Generalised Kernel Machines

Gavin C. Cawley Gareth J. Janacek Nicola L. C. Talbot

***Abstract*— The generalised linear model (GLM) is the standard approach in classical statistics for regression tasks where it is appropriate to measure the data misfit using a likelihood drawn from the exponential family of distributions. In this paper, we apply the *kernel trick* to give a non-linear variant of the GLM, the generalised kernel machine (GKM), in which a regularised GLM is constructed in a fixed feature space implicitly defined by a Mercer kernel. The MATLAB symbolic maths toolbox is used to automatically create a suite of generalised kernel machines, including methods for automated model selection based on approximate leave-one-out cross-validation. In doing so, we provide a common framework encompassing a wide range of existing and novel kernel learning methods, and highlight their connections with earlier techniques from classical statistics. Examples including kernel ridge regression, kernel logistic regression and kernel Poisson regression are given to demonstrate the flexibility and utility of the generalised kernel machine.**

I. Generalised Linear Models

Assume we are given data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\ell}$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ represents a vector of input variables and $y_i \in \mathbb{R}$ represent the corresponding responses. Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_\ell)$ represent the vector of responses, which we will assume is a realisation of a random variable, $\boldsymbol{Y}$, the components of which are identically and independently distributed (i.i.d.), with means given by the vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_\ell)$. The aim of regression is to estimate the conditional mean of the response, μ_i, as a function of the covariates, $\boldsymbol{x}_i$ for $i = 1, 2, \ldots, \ell$. A generalised linear model [22, 28] consists of three components: First a random component, which describes the conditional distribution of the responses, with mean vector $E[\boldsymbol{Y}] = \boldsymbol{\mu}$. For example, assuming the responses are normally distributed with common variance, σ^2, we have

$$\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I}).$$

Secondly, the systematic component (indirectly) describes relationship between the covariates and the mean of the random component, through a vector of latent variables, $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_\ell)$, in this case,

$$\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta} \tag{1}$$

where the rows of the design matrix $\boldsymbol{X} = [\boldsymbol{x}_i]_{i=1}^{\ell}$ are given by the covariate vectors and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_\ell)$ is the vector of model parameters. Lastly, a monotonic *link* function, $g(\cdot)$, that relates the systematic and random components, such that

$$\eta_i = g(\mu_i). \tag{2}$$

Gavin Cawley, Gareth Janacek and Nicola Talbot are with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K.; E-mail: {gcc,gjj,nlct}@cmp.uea.ac.uk

A. The Random Component

The generalised linear model extends the standard least-squares linear regression technique to allow the conditional distribution of the responses to be given by any member of exponential family. The exponential family consists of all distributions of the form,

$$f(y; \theta, \phi) = \exp\left\{[y\theta - b(\theta)]/a(\phi) + c(y, \phi)\right\} \tag{3}$$

for some functions $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$, where θ and ϕ are parameters of the distribution. Many of the distributions commonly encountered in statistical modelling fall within the exponential family. For example, in the case of the normal distribution

$$\begin{aligned} f(y; \theta, \phi) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}, \\ &= \exp\left\{\left(y\mu - \frac{\mu^2}{2}\right)\sigma^{-2} \right. \\ &- \left. \frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log\left\{2\pi\sigma^2\right\}\right)\right\}, \end{aligned} \tag{4}$$

by inspection we find that it is a member of the exponential family where $\theta = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = \theta^2/2$ and

$$c(y, \phi) = -\frac{1}{2}\left[\frac{y^2}{\sigma^2} + \log\left\{2\pi\sigma^2\right\}\right].$$

It can be shown that the mean of the distribution is governed solely by the *canonical* parameter, θ, whereas the variance is governed by both θ and the *dispersion* parameter, ϕ. The log-likelihood as a function of θ and ϕ, given the observed responses y for a distribution from the exponential family can be written as

$$l(\theta, \phi; y) = [y\theta - b(\theta)]/a(\phi) + c(y, \phi). \tag{5}$$

with partial derivatives

$$\frac{\partial l}{\partial \theta} = [y - b'(\theta)]/a(\phi) \quad \text{and} \quad \frac{\partial^2 l}{\partial \theta^2} = -b''(\theta)/a(\phi). \tag{6}$$

