

GAUSSIAN PROCESS CLASSIFICATION USING IMAGE DEFORMATION

David P. Williams

Signal Innovations Group
Research Triangle Park, NC, USA

ABSTRACT

An image deformation algorithm is integrated with a Gaussian process classifier for application to remote-sensing tasks in which data is in the form of imagery. To combine these disparate techniques, we introduce a novel kernel covariance function for the Gaussian process that allows us to incorporate the result of the image deformation algorithm into a rigorous Bayesian classification framework. The resulting classifier is completely non-parametric in the sense that no parameters or hyperparameters must be learned. The promise of the proposed algorithm is demonstrated on a data set of real, measured land mine data.

Index Terms— Classification, image deformation, Gaussian processes, remote sensing, automatic target recognition.

1. INTRODUCTION

The goal of an automatic target recognition (ATR) task is to correctly classify objects without the aid of a human. In this work, we develop a general classification framework that is applicable when data is in the form of imagery.

There are two basic approaches to solve the ATR problem when data exists in the form of imagery. By directly using the imagery itself, one operates at the *image level*. In contrast, if features are extracted from the imagery, one operates at the *feature level*. The data processing inequality [3] states that any processing of data cannot increase the amount of information that is contained in the data. This theorem implies that information can be lost when one operates at the feature level rather than at the original image level. Nevertheless, classification is typically performed at the feature level, in which a classifier is constructed for the extracted features.

Feature extraction can be a significant obstacle in many remote-sensing applications. For example, in land mine detection problems, models for the targets (or clutter) are typically not available. As a result, rather than fitting the data to models and using the resulting model parameters as features, one must typically rely on primitive features (*e.g.*, the energy in an image chip). In underwater mine detection problems, features are often intimately linked to the result of the notoriously difficult task of image segmentation (into highlight, shadow, and background regions).

Another drawback of the feature-based approach manifests itself when a classifier is learned using training data from one location, but then applied to testing data from a different environment. The classifier can fail drastically because the underlying statistics of the data (*i.e.*, features) in each location are typically different. This fact violates an implicit assumption of the classifier that the statistics of the training and testing data are identical. Lifelong learning [13] and algorithms that account for concept drift [6] attempt to address this problem, though those techniques require the knowledge of labels of some objects in the new environment, which may be infeasible in practice.

In this work, we develop a classification algorithm for which feature extraction is obviated. Rather than describing the data points (*i.e.*, image chips) via features, the amount of deformation required to deform each image chip into another chip via an image deformation algorithm is used. Collectively, these pairwise deformation quantities are used to form the kernel covariance matrix of a Gaussian process classifier. The result of taking this approach is a classification algorithm that has *no* parameters or hyperparameters to be learned. In turn, the inherently robust classifier should generalize well when applied to data collected in different environments or under different operating conditions.

The remainder of this paper is organized in the following manner. In Section 2, the image deformation algorithm is described. Gaussian processes for classification are explained in Section 3. The proposed classification framework that integrates the image deformation technique with Gaussian processes is outlined in Section 4. Experimental results of the proposed approach on a real, measured remote-sensing data set are shown in Section 5. A summary of the work and directions for future work are outlined in Section 6.

2. IMAGE DEFORMATION

2.1. Introduction

Significant work has been conducted in the medical imaging community on image registration [7] and image segmentation [10] via non-rigid image deformation. The work we present here is the first to apply image deformation techniques to an ATR remote-sensing problem.

An algorithm that permits large-scale deformations will be necessary for our application because the deformation will be applied to pairs of images that may be highly dissimilar. Elastic material models [12], in contrast, are restricted to small deformations because severe penalties discourage large deformations. These elastic models may be sufficient in applications such as brain imaging in which only minor perturbations are required, but they are not appropriate for our purposes. Other approaches that linearize nonlinear models [1] effectively assume that the deformation is small, which again conflicts with our intent. Additionally, since our goal is to develop a fully autonomous algorithm, approaches that require a human to specify landmark points or to label regions in the image must also be avoided. The viscous fluid model we employ in this work can account for large-scale deformations while preserving the topology of structures in the original image. Moreover, no human assistance is required.

