Bayesian Kernel Methods
Sparse Greedy Approximations

Alexander J. Smola
Machine Learning Group, RSISE
The Australian National University
Canberra, ACT 0200
Alex.Smola@anu.edu.au

Slides available at http://mlg.anu.edu.au/∼smola/icml2002/

A Simple Implementation

Idea

Minimize the negative log-likelihood with the Newton method.

Basic Algorithm

To minimize a function $\mathfrak{L}(f)$ which is twice differentiable in f approximate

$$\mathfrak{L}(f + \Delta f) \approx \mathfrak{L}(f) + \Delta f \mathfrak{L}'(f) + \frac{1}{2}\Delta f^\top \mathfrak{L}''(f)\Delta f$$

Hence we may approximately compute the minimum via

$$f \leftarrow f - (\mathfrak{L}''(f))^{-1}\mathfrak{L}'(f)$$

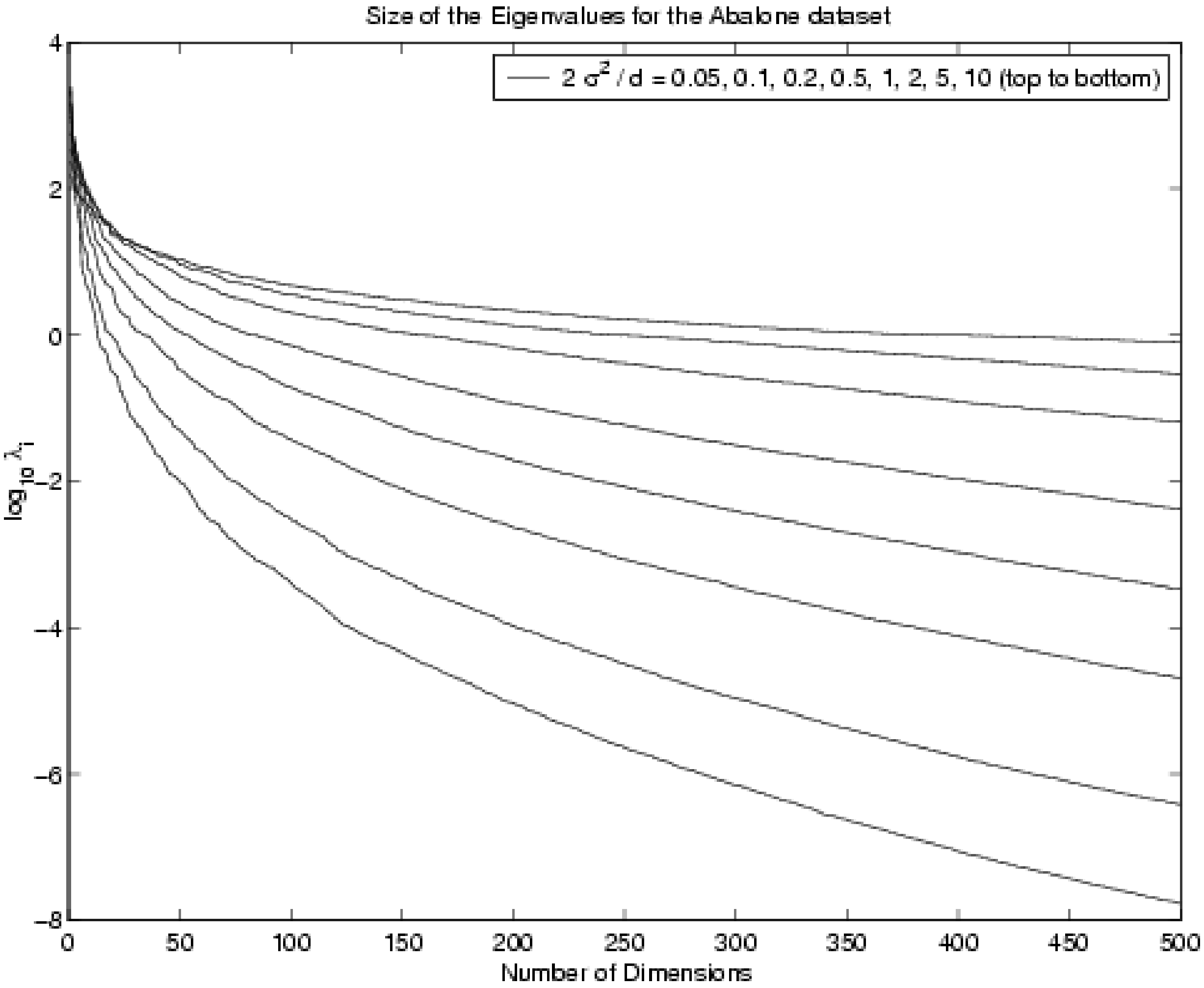
Practical Consequence

From $\mathfrak{L}(f) = \sum_{i=1}^m - \log p(y_i|[K\alpha]_i, x_i) + \frac{1}{2}\alpha^\top K\alpha$ (with the usual parameterization $f = K\alpha$) we obtain

$$\alpha \leftarrow \alpha - (K + K^\top C'' K)^{-1} K c'$$

where $c'_i = \partial^1_{[K\alpha]_i} - \log p(y_i|[K\alpha]_i, x_i)$ and $C''_{ii} = \partial^2_{[K\alpha]_i} - \log p(y_i|[K\alpha]_i, x_i)$.

Spectrum of Covariance Matrix



Practical Consequences

Ill conditioned matrix

Inverting K or products thereof is a numerically unstable procedure.

Observation

Removing the smallest eigenvalues/eigenvectors, we obtain almost the same solution.

Computational Speed

Smaller matrices mean that we can solve each Newton step more efficiently (in a nutshell, from $O(m^3)$ cost we go to $O(mn^2)$)

Prediction

If we **could** compute the functions corresponding to the eigensystem of K directly, this **would** speed prediction up from $O(m)$ to $O(n)$.

Plan

Replace the PCA with something more efficient, where we only need to compute n covariance functions $k(x_i, \cdot)$.

Recall: Gaussian Process Regression

Goal