The mean can be deduced from the relation

$$E\left[\frac{\partial l}{\partial \theta}\right] = 0. \quad \Longrightarrow \quad [\mu - b'(\theta)]/a(\phi) = 0,$$

such that

$$E[Y] = \mu = b'(\theta). \tag{7}$$

Similarly, noting that

$$E\left[\frac{\partial^2 l}{\partial \theta^2}\right] + E\left[\frac{\partial l}{\partial \theta}\right]^2 = 0,$$

we find that

$$\frac{\mathrm{var}(Y)}{a^2(\phi)} = \frac{b''(\theta)}{a(\phi)} \quad \Longrightarrow \quad \mathrm{var}(Y) = b''(\theta)a(\phi). \tag{8}$$

As our primary interest lies in estimating the conditional mean of the responses, the dispersion parameter, ϕ, is often treated as a nuisance parameter, and the emphasis is placed on estimating θ as a function of the covariates.

B. The Link Function

The main function of the link is to constrain the estimate of the conditional mean to lie within reasonable bounds for the particular member of the exponential family concerned, such that the domain of the link coincides exactly with the range of the mean of the distribution. The mean of normally distributed data is unconstrained, and so a linear link is appropriate,

$$g(\mu) = \mu. \tag{9}$$

For Binomial responses, the conditional mean lies in the range $(0,\ 1)$, and so a logit link,

$$g(\mu) = \text{logit}(\mu) = \log\left\{\frac{\mu}{1-\mu}\right\}, \tag{10}$$

is suitable. The link function such that $\theta_i = \eta_i$, is known as the *canonical* link, where

$$\eta_i = g(\mu_i) \implies \eta_i = g(b'(\eta_i)) \implies g^{-1}(\eta_i) = b'(\eta_i).$$

The use of the canonical link simplifies the optimisation procedure followed in fitting a generalised linear model. The canonical link also has many desirable statistical properties, for instance η_i becomes the sufficient statistic for the response distribution, however the choice of link is essentially arbitrary.

C. Parameter Estimation

Generalised linear models can be fitted to the data via a maximum likelihood approach, via the Newton-Raphson process. The partial derivative of the log-likelihood of the i^{th} observation, with respect to the output of the systematic component, η_i, is given by

$$\frac{\partial l_i}{\partial \eta_i} = \frac{\partial l_i}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}.$$

$$b'(\theta_i) = \mu_i \quad \implies \quad \frac{\partial \mu_i}{\partial \theta_i} = b''(\theta_i)$$

$$\frac{\partial l_i}{\partial \eta_i} = \frac{[y_i - \mu_i]}{a(\phi)}\frac{1}{b''(\theta)}\frac{\partial \mu_i}{\partial \eta_i}$$

where $\partial \mu_i / \partial \eta_i$ is simply the local gradient of the inverse link function, more commonly known as the *activation* function in the neural networks literature. In the case of the canonical link, the partial derivative can be considerably simplified, noting that

$$\theta_i = \eta_i \quad \implies \quad \mu_i = b'(\eta_i) \quad \implies \quad \frac{\partial \mu_i}{\partial \eta_i} = b''(\eta_i)$$

we obtain

$$\frac{\partial l_i}{\partial \eta_i} = \frac{[y_i - \mu_i]}{a(\phi)} \quad \implies \quad \frac{\partial l_i}{\partial \beta_j} = \frac{[y_i - \mu_i]x_{ij}}{a(\phi)}. \tag{11}$$

Again assuming the canonical link, the second order partial derivatives, with respect to the output of the systematic component, are given by

$$\frac{\partial^2 l_i}{\partial \eta_i^2} = \frac{-1}{a(\phi)}\frac{\partial \mu_i}{\partial \eta_i} = -\frac{b''(\eta_i)}{a(\phi)},$$

and those with respect to the model parameters, $\boldsymbol{\beta}$, by

$$\frac{\partial^2 l_i}{\partial \beta_j \partial \beta_k} = \frac{b''(\eta_i)x_{ij}x_{ik}}{a(\phi)}. \tag{12}$$