2.2. Theory

In [2], a simplified Navier-Poisson viscous fluid model that governs a non-rigid image deformation process is derived. Specifically, the model explains how particles — which in this case are the image pixels — “flow” in the image during a deformation process that attempts to transform a *template* image, \mathbf{T} , into a *study* image, \mathbf{S} . That is, the *velocity*, \mathbf{v} , of each pixel and its resulting *displacement*, \mathbf{u} , (from its initial location in the original template image) are determined via this fluid model. This flow of pixels is driven by a set of *body forces*, \mathbf{b} , acting on the image pixels.

The interplay of these quantities to deform a template image into a study image is expressed via a modified Navier-Stokes equation,

$$\mu \nabla^2 \mathbf{v}(x, y) + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{v}(x, y)) + \mathbf{b}(x, y) = \mathbf{0}, \quad (1)$$

where (x, y) indicates a specific pixel location in the image, and $\nabla^2 = \nabla^T \nabla$ is the Laplacian operator. In (1), μ and λ are the Lamé constants that control the relative amount of constant-volume viscous flow and the propensity of a volume to grow or shrink. The velocity at each (x, y) pixel location in the image are the quantities for which (1) must be solved.

The velocity is defined in terms of the displacement as

$$\mathbf{v}(x, y) = \frac{\partial \mathbf{u}(x, y)}{\partial t} + \left([\mathbf{v}(x, y)]^T \nabla \right) \mathbf{u}(x, y). \quad (2)$$

The second term on the right-hand side of (2) accounts for nonlinearities of the displacement, which in turn permit large-scale deformations of the image.

The body forces drive the deformation process that deforms the template image into the study image. The body force at the image location (x, y) at time t is defined to be

$$\mathbf{b}^{(t)}(x, y) = -\alpha (\mathbf{T}(x', y') - \mathbf{S}(x, y)) (\nabla \mathbf{T}(x', y')) \quad (3)$$

where $(x', y') = (x, y) - (u_x^{(t)}(x, y), u_y^{(t)}(x, y))$ accounts for the deformations that have transpired up until time t , and α is a scalar that controls the relative magnitudes of the body forces. It can readily be seen from (3) that the body forces act to encourage the template and study images to match.

In summary, body forces manifested by a difference between the template and study images drive a system of partial differential equations that is solved to obtain the velocities at which each pixel in the image deforms. This process is conducted iteratively until the deformed template image matches the study image.

2.3. Implementation

To solve the partial differential equation (1), the spatial derivatives are first discretized using a second-order central difference finite difference method. These equations can then be re-written in the form of a matrix equation, which must be solved for the pixel velocities. Here, the method of successive overrelaxation (SOR) is employed to solve the equation at each iteration (*i.e.*, time step) of the deformation process. After solving for the velocities via SOR, forward Euler integration is applied to (2) by discretizing the temporal derivative in the relationship between the velocity and the displacement.

After calculating the displacement for each pixel location at one time instant, the entire process is repeated for the next time step: the body forces are recomputed using (3), the matrix difference equation is solved for the velocities using SOR, and the Euler integration is performed again to obtain the updated displacement. This process is repeated until the deforming template image sufficiently matches the study image, indicating convergence. Due to space constraints here, more thorough implementation details are provided in [15].

3. GAUSSIAN PROCESSES

3.1. Introduction

In probit regression [8], the probability of label $y_i \in \{+1, -1\}$ for the i -th data vector \mathbf{x}_i is

$$p(y_i | \mathbf{x}_i) = \Phi(y_i f(\mathbf{x}_i)) = \Phi(y_i \mathbf{w}^T \mathbf{x}_i), \quad (4)$$

where Φ is the cumulative density function of the standard normal distribution, and \mathbf{w} constitutes a classifier. In a *parametric* binary classification task, one wishes to learn the function $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$ — or equivalently the classifier \mathbf{w} — that will correctly classify new unlabeled data points.

Alternatively, non-parametric approaches avoid the explicit parameterization of f , and hence do not obtain a classifier \mathbf{w} explicitly. Because the form of the classifier is not limited to a specific parametric form (such as the simple inner product in (4)), non-parametric techniques are more general than parametric approaches. One such non-parametric technique, Gaussian processes [11], places a prior on the space of latent

functions f directly. The Gaussian process prior can then be readily incorporated into a Bayesian framework for classification [11, 14].