Find distribution of y at location x (i.e. **mean** and **variance** of the normal distribution) by integrating out the normal distribution in the rest.

Solution: Denote by $\mathbf{k} = (k(x_1, x), \ldots, k(x_m, x))$. Then we have

$$\boxed{\mathbf{E}[y] = \mathbf{k}^\top (K + \sigma^2 \mathbf{1})^{-1} \mathbf{y}} \quad \text{and} \quad \boxed{\mathrm{Var}[y] = k(x, x) + \sigma^2 - \mathbf{k}^\top (K + \sigma^2 \mathbf{1})^{-1} \mathbf{k}}$$

Modified Solution

If we have to predict at several points it pays to compute $\alpha^* := (K + \sigma^2 \mathbf{1})^{-1} \mathbf{y}$ and predict the mean of y by $\mathbf{k}^\top \alpha$.

Idea: Find α and $\mathbf{k}^\top (K + \sigma^2 \mathbf{1})^{-1} \mathbf{k}$ by minimizing quadratic forms:

$$\begin{aligned}
\alpha^* &= \operatorname*{argmin}_{\alpha} \left[-\mathbf{y}^\top K \alpha + \frac{1}{2} \alpha^\top (K^\top K + \sigma^2 K) \alpha \right] \\
\mathbf{k}^\top (K + \sigma^2 \mathbf{1})^{-1} \mathbf{k} &= 2 \cdot \min_{\alpha} \left[-\mathbf{k}^\top \alpha + \frac{1}{2} \alpha^\top (K + \sigma^2 \mathbf{1}) \alpha \right]
\end{aligned}$$

Approximating Quadratic Forms

Theorem

Denote by $K \in \mathbb{R}^{m \times m}$ a positive semidefinite matrix, $\mathbf{y}, \alpha \in \mathbb{R}^m$ and define the two quadratic forms

$$\begin{aligned} Q(\alpha) &:= -\mathbf{y}^\top K\alpha + \frac{1}{2}\alpha^\top(\sigma^2 K + K^\top K)\alpha, \\ Q^*(\alpha) &:= -\mathbf{y}^\top \alpha + \frac{1}{2}\alpha^\top(\sigma^2 \mathbf{1} + K)\alpha. \end{aligned}$$

Suppose Q and Q^* have minima $Q_{\min}$ and $Q^*_{\min}$. Then for all $\alpha, \alpha^* \in \mathbb{R}^m$

$$\begin{aligned} Q(\alpha) \geq \ Q_{\min} &\geq -\frac{1}{2}\|\mathbf{y}\|^2 - \sigma^2 Q^*(\alpha^*), \\ Q^*(\alpha^*) \geq \ Q^*_{\min} &\geq \sigma^{-2}\left(-\frac{1}{2}\|\mathbf{y}\|^2 - Q(\alpha)\right), \end{aligned}$$

with equalities throughout when $Q(\alpha) = Q_{\min}$ and $Q^*(\alpha^*) = Q^*_{\min}$.