The relative ease of evaluating the gradient information required to fit the generalised linear model is a strong argument in favour of the canonical link. Assuming that the data, $\mathcal{D}$, represent an independent and identically distributed sample, the negative log-likelihood is given by

$$\mathcal{L} = -\sum_{i=1}^{\ell} l_i \propto \sum_{i=1}^{\ell}[y_i\theta_i - b(\theta)] \tag{13}$$

Note that since we are interested in minimising $\mathcal{L}$ with respect to $\boldsymbol{\beta}$, we can neglect any additive terms not involving $\boldsymbol{\theta}$, such as $c(y, \phi)$ or multiplicative scaling, e.g. $1/a(\phi)$. Given the gradient vector of $\mathcal{L}$, with respect to $\boldsymbol{\beta}$,

$$\boldsymbol{\Delta} = \left(\frac{\partial \mathcal{L}}{\partial \beta_i}\right)_{i=1}^{\ell} = \boldsymbol{X}^T\left[\boldsymbol{y} - \boldsymbol{\mu}\right], \tag{14}$$

and the Hessian matrix,

$$\boldsymbol{A} = \left[\frac{\partial^2 L}{\partial \beta_i \partial \beta_j}\right]_{i,j=1}^{d} = -\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}, \tag{15}$$

where $\boldsymbol{W} = \text{diag}(b''(\eta_1), b''(\eta_2), \ldots, b''(\eta_\ell))$. Newton's method updates the model parameters according to the following rule:

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \boldsymbol{A}^{-1}\boldsymbol{\Delta}. \tag{16}$$

Substituting (14) and (15) into (16) and re-arranging gives,

$$\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}\boldsymbol{\beta}_{t+1} = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} - \boldsymbol{X}^T[\boldsymbol{y} - \boldsymbol{\mu}].$$

Finally, noting that $\boldsymbol{X}\boldsymbol{\beta}_t = \boldsymbol{\eta}$ and defining

$$\boldsymbol{z} = \boldsymbol{\eta} - \boldsymbol{W}^{-1}[\boldsymbol{y} - \boldsymbol{\mu}], \tag{17}$$

we obtain

$$\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}\boldsymbol{\beta}_{t+1} = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{z}. \tag{18}$$

These are essentially the normal equations for a weighted least-squares problem, with weights $\boldsymbol{W}$ and modified targets, $\boldsymbol{z}$. The parameter estimation procedure proceeds iteratively, alternating between updates of the model parameters, $\boldsymbol{\beta}$, via equation (18), and updates of $\boldsymbol{\eta}$, $\boldsymbol{\mu}$ and $\boldsymbol{z}$, via equations (1), (2) and (17) respectively. Hence the algorithm is also known as Iteratively Re-Weighted Least-Squares (IRWLS) [26].

II. Generalised Kernel Machines

A non-linear variant of the generalised linear model can be derived in an elegant manner via the "kernel trick", where the systematic component is constructed in a feature space, $\mathcal{F}$, given by a fixed transformation of the input space, i.e. $\boldsymbol{\varphi}(\boldsymbol{x}) : \mathcal{X} \rightarrow \mathcal{F}$. The systematic component is then given by

$$\eta_i = \boldsymbol{\beta} \cdot \boldsymbol{\varphi}(\boldsymbol{x}_i) + b, \tag{19}$$

note that we have introduced an explicit bias term, b. However, rather than specify the fixed transformation directly, it is implicitly defined by a Mercer kernel [23], $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, which gives the inner product between the images of the data in the feature space, i.e. $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\varphi}(\boldsymbol{x}) \cdot \boldsymbol{\varphi}(\boldsymbol{x}')$. The interpretation of the kernel as performing an inner product in a fixed feature space is valid for any kernel for which the Gram matrix,

$$\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j]_{i,j=1}^{\ell},$$

is positive definite [4]. A common kernel in practical applications is the radial basis function (RBF) or squared exponential kernel,

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{\kappa \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right\} \tag{20}$$

where κ is a *kernel parameter* governing the sensitivity of the kernel. In this case, the transformation $\boldsymbol{\varphi}$ maps the data onto the positive orthant of an infinite dimensional unit hypersphere. As the feature space, $\mathcal{F}$, is of infinite dimension, the systematic component of the generalised kernel model (19) becomes a universal approximator, capable of representing arbitrary relationships between the mean of the response distributions and the explanatory variables [24]. For an alternative treatment of kernel methods and exponential family distributions, from the perspective of machine learning rather than classical statistics, see Canu and Smola [8].

