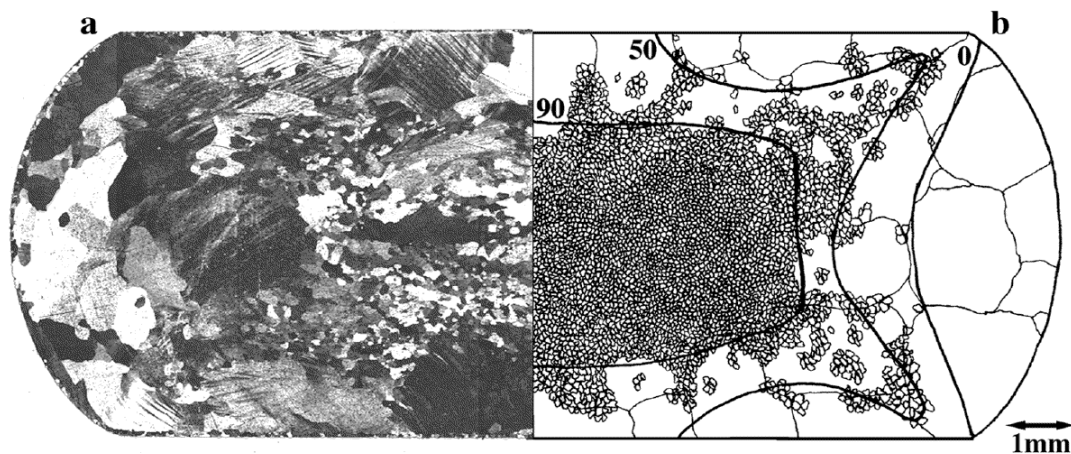


Figure 4: A 40% reduced sample annealed for 2 mins at 350°C. (a) The variation in microstructure across half a section; (b) schematic illustration of the same variation in microstructure deduced by combining the results from a Gaussian process model used to predict grain size variation and a second Gaussian process model used to predict extent of recrystallisation. The contours in (b) show 0%, 50% and 90% recrystallisation as predicted by the second Gaussian process model.

The uncertainty in the model's predictions is due largely to noise in the training data: the microstructural input data were noisy and sparse; the initial grain sizes were large (diameter  $\sim 1$  mm), making for inhomogeneous nucleation; in some cases large recrystallised grain sizes meant that the number of grains used to evaluate  $\langle A \rangle$  was fewer than ten; difficulties in accurately identifying recrystallized grains leads to biased estimations of  $\langle A \rangle$ . Despite these deficiencies, the Gaussian process model appears to provide good simulations (Figure 3b–d) which do not overfit this noisy data. This is one of the advantages of the Bayesian approach: It trades off the complexity of the model with obtaining a good fit of the training data. Smaller uncertainties in predictions can be obtained through use of a larger and more accurate data set.

## ACKNOWLEDGEMENTS

The authors are grateful to the EPSRC, DERA and INCO Alloys Limited for financial support and to Mark Gibbs for use of his Gaussian process software (obtainable from <http://wol.ra.phy.cam.ac.uk/>).

## REFERENCES

1. D. Lambert, INCO Alloys Ltd, Hereford, U.K. Private communications with T.J. Sabin, 1996.
2. G.I. Rosen, D. Juul Jensen, D.A. Hughes, N. Hansen, *Acta Metall.*, 43(7), 1996, p. 2563.
3. T. Furu, H.R. Shercliff, C.M. Sellars, M.F. Ashby, *Mat. Sci. Forum*, 217-222, 1996, p. 453.
4. J. Kusiak, M. Pietrzyk, J.-L. Chenot, *ISIJ Int.*, 34(9), 1994, p. 755.
5. P.L. Orsetti Rossi, C.M. Sellars, *Mat. Sci. Forum*, 217-222, 1996, p. 379.
6. M. Avrami, *J. Chem. Phys.*, 7, 1939, p. 1103-1112.
7. F.J. Humphreys, *Mater. Sci. Technol.*, 8, 1992, p. 135.
8. D.J.C. MacKay, *Network: Computation in Neural Systems*, 6, 1995, p. 469.
9. D.J.C. MacKay, *Neural Computation*, 4, 1992, p. 415.
10. N. Cressie, *Statistics for Spatial Data*, Wiley, Chichester, 1992.
11. C.K.I. Williams, C.E. Rasmussen, in *Advances in Neural Information Processing Systems 8* (D.S. Touretzky, M.C. Mozer, M.E. Hasselmo, eds.), MIT Press, Boston, 1996.
12. M.N. Gibbs, D.J.C. MacKay, in preparation (see <http://wol.ra.phy.cam.ac.uk/mng10/GP/>), 1997.
13. T.J. Sabin, C.A.L. Bailer-Jones, S.M. Roberts, D.J.C. MacKay, P.J. Withers, to be presented at THERMEC 97 (International Conference on Thermomechanical Processing), July 1997.