Proof

Minimum of $Q(\alpha)$

The minimum of $Q(\alpha)$ is obtained for $\alpha_{\mathrm{opt}} = (K + \sigma^2 \mathbf{1})^{-1}\mathbf{y}$ (which also minimizes Q^*), hence

$$Q_{\min} = -\frac{1}{2}\mathbf{y}^\top K(K + \sigma^2 \mathbf{1})^{-1}\mathbf{y} \text{ and } Q^*_{\min} = -\frac{1}{2}\mathbf{y}^\top (K + \sigma^2 \mathbf{1})^{-1}\mathbf{y}.$$

Combining Q and Q^*

This allows us to combine the minima to

$$Q_{\min} + \sigma^2 Q^*_{\min} = -\frac{1}{2}\|\mathbf{y}\|^2.$$

Minimum Property of Q, Q^*

Since by definition $Q(\alpha) \geq Q_{\min}$ for all α (and likewise $Q^*(\alpha^*) \geq Q^*_{\min}$ for all α^*), we may solve $Q_{\min} + \sigma^2 Q^*_{\min}$ for either Q or Q^* to obtain lower bounds for each of the two quantities.

Decomposition and Update

Recall: Objective Functions

$$\begin{aligned} Q(\alpha) &:= -\mathbf{y}^\top K\alpha + \frac{1}{2}\alpha^\top(\sigma^2 K + K^\top K)\alpha, \\ Q^*(\alpha) &:= -\mathbf{y}^\top\alpha + \frac{1}{2}\alpha^\top(\sigma^2\mathbf{1} + K)\alpha. \end{aligned}$$

Ansatz

Use $P \in \mathbb{R}^{m\times n}$ (as an **extension** matrix) to approximate α by $P\beta$. In particular, P contains only one nonzero entry per column.

Optimal solution in β

$$\begin{aligned} \beta_{\text{opt}} &= \left(P^\top\left(\sigma^2 K + K^\top K\right)P\right)^{-1} P^\top K^\top \mathbf{y} \\ \beta^*_{\text{opt}} &= \left(P^\top\left(\sigma^2\mathbf{1} + K\right)P\right)^{-1} P^\top \mathbf{k} \end{aligned}$$

Decomposition and Update

Idea

We can obtain the inverse matrices by a rank 1 update at $O(mn)$ cost if we know the inverse for P_{old} where $P = [P_{\text{old}}, \mathbf{e}_j])$.

$$P^\top K^\top \mathbf{y} = [P_{\text{old}}, \mathbf{e}_i]^\top K^\top \mathbf{y} = (P_{\text{old}}^\top K^\top \mathbf{y}, \mathbf{k}_i^\top \mathbf{y})$$

$$P^\top \left(K^\top K + \sigma^2 K\right) P = \begin{bmatrix} P_{\text{old}}^\top \left(K^\top K + \sigma^2 K\right) P_{\text{old}} & P_{\text{old}}^\top \left(K^\top + \sigma^2 \mathbf{1}\right) \mathbf{k}_i \\ \mathbf{k}_i^\top (K + \sigma^2 \mathbf{1}) P_{\text{old}} & \mathbf{k}_i^\top \mathbf{k}_i + \sigma^2 K_{ii} \end{bmatrix}$$

Strategy

Try out several new randomly chosen basis functions at each iteration and pick the one which minimizes the objective function most.

Performance Guarantee

With high probability we will find one of the best basis functions (e.g., with a subset of 59 we'll get a 95% guarantee).

Why do random subsets work?

Theorem

Given a random variable ξ with cumulative distribution function $F(\xi)$, then for n instances $\xi_1, \ldots \xi_m$ of ξ and $\xi_i \sim \partial_\xi F(\xi)$

$$\zeta := \max\{\xi_1, \ldots, \xi_n\} \text{ we have } F(\zeta) = F^n(\xi).$$

Corollary

The cumulative distribution of percentiles χ (i.e. fraction of samples larger than χ) for ζ is bounded from below by $F(\chi) = \chi^n$.