A. Parameter Estimation

Assuming the canonical link, such that $\eta_i = \theta_i$, the primal model parameters, $\boldsymbol{\beta}$ are determined using penalised maximum-likelihood, via minimisation of the criterion

$$L = \sum_{i=1}^{\ell} [y_i \eta_i - b(\eta)] + \frac{\lambda}{2} \|\boldsymbol{\beta}\|^2 \tag{21}$$

where λ is a regularisation parameter [37] controlling the bias-variance trade-off [15]. Fortunately, this represents a convex optimisation problem [7], with a unique global minimum. The representer theorem [19] indicates that the solution of this optimisation problem can be expressed as an expansion over the data of the form

$$\boldsymbol{\beta} = \sum_{j=1}^{\ell} \alpha_j \boldsymbol{\varphi}(\boldsymbol{x}_j) \quad \Longrightarrow \quad \eta_i = \sum_{i=1}^{\ell} \alpha_i \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_i) + b,$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ is a vector of *dual* model parameters. Again an iteratively re-weighted least-squares (IRWLS) procedure can be used. Using the method of Lagrange multipliers, the minimiser of the weighted least-squares problem is given by the solution of a simple system of linear equations,

$$\begin{bmatrix} \boldsymbol{K} + \lambda \boldsymbol{W}^{-1} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{z} \\ 0 \end{bmatrix}, \tag{22}$$

and the updates of $\boldsymbol{W}$ and $\boldsymbol{z}$ are exactly as before. This system of linear equations can be solved efficiently using the Cholesky decomposition [16, 35] of the matrix $\boldsymbol{K} + \lambda \boldsymbol{W}^{-1}$.

B. Model Selection

The generalised kernel machine introduces a small number of additional hyper-parameters, that must also be estimated from the data, the regularisation parameter, λ, and any kernel parameters, e.g. κ. The values of these hyper-parameters can be determined by minimising a cross-validation [34] estimate of the negative log-likelihood. The k-fold cross-validation strategy partitions the available data into k disjoint subsets. In each iteration, a model is constructed using a different combination of $k-1$ subsets, the remaining subset being used for performance estimation. The average of the performance estimates for the k models is the k-fold cross-validation estimate. The most extreme form of cross-validation, where each subset consists of a single pattern, is known as leave-one-out cross-validation [20], which has been shown to provide an almost unbiased estimate of performance on unseen data [21].

Leave-one-out cross-validation is computationally expensive, and so is generally impractical for use with all but the smallest datasets. However, in the case of least-squares linear regression, the leave-one-out procedure can be performed analytically as a by-product of fitting a model on the entire dataset (e.g. [1, 14, 38]). These methods can be adapted to provide an approximate leave-one-out cross-validation method for generalised kernel models, as they are based on iteratively re-weighted least-squares. Let $\boldsymbol{C}$ represent the matrix on the left hand side of the linear system (22),

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{K} + \lambda \boldsymbol{W}^{-1} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}$$

Furthermore, let us also assume that $\boldsymbol{W}$ and $\boldsymbol{z}$ remain approximately constant during each iteration of the leave-one-out procedure. It is then relatively straight-forward to show that [9, 10] that the output of the systematic component of a generalised kernel machine, for the i^{th} training pattern in the i^{th} fold of the leave-one-out process, can be approximated by,

$$\eta_i^{(-i)} \approx z_i - \frac{\alpha_i}{\boldsymbol{C}_{ii}^{-1}}. \tag{23}$$

This provides the basis for an efficient leave-one-out cross-validation estimate of the test likelihood, which can be used for model selection. The model selection criterion can be optimised using the Nelder-Mead simplex algorithm [27], or gradient based methods, e.g. scaled conjugate gradient descent [40] (c.f. [6, 13]). Model selection methods based on similar ideas have also been developed for the generalised linear models (e.g. [17, 22]).

