# Generalized Additive Models

Simon Wood

# Additive smooth models

- Given the approach to modelling smooth functions covered already, it is easy to work with *additive models* of the form

$$y_i = \alpha + \sum_j f_j(x_{ji}) + \epsilon_i$$

- The smooth functions, $f_j$, each get a basis and penalty but now require sum-to-zero identifiability constraints.
- The intercept, $\alpha$, can be replaced by parametric model terms and some of the covariates $x_j$ might be vector quantities.
- Inference methods are similar to those for single smooths, but
  1. the model matrix, $\mathbf{X}$, is now made up of the concatenated model matrices for each model term (smooth and parametric).
  2. the penalty matrix is now the sum of multiple penalty matrices, each multiplied by its own smoothing parameter.
- Vector $\boldsymbol{\lambda}$ makes efficient GCV/REML optimization challenging.

# Generalized Additive Models

- Another generalization relaxes the Gaussian response assumption, so that the model becomes

$$y_i \underset{ind.}{\sim} \text{EF}(\mu_i, \phi) \quad g(\mu_i) = \alpha + \sum_j f_j(x_{ji}) \ \ (\equiv \eta_i)$$

- $\text{EF}(\mu_i, \phi)$ denotes some exponential family distribution* with mean $\mu_i$ and scale parameter $\phi$. $\boldsymbol{\eta}$ is the *linear predictor*.
- $g$ is a known smooth monotonic *link function*.
- $\boldsymbol{\lambda}$ estimation requires GCV or REML criteria to be modified.
- The $\boldsymbol{\beta}$ estimation given $\boldsymbol{\lambda}$ is now a non-linear optimization and has to be done using Newton's method. Let's look at this first.

---

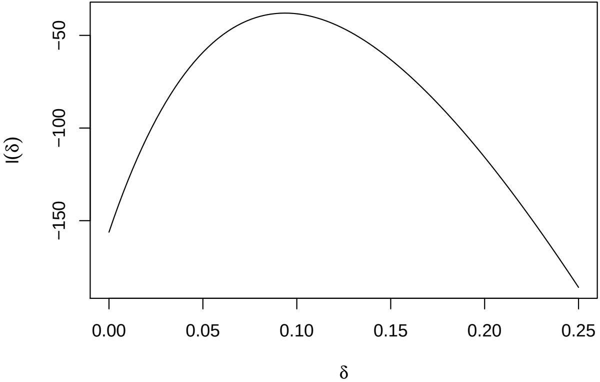
*e.g. Gaussian, Poisson, binomial, gamma, Tweedie etc.

# Newton's method: basic idea

- Newton's method is used to maximize or minimize smooth *objective functions*, such as quadratically penalized likelihoods, w.r.t. some parameters.
- We start with a guess of the parameter values.
- Then evaluate the function and its first and second derivatives w.r.t. the parameters at the guess.
- There is a unique quadratic function matching the value and derivatives, so we find that and optimize it to find the next guess at the optimizer of the objective.
- This derivative – quadratic approximation – maximize quadratic cycle is repeated to convergence.
- Convergence occurs when the first derivatives are zero[†].

[†]The (negative) *Hessian* matrix of second derivatives should be positive definite at a minimum (maximum).

# Newton's method illustrated in one dimension



# Newton's method in more detail

- Consider minimizing $D(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$. Taylor's theorem says

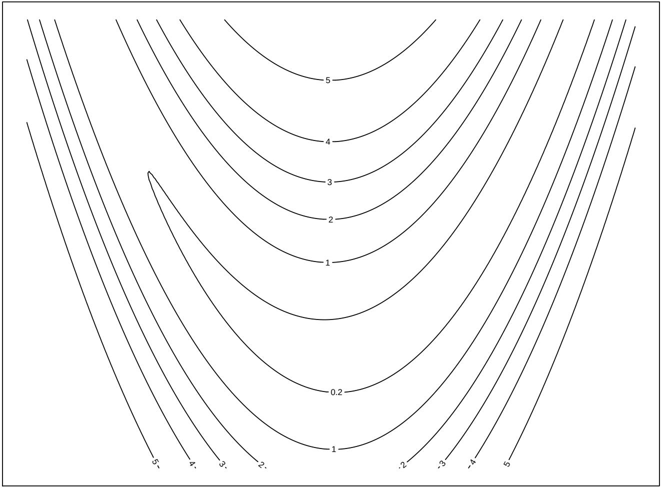
$$D(\boldsymbol{\theta}+\boldsymbol{\Delta}) = D(\boldsymbol{\theta}) + \boldsymbol{\Delta}^{\mathsf{T}}\nabla_\theta D + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\nabla_\theta^2 D\boldsymbol{\Delta} + o(\|\boldsymbol{\Delta}\|^2)$$