Practical Consequence

We only need at most $\left\lceil \frac{\log \delta}{\log(1-\eta)} \right\rceil$ samples in order to obtain a sample among the best δ with $1 - \eta$ confidence.

In particular 59 samples suffice to obtain with 95% probability a sample that is better than 95% of the rest.

Comparison with Other Methods

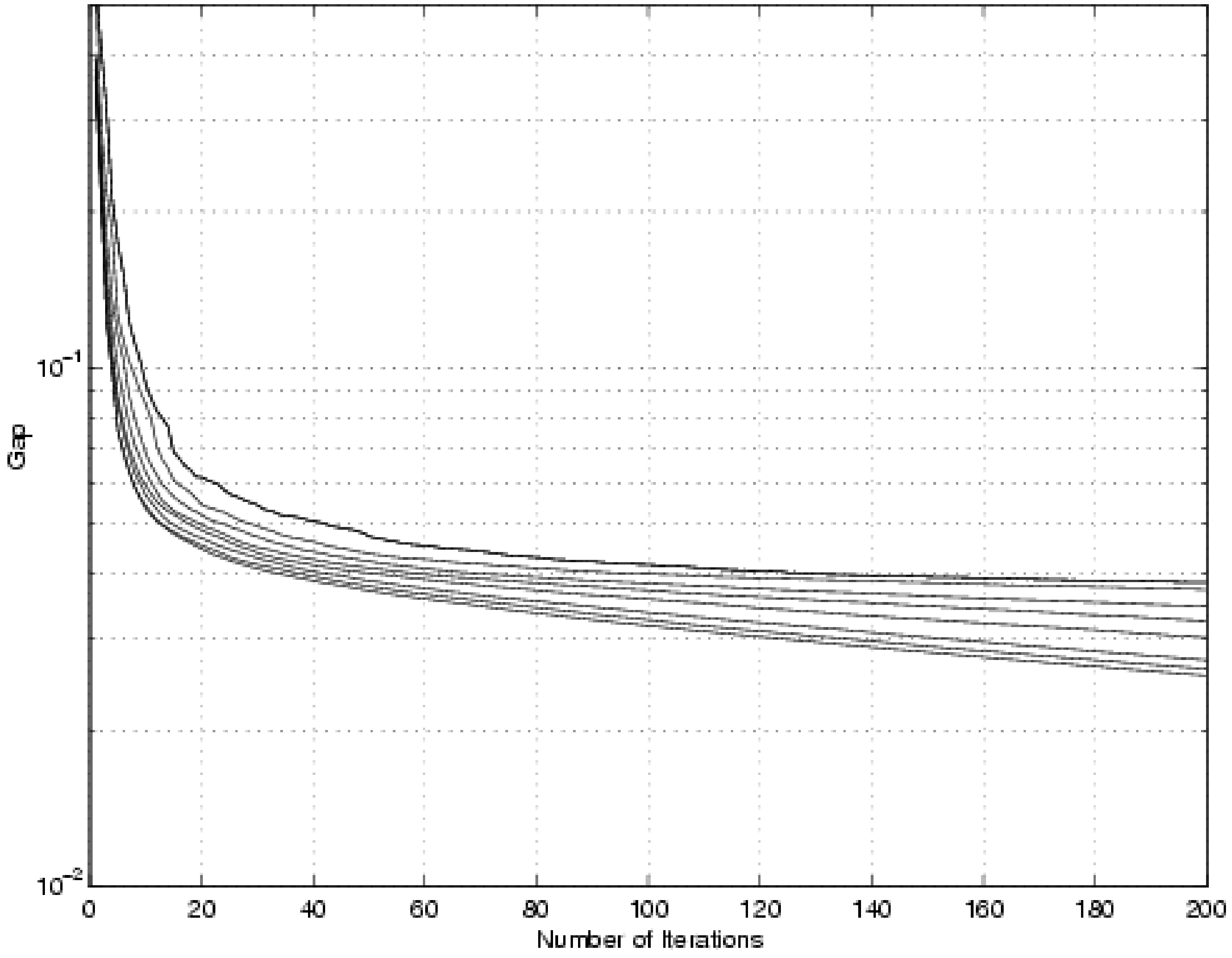
	Exact Solution	Conjugate Gradient	Sparse Decomposition	Sparse Greedy Approximation
Memory	$O(m^2)$	$O(m^2)$	$O(nm)$	$O(nm)$
Initialization	$O(m^3)$	$O(nm^2)$	$O(n^2m)$	$O(\kappa n^2 m)$
Prediction:				
Mean	$O(m)$	$O(m)$	$O(n)$	$O(n)$
Error Bars	$O(m^2)$	$O(nm^2)$	$O(n^2m)$ or $O(n^2)$	$O(\kappa n^2 m)$ or $O(n^2)$

Optimal Rate

The sparse decomposition rates would be optimal but can only be obtained after an NP hard search for the best basis.