III. Generalised Kernel Machine Toolbox

We have implemented an object-oriented MATLAB toolbox[1] implementing the generalised kernel machine. The toolbox provides:

- A number of simple kernel objects suitable for many basic applications, `@linear`, `@polynomial` and `@rbf`.
- The base class, `@gkm`, representing the functionality common to all generalised kernel machines.
- Concrete subclasses used to implement the examples detailed in this section, namely `@krr` - kernel ridge regression, `@klr` - kernel logistic regression and `@kpor` - kernel Poisson regression.
- Optimisation objects, `@simplex` and `@scg`, providing automatic model selection.
- An object representing a model selection criterion, `@aloo`, which corresponds to the approximate leave-one-out estimate of the test likelihood, as described in Section II-B.

The design of the toolbox is quite flexible and may be extended at a later date to provide additional functionality. The toolbox is also able to automatically generate new instances of the generalised kernel machine, using the MATLAB symbolic math toolbox to evaluate the loss function, the canonical link and the weighting function used in the iteratively re-weighted least squares procedure, from a string describing the canonical function, $b(\cdot)$. The `fix` method can the be used to save the new form of GKM to disk as a new concrete subclass of `@gkm`. A web-server is currently under construction to produce bespoke GKM classes for users without access to the symbolic math toolbox.

A. Example: Kernel Ridge Regression

A variety of kernel learning algorithms are exactly analogous to a generalised kernel machine taking for the random component, a homoscedastic Gaussian distribution (5). Noting that in this case, $b(\theta) = \theta^2/2$, and therefore the canonical link is the identity function, $g(\mu) = b'(\eta) = \eta$. Furthermore, $b''(\theta) = 1$, such that

$$\boldsymbol{W} = \mathrm{diag}(b''(\eta_1), b''(\eta_2), \ldots, b''(\eta_\ell)) = \boldsymbol{I}.$$

Similarly, as $\boldsymbol{\eta} = \boldsymbol{\mu}$, the modified targets are given by

$$\boldsymbol{z} = \boldsymbol{\eta} - \boldsymbol{W}^{-1}[\boldsymbol{y} - \boldsymbol{\mu}] = \boldsymbol{y}.$$

Therefore the system of linear equations giving the model parameters, $\boldsymbol{\alpha}$, simplifies to give,

$$\begin{bmatrix} \boldsymbol{K} + \lambda \boldsymbol{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}.$$

This is identical to the system of linear equations to be solved in fitting a kernel ridge regression (KRR) model [32], or equivalently the least-squares support vector machine (LS-SVM) [35, 36], regularisation network (RN) [29] and as Fishers' linear discriminant analysis is equivalent to least-squares regression to the class labels (e.g. [41]), the kernel Fisher discriminant (KFD) classifier [25]. A Bayesian treatment of this type of generalised kernel machine is equivalent to Gaussian process regression [30].

[1]We will shortly make the toolbox freely available under the terms of the GNU general public license.

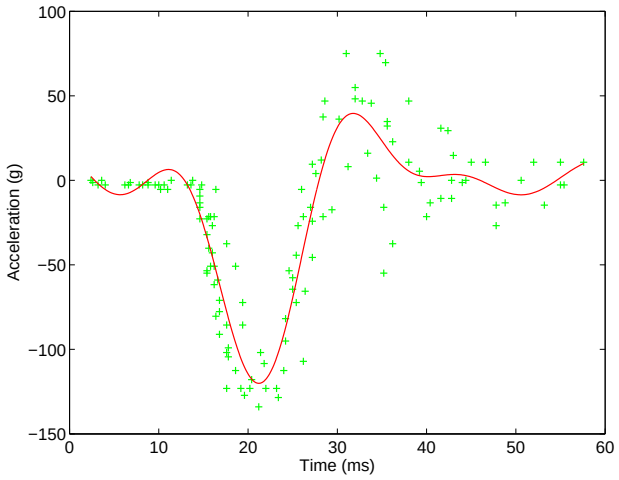

Fig. 1. Kernel ridge regression (KRR) model of Silverman's motorcycle benchmark dataset [33].