- Provided $\nabla_\theta^2 D$ is positive definite, the $\boldsymbol{\Delta}$ minimizing the quadratic on the right is

$$\boldsymbol{\Delta} = -(\nabla_\theta^2 D)^{-1}\nabla_\theta D$$

- This also minimizes $D$ in the small $\boldsymbol{\Delta}$ limit, which is the one that applies near $D$'s minimum.
- Interestingly, $\boldsymbol{\Delta}$ is still a descent direction with *any positive definite matrix* in place of the Hessian $\nabla_\theta^2 D$.
- So if $\nabla_\theta^2 D$ is not positive definite we just perturb it to be so.
- Far from the optimum $\boldsymbol{\Delta}$ might overshoot. If so, repeatedly halve $\boldsymbol{\Delta}$ until $D(\boldsymbol{\theta}+\boldsymbol{\Delta}) < D(\boldsymbol{\theta})$ to guarantee convergence,

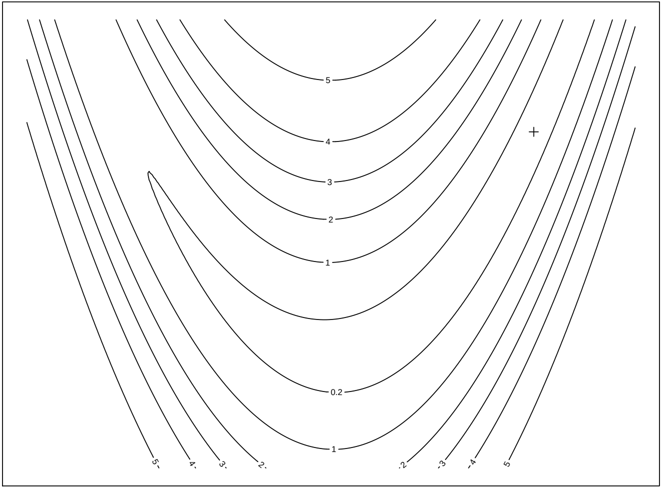
# Newton in 2D with Hessian perturbation and step halving



# Why Newton?

- Why not not simplify and use a first order Taylor expansion in place of the Newton method's second order expansion?
- Doing so gives the method of *steepest descent* and two problems
  1. As we approach the optimum the first derivative of the objective vanishes, so that there is ever less justification for dropping the second derivative term.
  2. Without second derivative information we have nothing to say how long the step should be.
- In practice 1. leads to steepest descent often requiring huge numbers of steps as the optimum is approached.
- To use only first derivatives check out *quasi-Newton* methods.
- What about co-ordinate descent, that worked well for the Lasso?
- This can also take forever for some problems.
- For the previous example, Newton takes 20 steps and co-ordinate descent over 4000 (for reduced accuracy). Here are the first 20...

# First 20 coordinate descent steps



# Computing $\hat{\boldsymbol{\beta}}$ and $\pi(\boldsymbol{\beta}|\mathbf{y})$

- Let $l(\boldsymbol{\beta}) \equiv \log \pi(\mathbf{y}|\boldsymbol{\beta})$ and $\mathbf{S}_\lambda$ be the combined penalty matrix.
- $\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\, l(\boldsymbol{\beta}) - \boldsymbol{\beta}^\mathsf{T}\mathbf{S}_\lambda\boldsymbol{\beta}/2 \Rightarrow \left.\frac{\partial l}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} - \mathbf{S}_\lambda\hat{\boldsymbol{\beta}} = \mathbf{0}$
- Optimized using Newton iteration (until $\hat{\boldsymbol{\beta}}$ converged):
  $\hat{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}} + (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1}\left(\left.\frac{\partial l}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} - \mathbf{S}_\lambda\hat{\boldsymbol{\beta}}\right)$, where $\hat{\boldsymbol{\mathcal{I}}} = -\frac{\partial^2 l}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^\mathsf{T}}$.
- Taylor expand about $\hat{\boldsymbol{\beta}}$ for approximate posterior