Note that $n \ll m$ and that the n used in CG, SD, and SGA methods will differ, with $n_{\mathrm{CG}} \leq n_{\mathrm{SD}} \leq n_{\mathrm{SGA}}$ since the search spaces are more restricted.

Speed of Convergence



Size of the gap between upper and lower bound of the log posterior, i.e. $Q(\alpha)$ for the first 4000 samples from the Abalone dataset. From top to bottom: subsets of size 1, 2, 5, 10, 20, 50, 100, 200.

Basis Functions and Performance

Generalization Performance of Greedy Gaussian Processes

	Generalization Error	Log Posterior
Optimal Solution	1.782 ± 0.33	$-1.571 \cdot 10^5(1 \pm 0.005)$
Sparse Greedy Approximation	1.785 ± 0.32	$-1.572 \cdot 10^5(1 \pm 0.005)$

Kernels needed to minimize the log posterior, depending on the width of the Gaussian kernel ω. Also, number of basis functions required to approximate $\mathbf{k}^\top(K + \sigma^2\mathbf{1})^{-1}\mathbf{k}$ which is needed to compute the error bars.

Kernel width $2\omega^2$	1	2	5	10	20	50
Kernels for log-posterior	373	287	255	257	251	270
Kernels for error bars	79±61	49±43	26±27	17±16	12±9	8±5

Projections on Subspace

Basic Idea

Even for arbitrary posteriors, using only a subset of coefficients, i.e., $P\beta$ instead of α, will allow us to find rather good approximations. We then minimize

$$-\log \mathfrak{L}(P\beta, X, Y) = \sum_{i=1}^{m} -\log p(y_i | x_i, [KP\beta]_i) + \frac{1}{2}\beta^\top P^\top K P\beta$$

Now we can minimize a smaller optimization problem which costs $O(mn^2)$ (details on this later).

Parameter Transformation

We now switch to a parameter space in which the GP prior will become **diagonal**.

Without loss of generality assume that P picks the first n coefficients: $P = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$.

Note: in numerical mathematics this process arises from Gauss elimination of the the rows of the covariance matrix .

Projections on Subspace, Part II

Gauss Elimination

Transform $K = \begin{bmatrix} K^{nn} & K^{mn} \\ (K^{mn})^\top & K^{mm} \end{bmatrix}$ into $\tilde{K} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & K^{mm} - (K^{mn})^\top (K^{nn})^{-1} K^{mn} \end{bmatrix}$ by $\begin{bmatrix} (K^{nn})^{-\frac{1}{2}} & -(K^{nn})^{-1} K^{mn} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$.

The term $\tilde{K} := K^{mm} - (K^{mn})^\top (K^{nn})^{-1} K^{mn}$ is often referred to as the Schur complement.

Terms of the Optimization Problem

Reparameterizing by $\alpha = \begin{bmatrix} (K^{nn})^{-\frac{1}{2}} & -(K^{nn})^{-1} K^{mn} \\ & \mathbf{1} \end{bmatrix} \begin{bmatrix} \beta_n \\ \beta_m \end{bmatrix}$ yields

$$\alpha^\top K \alpha \longrightarrow \|\beta_n\|^2 + \beta_m^\top \tilde{K} \beta_m \text{ and } K\alpha \longrightarrow \begin{bmatrix} (K^{nn})^{\frac{1}{2}} \\ K^{mn}(K^{nn})^{-\frac{1}{2}} \end{bmatrix} \beta_n + \begin{bmatrix} \mathbf{0} \\ \tilde{K} \end{bmatrix} \beta_m$$

Projections on Subspace, Part III

Gradients of Log-Posterior

$$\begin{aligned}
\partial_{\beta_n} - \log \mathfrak{L} &= \begin{bmatrix} (K^{nn})^{\frac{1}{2}} \\ K^{mn}(K^{nn})^{-\frac{1}{2}} \end{bmatrix} \mathbf{c}' + \beta_n \\
\partial_{\beta_m} - \log \mathfrak{L} &= \begin{bmatrix} \mathbf{0} \\ \tilde{K} \end{bmatrix} \mathbf{c}' + \tilde{K}\beta_m
\end{aligned}$$

