Gaussian Process Classification Using Image Deformation

David P. Williams

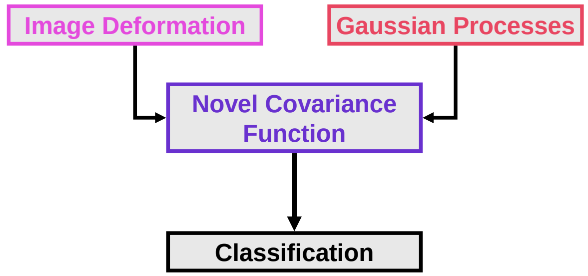
dwilliams@siginnovations.com

Signal Innovations Group, 1009 Slater Road, Suite 200, Durham, NC 27703, USA

Hawaii 2007 ICASSP

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.



Summary:

- A new, novel covariance function integrates the result of an image deformation algorithm into a Gaussian process classification framework
- Method is relevant when data points are initially in the form of images
- Approach obviates feature extraction process
- Image deformation algorithm permits large-scale deformations, preserves topology of structure in original image, and requires no human assistance
- Gaussian process classifier is general and has no parameters to learn
- Paper and extensive additional details available at www.duke.edu/~dpw5

- Image deformation objective: Deform a template image *(T)* into a study image *(S)*
- Simplified viscous fluid model governs the non-rigid deformation process
- Model explains how "particles" (image pixels) "flow" (are displaced) in the image during deformation
- Pixel displacement *(u)* is driven by a set of body forces *(b)*
- Body forces are manifested by a difference between the template and study images
- Model PDE is solved for velocity *(v)* of each pixel location *(x,y)*
- Process is conducted iteratively until deformed template image matches study image

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

- Discretize spatial derivatives using second-order finite difference method
- Solve resulting matrix equation for velocity via successive overrelaxation

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

- Discretize temporal derivative using forward Euler integration
- Solve resulting equation for displacement

$$(x',y') = (x,y) - (u_x^{(t)}(x,y), u_y^{(t)}(x,y))$$

- Calculate new pixel locations

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

- Re-compute body forces

- 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 (kernel) covariance function.
- Covariance function provides measure of similarity between pairs of data points.
- The problem of learning a GP classifier is the problem of learning the hyperparameters of the covariance function.
- Implementation of GP employed probit model and Expectation-Propagation approach.

- A new, novel covariance function integrates the result of the image deformation algorithm into the Gaussian process classification framework

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

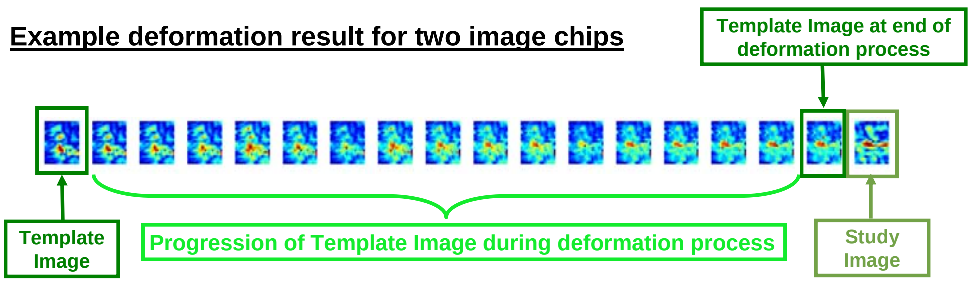
$$g(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{\theta}) = \Delta(\boldsymbol{x}_i,\boldsymbol{x}_j) + \Delta(\boldsymbol{x}_j,\boldsymbol{x}_i)$$

- Deformation process provides measure of similarity between pair of images
- Average distance by which each pixel in the image must be displaced to deform the *i*-th image into the *j*-th image:

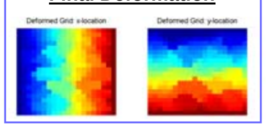
$$\Delta(\boldsymbol{x}_i,\boldsymbol{x}_j) = \frac{1}{N_rN_c}\sum_{r=1}^{N_r}\sum_{c=1}^{N_c}\sqrt{(u_x(r,c))^2 + (u_y(r,c))^2}$$

- No covariance function hyperparameters to learn for GP classifier

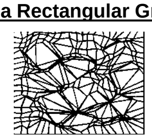
Example deformation result for two image chips

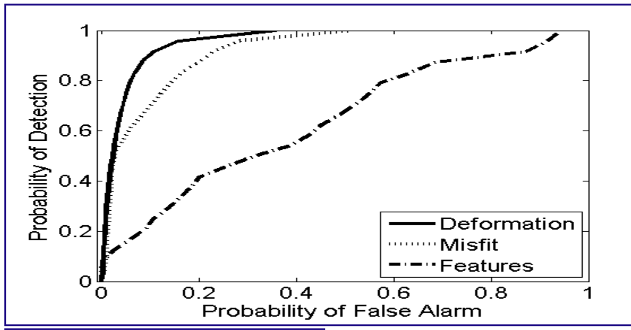


Final Deformation



Final Deformation Applied to a Rectangular Grid





Objective: Detect land mines in a scene from an airborne sensor that collects a ground-penetrating radar image.

110 objects (25 mines, 85 clutter) detected, each characterized by a 20 pixel by 20 pixel image chip.

Train GP classifier on 2 objects (1 mine and 1 clutter, randomly selected), test on remaining 108 objects.

ROC curves are averages over 1000 trials.

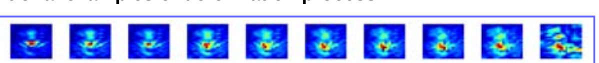
Deformation: Proposed method.

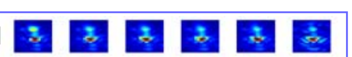
Misfit: New covariance function, but with misfit defined (*without* performing deformation) as

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

Features: Extract 3 features from each chip; use standard (Gaussian) covariance function.

Additional examples of deformation process:

[1] 

[2] 

[3]