Generalised Kernel Machine Toolbox

Gavin Cawley, University of East Anglia, Norwich, U.K.

http://theoval.cmp.uea.ac.uk/~gcc/projects/gkm

A MATLAB Toolbox for Machine Learning

- Kernel machines with exponential family likelihood [1].
- Object oriented design, for simplicity of use and extendability.
- Automated model selection.
- Automated code generation using Symbolic Math Toolbox.
- Spider/CLOP interface.
- Excellent performance on WCCI-2006 [2], NIPS-2006 and IJCNN-2007 [3] challenges.
- Freely available under the GNU General Public License (GPL).

Objects

- Models
  - **@gkm** — abstract base class for all generalised kernel machines, including implementation of iteratively re-weighted least squares training algorithm.
  - Predefined kernel machines:
    - **@krr** — kernel ridge regression a.k.a. least-squares support vector machine. Primarily for regression problems.
    - **@klr** — kernel logistic regression, the preferred model for classification tasks.
  - Additional kernel machines can be generated automatically using the MATLAB Symbolic Math Toolbox.
- Kernels
  - **@kernel** — abstract base class allowing straightforward extension of the toolbox by adding new kernels.
  - Predefined kernels:
    - **@linear**
    - **@polynomial**
    - **@rbf**
    - More to be added soon!
- Model selection
  - **@estimator** — objects that implement estimators, e.g. **@aloo** implementing efficient approximate leave-one-out cross-validation.
  - **@criterion** — which implement performance criteria, **@nlp** implementing negative log-probability.
  - **@simplex** — optimise kernel and regularisation hyper-parameters using Nelder-Mead simplex algorithm.
- Performance estimation
  - **@optimized** — wrapper facilitating independent model selection in each fold, avoiding selection bias.
  - **@crossvalidation** — perform k-fold and l-o-o cross-validation for performance evaluation and model selection.
  - **@splitsample** — evaluate performance using independent training and test sets.
- CLOP/Spider Interface

Kernel Logistic Regression Example

Step 1 - Generate code for KLR

```
fix(gkm('acronym', 'klr', ...
        'name', 'kernel logistic regression', ...
        'canonical', 'log(1+exp(eta))'));
```

Step 2 - Initialise KLR machine

```
network = klr('kernel', rbf('eta', [1;1]), ...
              'lambda', 0.01, ...
              'Verbosity', 'ethereal');
```

Step 3 - Perform model selection

```
selector = simplex('estimator', aloo);

net = select(selector, network, x_train, y_train);
```

Step 4 - Generate test predictions

```
mu_test = fwd(net, x_test);
```

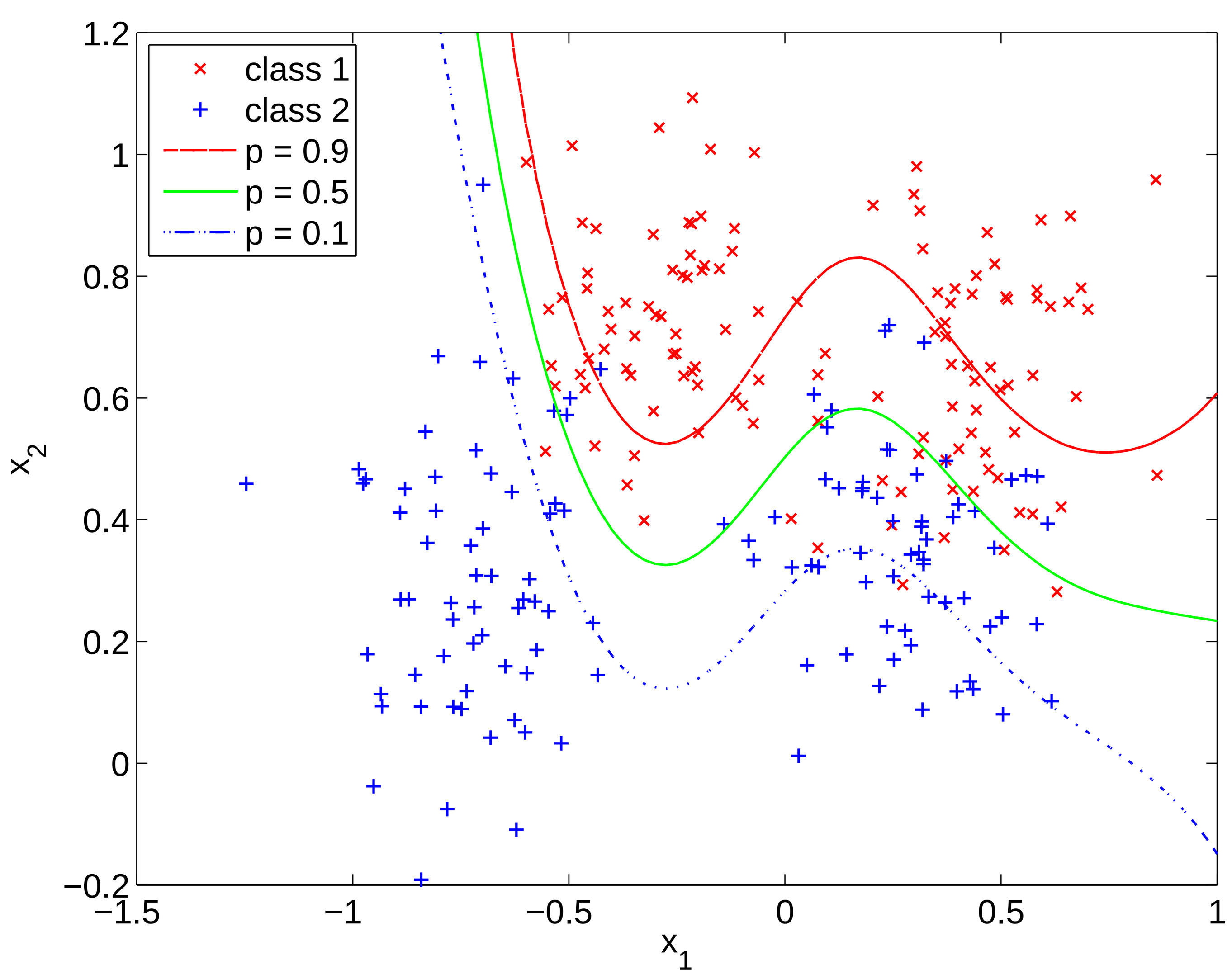
Step 5 - Generate performance estimate

```
estmtr = crossvalidation('k',10,'criterion',erate);

model = optimised('gkm',network,'selector',selector));

err = estimate(estmtr, model, x_train, y_train);
```

Step 6 - Plot decision surface



References

[1] G. C. Cawley, G. J. Janacek, and N. L. C. Talbot. Generalised kernel machines. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-07)*, pages 1732–1737, Orlando, Florida, USA, August 12–17 2007.

[2] G. C. Cawley. Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-2006)*, pages 2970–2977, Vancouver, BC, Canada, July 16–21 2006.

[3] G. C. Cawley and N. L. C. Talbot. Agnostic learning versus prior knowledge in the design of kernel machines. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-07)*, pages 1720–1725, Orlando, Florida, USA, August 12–17 2007.