A new class, `@krr`, implementing the kernel ridge regression machine can be added to the MATLAB toolbox, using the following command:

```
fix(gkm('acronym', 'krr', ...
   'name', 'kernel ridge regression', ...
   'canonical', '0.5*eta^2'));
```

Here, we have specified an acronym, used to specify the name of the new class, and the full name of the model, for the purpose of displaying the model. The kernel ridge regression model was then used to model Silverman's motorcycle benchmark dataset [33], using a radial basis function kernel, as shown in Figure 1.

B. Example: Kernel Logistic Regression

Generalised linear models, and by extension generalised kernel machines can also be applied to statistical pattern recognition, where the target, y_i, indicates whether the i^{th} pattern belongs to the positive ($y_i = 1$) or negative class ($y_i = 0$) respectively. In this case the responses can be viewed as realisations of a series of Bernoulli trials, such that

$$f(y_i; \pi_i) = \pi_i^{y_i}(1 - \pi_i)^{1-y_i},$$

where π_i represents the probability that the i^{th} example belongs to the positive class (conditioned on the input vector, $\boldsymbol{x}_i$). The Bernoulli distribution can be written as a one-parameter member of the exponential family as

$$f(y; \theta) = \exp\left\{y\theta - \log\left[1 + \exp\left(\theta\right)\right]\right\},$$

where

$$\pi = \frac{\exp\{\theta\}}{1 + \exp\{\theta\}}.$$

In this case, the Bernoulli distribution is defined by the functions, $a(\phi) = 1$, $b(\theta) = \log[1+\exp(\theta)]$ and $c(y_i; \phi) = 0$.

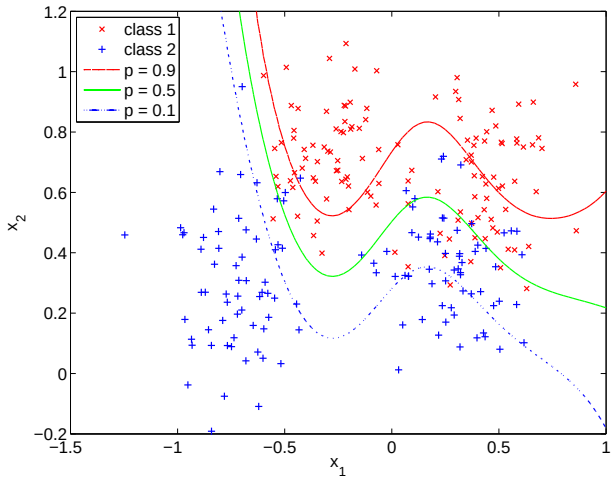

Fig. 2. Kernel logistic regression (KLR) model of Ripley's synthetic benchmark dataset [31].

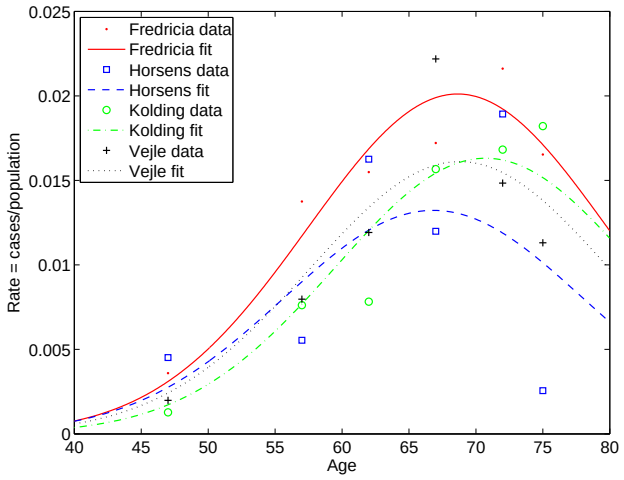

Fig. 3. Kernel Poisson regression (KPoR) model of Andersen's lung cancer benchmark dataset.