3.2. Theory

A Gaussian process (GP) is a collection of random variables, any finite number of which has a joint Gaussian distribution. A GP is fully specified by a mean function and a covariance function. In classification problems, the mean function is typically taken to be the zero function, which we also assume. The (kernel) covariance function K_{ij} , which expresses the covariance between the value of a latent function f at the points \mathbf{x}_i and \mathbf{x}_j , is usually chosen to be a Gaussian covariance function of the form

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left\{ -\frac{1}{2}g(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}) \right\} \quad (5)$$

with

$$g(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}) = \sum_{m=1}^d \theta_m (x_i^m - x_j^m)^2 \quad (6)$$

and where x_i^m is the m -th feature of \mathbf{x}_i and $\boldsymbol{\theta}$ is a vector of length-scale hyperparameters. The kernel function essentially provides a measure of similarity between pairs of data points; the hyperparameters $\boldsymbol{\theta}$ weight the relative importance and contribution of each feature in measuring this similarity. Collecting these similarity measures for each pair of data points then forms the kernel covariance matrix, \mathbf{K} , of the GP.

The kernel covariance function K_{ij} drives the entire Gaussian process. In fact, the problem of learning a GP classifier is simply the problem of learning the kernel function hyperparameters $\boldsymbol{\theta}$. With K_{ij} defined and $\boldsymbol{\theta}$ specified, making predictions with the GP classifier is a well-established technique. In our implementation, we employed a probit model and the (now-standard) Expectation-Propagation (EP) approach [9] to combat the well-known intractable integral that arises. Due to space constraints, we omit the details of GP classification here, instead directing the interested reader to [5]. With the GP classifier learned, the probability that any new unlabeled (test) data point belongs to each class can be computed easily.

4. CLASSIFICATION

Central to our proposed classification algorithm is a novel covariance function that we employ to integrate the result of the image deformation algorithm into the Bayesian classification framework of the Gaussian process.

The deformation process between a pair of images provides a measure of similarity without first resorting to feature extraction. As a result, instead of employing the standard Gaussian covariance function with (6), we define the function $g(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$ in (5) as

$$g(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}) = \Delta(\mathbf{x}_i, \mathbf{x}_j) + \Delta(\mathbf{x}_j, \mathbf{x}_i) \quad (7)$$

where an abuse of notation allows us to use \mathbf{x}_i to represent the i -th *image* (rather than the vector of features extracted from the i -th image). In (7), $\Delta(\mathbf{x}_i, \mathbf{x}_j)$ is the average distance by which each pixel in the image must be displaced to deform the i -th image into the j -th image. This deformation measure is quantified by

$$\Delta(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{N_r N_c} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \sqrt{(u_x(r, c))^2 + (u_y(r, c))^2} \quad (8)$$

where $u_x(r, c)$ and $u_y(r, c)$ are the final x -direction and y -direction displacements from the deformation process at the pixel location (r, c) , respectively.

The definition of (7) is composed of two terms to ensure that the covariance matrix \mathbf{K} is symmetric (*i.e.*, $K_{ij} = K_{ji}$) and positive definite, as is required for the GP algorithm. The task of learning a GP classifier is that of learning the hyperparameters $\boldsymbol{\theta}$. Defining $g(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$ as in (7) implies that *no* classifier learning must be performed, as our formulation has no parameters to be learned.

Assume we are given L labeled images (labeled as containing either a target (*i.e.*, mine) or clutter), and U unlabeled images that we wish to classify. We proceed to classify the unlabeled images in the following manner. First, each of the L labeled images is deformed into each of the other labeled *and* unlabeled images using the image deformation algorithm discussed in Section 2. Next, each of the U unlabeled images is deformed into each of the L labeled images. The result of the deformation between a pair of images will be the deformation quantity defined in (8). Using (7) in (5), the GP classifier can be readily computed. The output of the GP classifier gives the probability that each unlabeled image belongs to each of the two classes (targets and clutter).

5. EXPERIMENTS

To evaluate our proposed algorithm, we conduct experiments on a data set of real, measured land mine data collected from a radar sensor. The raw data is a large radar image of a scene, from which (20 pixel by 20 pixel) image chips are extracted. The objective is to properly classify each data point (*i.e.*, image chip) as a target or clutter. The data set contains 25 targets and 85 clutter. We conduct 1000 independent trials, where each trial uses a random partition of the data into labeled and unlabeled data. In every trial, however, only two data points (one data point from each class) are treated as labeled training data; the remaining 108 unlabeled data points are used as testing data.