Hessian

$$\begin{aligned}
\partial^2_{\beta_n} - \log \mathfrak{L} &= \begin{bmatrix} (K^{nn})^{\frac{1}{2}} \\ K^{mn}(K^{nn})^{-\frac{1}{2}} \end{bmatrix}^\top \mathbf{c}'' \begin{bmatrix} (K^{nn})^{\frac{1}{2}} \\ K^{mn}(K^{nn})^{-\frac{1}{2}} \end{bmatrix} + \mathbf{1} \\
\partial^2_{\beta_m} - \log \mathfrak{L} &= \begin{bmatrix} \mathbf{0} \\ \tilde{K} \end{bmatrix}^\top \mathbf{c}'' \begin{bmatrix} \mathbf{0} \\ \tilde{K} \end{bmatrix} + \tilde{K}
\end{aligned}$$

where $c_i = -\log p(y_i|x_i, f(x_i)$ and the derivatives are taken wrt. $f(x_i)$.

Newton Method

Recall

We have updates $f \leftarrow f - (\mathfrak{L}''(f))^{-1}\mathfrak{L}'(f)$.

Updates in β_n

To optimize over the subspace spanned by the first n covariance functions, we only need to compute

$$\beta_n \leftarrow \beta_n - (Z\mathbf{c}''Z^\top)^{-1}(Z\mathbf{c}' + \beta_n) \text{ where } Z := \begin{bmatrix} (K^{nn})^{\frac{1}{2}} \\ K^{mn}(K^{nn})^{-\frac{1}{2}} \end{bmatrix}.$$

Computational Cost

Storage requirement is $O(mn)$ for Z and $O(n^2)$ for K^{nn}. CPU cost per inversion is $O(mn^2)$ to compute $(Z\mathbf{c}''Z^\top)$, plus $O(n^3)$ for the inversion. That is, if the space is spanned by a small number of basis functions, the estimation process is **linear** in the number of observations.

A Gradient Lemma

Problem

We need to know when to stop the optimization. For this purpose we use a bound in terms of the gradient of the log likelihood.

Lemma

Denote by $\mathcal{P}(\beta)$ a differentiable convex functions with $\mathcal{P}(\beta) = \mathcal{L}(\beta) + \frac{1}{2}\beta^\top M \beta$. Then we have

$$\min_\beta \mathcal{P}(\beta) \geq \mathcal{P}(\tilde{\beta}) - \frac{1}{2}\left[\partial_\beta \mathcal{P}(\tilde{\beta})\right]^\top M^{-1}\left[\partial_\beta \mathcal{P}(\tilde{\beta})\right].$$

Proof Idea

A linear approximation of $\mathcal{L}(\beta)$ at $\mathcal{L}(\tilde{\beta})$ is a lower bound on $\mathcal{L}(\beta)$. This allows us to compute lower bound the minimum of $\mathcal{P}(\beta)$.

Selection Rule

Application of the Bound

If the gradients and the Hessian in β factorize as in the previous case, we obtain

$$\Delta\left[-\log p(\beta|X,Y)\right] \leq \frac{1}{2}\|Z\mathbf{c}' + \beta_n\|^2 + \frac{1}{2}(\mathbf{c}'_m + \beta_m)^\top \tilde{K}(\mathbf{c}'_m + \beta_m).$$

Here $\mathbf{c}'_m$ is the part of $\mathbf{c}'$ corresponding to β_m.

Problem

Which basis function to add to β_n (after the gradient on β_n vanishes)?

Approximate Solution

Since $\beta_m = 0$ we can rewrite the β_m term as $\frac{1}{2}(\mathbf{c}'_m)^\top \tilde{K}\mathbf{c}'_m$. Computing this is **expensive**, the diagonal terms, however, are cheap. We bound

$$\sqrt{(\mathbf{c}'_m)^\top \tilde{K}\mathbf{c}'_m} \leq \sum_{i=n+1}^{m} \sqrt{\tilde{K}_{ii}}|c'_i|$$

Hence, pivoting for i with large $\tilde{K}_{ii}(c'_i)^2$ is a good idea.