Note that the dispersion parameter, ϕ, is redundant as the Bernoulli distribution is completely specified by the mean, π. If we take the canonical link, such that $\theta = \eta$, we have that

$$\eta = g(\pi) = \text{logit}(\pi) = \log\left\{\frac{\pi}{1+\pi}\right\}.$$

As usual, the parameters of the model can be estimated by iteratively re-weighted least-squares, where

$$\boldsymbol{W} = \text{diag}(\pi_1(1-\pi_1), \pi_2(1-\pi_2), \ldots, \pi_\ell(1-\pi_\ell)),$$

and $\boldsymbol{z}$ is given by (17). The resulting model is known as kernel logistic regression (KLR) [11, 18]. The closely related kernel probit regression KPR method [5, 9] can be viewed as a GKM with a Bernoulli random component and the non-canonical probit link. A Bayesian treatment of this form of generalised kernel machine gives rise to the Gaussian process classifier [30, 39]. A new class implementing the kernel logistic regression algorithm can be saved to disk using the following command,

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

Figure 2 shows the results obtained using kernel logistic regression for Ripley's synthetic benchmark dataset [31].

C. Example: Kernel Poisson Regression

For the final example, we chose a more unusual model that does not appear to have an existing kernel variant, namely Poisson regression. The Poisson distribution arises naturally as the distribution of a random variable recording the number of occurrences of a rare event with a constant average rate, over a given period or population [3]. For example, Andersen [2] considers the incidence of lung cancer in the populations of four Danish cities, namely Fredricia, Horsens, Kolding and Vejla, within six different age groups ($40-54$, $55-59$, $60-64$, $65-69$, $70-74$ and >75). We therefore construct a generalised kernel machine, where the random component is given by a Poisson distribution,

$$f(y_i;\mu_i) = \frac{\exp\{-\mu_i\}\mu_i^{y_i}}{y!}$$

such that the canonical function is $b(\theta) = \exp\{\theta\}$, and so we obtain a logarithmic canonical link,

$$\eta_i = \log\{\mu_i\}$$

A class implementing kernel Poisson regression can be saved to disk using the following command:

```
fix(gkm('acronym', 'kpor', ...
   'name', 'kernel Poisson regression', ...
   'canonical', 'exp(eta)'));
```

The kernel Poisson regression model is then applied to Andersen's lung cancer dataset, with results shown in Figure 3. Here we adopt an inhomogeneous quadratic kernel,

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + c)^2$$

where c is a kernel parameter controlling the relative importance of first and second order terms. The model has four input features, the first represents the mid-point of the age range, and three binary variables that are set to one to indicate the city is Fredricia, Horsens or Kolding respectively (Vejla is indicated by all three of these inputs being set to zero).

D. Agnostic Learning versus Prior Knowledge Challenge

The examples given in this paper are intentionally small-scale, for the purpose of illustration, however, the toolbox has also been used to implement competitive solutions for the prior knowledge track of the IJCNN-2007 Agnostic Learning versus Prior Knowledge Challenge[2], using much larger datasets (current placings: ADA 1^{st}, GINA $= 1^{\text{st}}$, HIVA 1^{st}, NOVA 1^{st}, SYLVA $= 3^{\text{rd}}$). These examples are described in detail in a companion paper [12].

[2] http://www.agnostic.inf.ethz.ch/

IV. Conclusions

In this paper, we have described a common framework, uniting a wide variety of existing and novel kernel learning methods, viewing each as a non-linear variant of a particular generalised linear model based on the "kernel trick". This framework has been implemented in the form of a MATLAB toolbox supporting the creation of novel generalised kernel machines, with fully automated training and model selection procedures. The toolbox has also been used successfully to create methods for kernel survival analysis and modelling of extreme values. In relating this family of kernel learning methods to generalised linear models, we also inherit a vast body of theory, including deviance and goodness-of-fit, analysis of variance, asymptotic distributions for parameter estimates and confidence intervals on predictions. These ideas potentially have a great impact in the practical application of kernel learning methods, but are beyond the scope of this paper.