$$\begin{aligned}\log \pi(\boldsymbol{\beta}|\mathbf{y}) &= l(\boldsymbol{\beta}) - \boldsymbol{\beta}^\mathsf{T}\mathbf{S}_\lambda\boldsymbol{\beta}/2 + c \\ &\simeq l(\hat{\boldsymbol{\beta}}) - \frac{1}{2}\hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}_\lambda\hat{\boldsymbol{\beta}} - \frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^\mathsf{T}(\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + c\end{aligned}$$

- Hence[‡] approximately $\pi_G(\boldsymbol{\beta}|\mathbf{y}) \propto e^{-\frac{1}{2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^\mathsf{T}(\hat{\boldsymbol{\mathcal{I}}}+\mathbf{S}_\lambda)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}$, so

$$\boldsymbol{\beta}|\mathbf{y} \sim \mathrm{N}(\hat{\boldsymbol{\beta}}, (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1})$$

[‡] generally requires $\dim(\boldsymbol{\beta}) = o(n^{1/3})$

# Smoothing parameter selection

- Marginal likelihood $\pi(\mathbf{y}|\boldsymbol{\beta}) = \int \pi(\mathbf{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}|\boldsymbol{\lambda})d\boldsymbol{\beta}$ is intractable.
- But we can re-use the Gaussian approximate posterior, $\pi_G$

$$\pi(\mathbf{y}|\boldsymbol{\lambda}) = \frac{\pi(\mathbf{y}|\hat{\boldsymbol{\beta}})\pi(\hat{\boldsymbol{\beta}}|\boldsymbol{\lambda})}{\pi(\hat{\boldsymbol{\beta}}|\mathbf{y})} \simeq \frac{\pi(\mathbf{y}|\hat{\boldsymbol{\beta}})\pi(\hat{\boldsymbol{\beta}})|\boldsymbol{\lambda})}{\pi_G(\hat{\boldsymbol{\beta}}|\mathbf{y})}$$

- This is tractable and is also equivalent to replacing the log of the ML integrand with its second order Taylor expansion about $\hat{\boldsymbol{\beta}}$ and integrating the tractable result: *Laplace Approximation*.

$$2\log\pi(\mathbf{y}|\boldsymbol{\lambda}) \simeq 2l(\hat{\boldsymbol{\beta}}) - \hat{\boldsymbol{\beta}}^{\mathsf{T}}\mathbf{S}_\lambda\hat{\boldsymbol{\beta}} + \log|\mathbf{S}_\lambda|_+ - \log|\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda| + c$$

- Proceeding as in the Gaussian case:

$$\text{EDF} = \text{trace}\{(\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1}\hat{\boldsymbol{\mathcal{I}}}\}$$

# The penalized least squares link

- The above theory was not tied to exponential families, but in the EF case $\text{var}(y_i) = V(\mu_i)\phi$, and $V$ is known for each distribution.
- Let $\alpha(\mu_i) = 1 + (y_i - \mu_i)(V'(\mu_i)/V(\mu_i) + g''(\mu_i)/g'(\mu_i))$ and $w_i = \alpha(\mu_i)V(\mu_i)^{-1}g'(\mu_i)^{-2}$ (and note that $\mathbb{E}(\alpha) = 1$).
- The Hessian of the negative log likelihood $\hat{\boldsymbol{\mathcal{I}}} = \mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}$ where $\mathbf{W}$ is diagonal and $\mathbf{W}_{ii} = w_i$.
- Defining $z_i = g'(\mu_i)(y_i - \mu_i)/\alpha(\mu_i) + \eta_i$ Newton's method is identical to *Penalized Iteratively Re-weighted Least Squares*[§]…
  1. Set $\hat{\mu}_i = y_i + \iota_i$ and iterate 2 and 3 to convergence.
  2. Compute $z_i$ and $w_i$ from the current $\hat{\eta}_i$ and $\hat{\mu}_i = g^{-1}(\hat{\eta}_i)$.
  3. Find $\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \ \|\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\|_W + \boldsymbol{\beta}^\mathsf{T}\mathbf{S}_\lambda\boldsymbol{\beta}$ and $\hat{\boldsymbol{\eta}} = \mathbf{X}\hat{\boldsymbol{\beta}}$.
- Replacing $w_i$ with $E(w_i)$ is known as *Fisher Scoring*.
- A simple approach estimates $\boldsymbol{\lambda}$ for each the working model.