In addition to our proposed algorithm employing the image deformation technique, we consider two alternative methods. All three methods employ a GP classifier. The proposed approach and the first alternative method both use (7) in defining the covariance function; the difference between the two methods is that the alternative method defines $\Delta(\mathbf{x}_i, \mathbf{x}_j)$ as

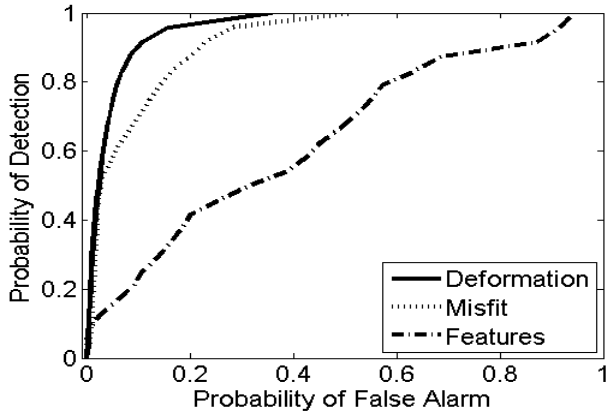


Fig. 1. Experimental results.

the misfit between two chips *without* performing any deformation:

$$\Delta(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\frac{1}{N_r N_c} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} (\mathbf{x}_i(r, c) - \mathbf{x}_j(r, c))^2}. \quad (9)$$

This alternative is considered to illustrate the necessity of deforming the images before computing a similarity measure.

In the second alternative method considered, each image chip is characterized by three basic features. The features are the mean pixel value of the chip, the variance of the pixel values of the chip, and the spatial variance of the pixel of the chip (computing via a correlation with a “checkerboard template”). This approach uses these features in conjunction with (6), instead of (7); the hyperparameters θ in (6) are learned via evidence maximization [5]. This alternative method is considered to illustrate the limitations of the feature-based approach for the data set under study.

We present classification results in the form of receiver operating characteristic (ROC) curves (based on the output of the GP classifier) in Figure 1. As can be observed from Figure 1, the proposed approach outperforms the alternative approaches. The ROC curves show the average performance (over the 1000 trials) on the testing data. To prevent cluttering the figure, error bars for the curves are withheld. In lieu of error bars, the area under the ROC curve (AUC) [4] is used to perform paired *t*-tests, according to which the proposed algorithm is statistically significantly better than the two alternatives, at a confidence level of 95%.

6. CONCLUSION

We have integrated an image deformation algorithm with a Gaussian process classifier for application to remote-sensing tasks in which data is in the form of imagery. To combine these disparate techniques, we introduced a novel kernel function for the Gaussian process that allowed us to incorporate

the result of the image deformation algorithm into a rigorous Bayesian classification framework. The promise of the proposed algorithm was demonstrated on a data set of real, measured land mine data.

By combining the novel covariance function with the GP framework, a non-parametric classifier — meaning a classifier with *no* parameters *or* hyperparameters — can be constructed. This aspect of the algorithm suggests that the classifier may have the ability to generalize well to new environments with different characteristics because the classifier will not have learned parameters specific to the training data or site. In the future, more extensive experiments will investigate this potential generalization ability of the approach.

7. REFERENCES

- [1] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Computer Vision, Graphics and Pattern Recognition*, 46:1–21, 1989.
- [2] G. Christensen, R. Rabbitt, and M. Miller. Deformable templates using large deformation kinematics. *IEEE Trans. on Image Processing*, 5(10):1435–1447, 1996.
- [3] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.
- [4] J. Hanley and B. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.
- [5] M. Kuss and C. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.
- [6] X. Liao, Y. Xue, and L. Carin. Logistic regression with an auxiliary data source. In *ICML*, pages 505–512, 2005.
- [7] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–37, 1998.
- [8] P. McCullagh and J. Nelder. *Generalized Linear Models, 2nd Edition*. Chapman & Hall, 1989.
- [9] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT, 2001.
- [10] D. Pham, C. Xu, and J. Prince. Current methods in medical image segmentation. *Annu. Rev. Biomedical Eng.*, 2:315–337, 2000.
- [11] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [12] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205–214, 1987.
- [13] S. Thrun and T. Mitchell. Learning one more thing. Technical Report CMU-CS-94-184, Carnegie Mellon University, Pittsburgh, PA, September 1994.
- [14] C. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(12), 1997.
- [15] D. Williams and V. Myers. Classification via image deformation: Theory and implementation details. Technical report, Available at www.duke.edu/~dpw5, 2006.