References

[1] D. M. Allen. The relationship between variable selection and prediction. *Technometrics*, 16:125–127, 1974.

[2] E. B. Andersen. Multiplicative Poisson models with unequal cell rates. *Scandinavian Journal of Statistics*, 4:153–158, 1977.

[3] P. K. Andersen, Borgan Ø., R. D. Gill, and N. Keilding. *Statistical models based on counting processes*. Springer Series in Statistics. Springer-Verlag, 1993.

[4] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[5] C. M. Bishop. *Pattern Recognition amd Machine Learning*. Springer, 2006.

[6] L. Bo, L. Wang, and L. Jiao. Feature scaling for kernel Fisher discriminant analysis using leave-one-out cross validation. *Neural Computation*, 18(4):961–978, April 2006.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[8] S. Canu and A. Smola. Kernel methods and the exponential family. *Neurocomputing*, 69:714–720, 2006.

[9] G. C. Cawley. Model selection for kernel probit regression. In *Proceedings of the European Symposium on Artificial Neural Networks* (submitted), Bruges, Belgium, April 25–27 2007.

[10] G. C. Cawley and N. L. C. Talbot. Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. *Pattern Recognition*, 36(11):2585–2592, November 2003.

[11] G. C. Cawley and N. L. C. Talbot. Efficient model selection for kernel logistic regression. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR-2004)*, volume 2, pages 439–442, Cambridge, United Kingdom, August 23–26 2004.

[12] 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-2007)* (submitted), Orlando, FL, USA, August12–17 2007.

[13] C. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[14] R. D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Monographs on Statistics and Applied Probability. Chapman and Hall, New York, 1982.

[15] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilema. *Neural Computation*, 4(1):1–58, 1992.

[16] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, third edition edition, 1996.

[17] P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models - A Roughness Penalty Approach*, volume 58 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, 1994.

[18] S. S. Keerthi, K. Duan, S. K. Shevade, and A. N. Poo. A fast dual algorithm for kernel logistic regression. In *Proceedings of the International Conference on Machine Learning*, 2002.

[19] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[20] P. A. Lachenbruch and M. R. Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10(1):1–11, February 1968.

[21] A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of reognition (in Russian). *Technicheskaya Kibernetica*, 3, 1969.

[22] P. McCullagh and J. A. Nelder. *Generalized Linear Models*, volume 37 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, second edition, 1989.

[23] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, A*, 209:415–446, 1909.

[24] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[25] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing*, volume IX, pages 41–48. IEEE Press, New York, 1999.

[26] I. T. Nabney. Efficient training of RBF networks for classification. *International Journal of Neural Systems*, 14(3):201–208, 2004.

[27] J. A. Nelder and R. Mead. A simplex method for function minimisation. *Computer Journal*, 7:308–313, 1965.

[28] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135:370–384, 1972.

[29] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, September 1990.

[30] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. MIT Press, 2006.

[31] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[32] C. Saunders, A. Gammermann, and V. Vovk. Ridge regression in dual variables. In J. Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-1998)*. Morgan Kaufmann, 1998.

[33] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric curve fitting. *Journal of the Royal Statistical Society, B*, 47:1–52, 1985.

[34] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, B*, 36(1):111–147, 1974.

[35] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vanderwalle. *Least Squares Support Vector Machines*. World Scientific Publishing, 2002.

[36] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, June 1990.

[37] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. John Wiley, New York, 1977.

[38] S. Weisberg. *Applied linear regression*. John Wiley and Sons, New York, second edition, 1985.

[39] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In *Neural Information Processing Systems 8*, pages 514–520. Morgan Kaufmann, 1996.

[40] P. M. Williams. A Marquardt algorithm for choosing the step-size in backpropagation learning with conjugate gradients. Cognitive Science Research Paper CSRP-229, University of Sussex, Brighton, U.K., February 1991.

[41] J. Xu, X. Zhang, and Y. Li. Kernel MSE algorithm: A unified framework for KFD, LS-SVM and KRR. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-2001)*, pages 1486–1491, Washington, DC, July 2001.