[§] $\iota_i$ is usually zero, but may be a small constant ensuring finite $\hat{\eta}_i$. $\|\mathbf{V}\|_W^2 = \mathbf{v}^\mathsf{T}\mathbf{W}\mathbf{v}$.

# Deviance based GCV

- For exponential family GAM/GLM there is a generalization of the residual sum of squares, know as the *deviance*:

$$D(\boldsymbol{\beta}) = 2(l_s - l(\boldsymbol{\beta}))\phi$$

  where $l_s$ is the saturated likelihood — the highest value the likelihood could take if there was a parameter for each $y_i$.
- For Gaussian data the deviance *is* the residual sum of squares.
- The GCV criterion then generalizes to

$$\text{GCV} = nD(\hat{\boldsymbol{\beta}})/(n - \text{EDF})^2.$$

# Nested optimization for $\hat{\lambda}$ and implicit differentiation

- ML or GCV are optimized w.r.t. $\boldsymbol{\rho} = \log \boldsymbol{\lambda}$ by Newton's method.
- Each trial $\boldsymbol{\rho}$ vector proposed by Newton's method requires an inner Newton iteration for the corresponding $\hat{\boldsymbol{\beta}}$, plus evaluation of the gradient and Hessian of the ML or GCV criterion.
- These derivatives in turn require derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$.

By definition of $\hat{\boldsymbol{\beta}}$, $$\left.\frac{\partial l}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} - \mathbf{S}_\lambda \hat{\boldsymbol{\beta}} = \mathbf{0}$$

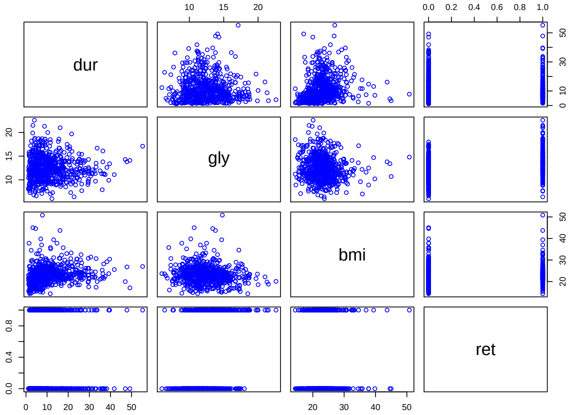
- Noting that $\mathbf{S}_\lambda = \sum_j \lambda_j \mathbf{S}_j$ and differentiating w.r.t. $\rho_j$

$$\left.\frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\mathsf{T}}\right|_{\hat{\boldsymbol{\beta}}} \frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} - \lambda_j \mathbf{S}_j \hat{\boldsymbol{\beta}} - \mathbf{S}_\lambda \frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} = \mathbf{0} \Rightarrow \frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} = -\lambda_j (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1} \mathbf{S}_j \hat{\boldsymbol{\beta}}.$$

- 2nd derivs follow similarly. Criterion derivs are then routine.

# Example: diabetic retinopathy

- The `wesdr` data[¶] look at the relationship between development of retinopathy, duration of disease, BMI and percentage glycocylated haemoglobin in a cohort of diabetics.




---

[¶]see Chong Gu's `gss` package in R

# A retinopathy model

- A possible model for these data is $\texttt{ret}_i \sim \text{bin}(1, \mu_i)$

$$\text{logit}(\mu_i) = \alpha + f_1(\texttt{dur}_i) + f_2(\texttt{gly}_i) + f_3(\texttt{bmi}_i)$$
$$+ f_4(\texttt{dur}_i, \texttt{gly}_i) + f_5(\texttt{dur}_i, \texttt{bmi}_i) + f_6(\texttt{gly}_i, \texttt{bmi}_i)$$

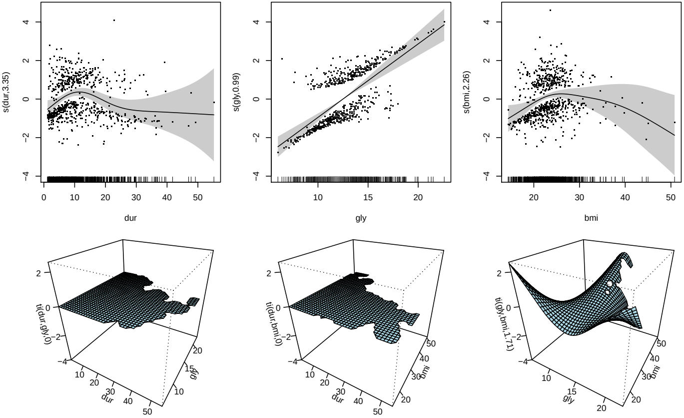
  where $\text{logit}(\mu) = \log\{\mu/(1-\mu)\}$.
- The model can be estimated using `gam` from R package `mgcv`:

```
k <- 7 ## choosing basis size
b <- gam(ret~s(dur,k=k)+s(gly,k=k)+s(bmi,k=k)+
     ti(dur,gly,k=k)+ti(dur,bmi,k=k)+ti(gly,bmi,k=k),
     select=TRUE,data=wesdr,family=binomial,method="REML")
```

- `ti` are tensor product smooths with main effects excluded as covered previously.
- `select=TRUE` adds a penalty for each smooth, so that it can be penalized to zero. Consider the eigen decomposition of a penalty matrix $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$. Let $\mathbf{U}_0$ be the cols of $\mathbf{U}$ with corresponding eigenvalues 0. $\mathbf{S}_0 = \mathbf{U}_0\mathbf{U}_0^\mathsf{T}$ is a penalty on the null space of $\mathbf{S}$.

# Retinopathy results

- Using `plot(b,scheme=1,...)` we see that there is a non-zero interaction between `gly` and `bmi`.



# Retinopathy summary

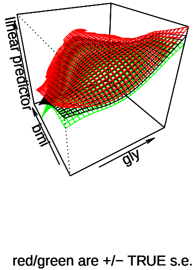
```
> summary(b)

Family: binomial
Link function: logit

Formula:
ret ~ s(dur, k = k) + s(gly, k = k) + s(bmi, k = k) + ti(dur,
    gly, k = k) + ti(dur, bmi, k = k) + ti(gly, bmi, k = k)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.40366    0.08979  -4.496 6.93e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                  edf Ref.df Chi.sq p-value
s(dur)      3.347e+00      6 15.092 0.00103 **
s(gly)      9.892e-01      6 87.169 < 2e-16 ***
s(bmi)      2.263e+00      6 11.724 0.00138 **
ti(dur,gly) 2.539e-04     36  0.000 0.64886
ti(dur,bmi) 8.409e-05     36  0.000 0.61919
ti(gly,bmi) 1.706e+00     35  7.505 0.00581 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.221   Deviance explained = 18.4%
-REML = 387.27  Scale est. = 1         n = 669
```

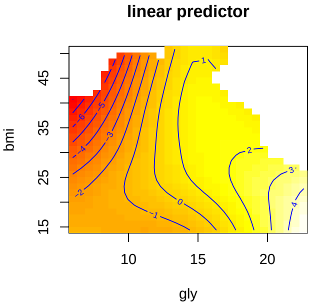
…the interaction seems to be ‘significant’.

# Retinopathy interpretation

- So the duration effect can be interpreted alone — steadily increasing risk for the first decade, then a decline — this may be an age or 'harvesting effect' the long duration individuals being those with good disease control.
- For the interaction we need to look at the combined effect. e.g.

```
vis.gam(b,view=c("gly","bmi"),se=T,phi=30,theta=-30,too.far=.15)
vis.gam(b,view=c("gly","bmi"),plot.type="contour",too.far=.15)
```



**Temporary page!**

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to th final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page wi go away, because LaTeX now knows how many pages to expect for document.