Issues and Challanges in On-Line Gaussian Process (?) Estimation

Tony J. Dodd, Robert F. Harrison, Visakan Kadirkamanathan and Supawan Phonphitakchai
Department of Automatic Control
and Systems Engineering
The University of Sheffield, UK
t.j.dodd@shef.ac.uk

Outline

- On-line, sequential and incremental learning.
- Gaussian processes and Hilbert spaces.
- Batch learning in RKHS.
- On-line learning in RKHS.
- Sparse solutions.
- Examples.
- Challenges and open questions.

On-line, Sequential and Incremental Learning

- Batch learning - all the data available.
- On-line learning - process single data point at a time.
- aka sequential learning.
- Should be recursive and incremental.

Applications

- On-line learning.
- Large data sets.
- Adaptive, non-stationary, learning.

Historial Perspective

- Simple on-line parameter estimation (LMS, NLMS, RLS, Kalman filter...).
- Method of potential functions (Aizerman and Braverman).
- Stochastic approximation (many).
- Resource allocating network (Platt and others).
- Constrained sequential projections in Hilbert space (Kadirkamanathan and Niranjan).
- On-line Gaussian processes (Csató and Opper).
- Exact incremental methods (Sugiyama and Ogawa).
- On-line kernel methods (Various including Kivinen et al).

Gaussian Processes and Hilbert Spaces

This talk is based on RKHS.

So what has this go to do with Gaussian processes?

Fundamental link is the covariance function.

Let $X(t)$ be a family of zero-mean Gaussian variables with $E[X(s)X(t)] = k(s,t)$.

Can also define a RKHS with reproducing kernel k.

Then the Hilbert space spanned by $X(t)$ is isometrically isomorphic to the RKHS.

There exists a 1:1 inner product preserving correspondence.

This is simplifying matters but is sufficient to motivate the rest of the talk.

More on RKHS in a minute.

Finite Data Function Approximation

Assume some unknown function, f.

Can only observe at finite number, N, of points.

f belongs to Hilbert space, $\mathcal{F}$, defined on input set $\mathcal{X} \subseteq \mathbb{R}^n$.

Denote observations by linear operator

$$z_i = L_i f.$$

Given class, $\mathcal{F}$, and observations, $\{z_i\}$, approximation problem is then to estimate f.

Written as linear operator equation

$$z = Lf = \sum_{i=1}^{N} (L_i f) s_i.$$

Reproducing Kernel Hilbert Spaces

Assume $\mathcal{F}$ is a RKHS then observation functionals, L_i, continuous (hence bounded).

By Riesz representation theorem

$$L_i f = \langle f, k(x_i), \cdot) \rangle$$

where $k(x_i, \cdot)$ is the reproducing kernel.

Conditions on $k(\cdot, \cdot)$:

1. $k(x, \cdot) \in \mathcal{F}$; and
2. $\langle f, k(x, \cdot) \rangle = f(x)$.

$k(\cdot, \cdot)$ is positive definite (RBF).

Functions, $g \in \mathcal{F}$,

$$g(\cdot) = \sum_{i=1}^{N} c_i k(x_i, \cdot).$$

Least Squares Solution

Since range of linear operator equation is finite dimensional it is closed.

Least squares solution, u, satisfies:

1. $Lu = Pz$;
2. $\|Lu - z\| \leq \|Lf - z\|$ for any $f \in \mathcal{F}$; and
3. $L^*Lu = L^*z$.

P denotes projection of z onto $R(L)$, and

L^* is the adjoint operator of L defined by

$$\langle Lf, z\rangle = \langle f, L^*z\rangle$$

(think matrix transpose).

Generalised Solution

Since $R(L)$ is closed a least-squares solution always exists, but may be many...

Seek the least-squares solution of minimum norm - generalised solution.

$$L^\dagger = (L^*L)^\dagger L^* = L^*(LL^*)^\dagger$$

Since finite-dimensional we have

$$L^*c = \sum_{i=1}^{N} k(x_i, \cdot)c_i,$$

$$LL^* = \sum_{j=1}^{N} \sum_{i=1}^{N} k(x_i, x_j)e_j e_i^T = K.$$

Then

$$f^\dagger(\cdot) = L^*(LL^*)^\dagger z = L^*c$$

and

$$f^\dagger(x) = \langle f^\dagger(\cdot), k(x, \cdot)\rangle = k^T K^{-1} z.$$

Regularised Solution

Generalised solution may still be sensitive to noise (but never ill-posed as problem is finite dimensional).

Method of Tikhonov regularisation

$$f_{reg} = \arg\min_{f\in\mathcal{F}} \frac{1}{2}\|Lf - z\|^2 + \frac{\rho}{2}\|f\|^2.$$

Unique minimiser

$$\begin{aligned} f_{reg}(\cdot) &= (\rho I + L^*L)^{-1}L^*z \\ &= L^*(\rho I + LL^*)^{-1}z \end{aligned}$$

and

$$\begin{aligned} f_{reg}(x) &= \langle f_{reg}(\cdot), k(x,\cdot)\rangle \\ &= k^T(\rho I + K)^{-1}z. \end{aligned}$$

Batch Gradient Methods

Assume all the data, z, is available and seek iterative solutions.

Define

$$J_{reg}(f) = \frac{1}{2}\|Lf - z\|^2 + \frac{\rho}{2}\|f\|^2$$

which is Fréchet differentiable at each point of $\mathcal{F}$ and

$$\nabla J_{reg}(f) = L^*Lf - L^*z + \rho f.$$

General iterative solutions - move in direction of negative gradient

$$f_0 \in R(L^*), \quad f_{n+1} = f_n - \eta_n \nabla J_{reg}(f_n).$$

Applicable to large data sets.

Gradient and Steepest Descent

Gradient descent:

$$f_0 \in R(L^*), \quad f_{n+1} = f_n - \eta_n \nabla J_{reg}(f_n),$$

$$0 < \eta_n < \frac{2}{\lambda_{max}(LL^*) + \rho}, \quad \sum_{n=0}^{\infty} \eta_n = \infty.$$

Steepest descent:

$$f_0 \in R(L^*), \quad f_{n+1} = f_n - \eta_n \nabla J_{reg}(f_n),$$

$$\eta_n = \frac{\|\nabla J_{reg}(f_n)\|^2}{\|L\nabla J_{reg}(f_n)\|^2 + \rho\|\nabla J_{reg}(f_n)\|^2}.$$

Conjugate gradient can also be developed similarly.

Early stopping.

Computational Forms

Since $f_n = L^* c_n$

$f_{n+1} = L^* c_n - \eta_n [L^*(LL^* c_n - z) + \rho L^* c_n]$

and letting

$$c_{n+1} = c_n - \eta_n (LL^* c_n - z) + \rho c_n$$

we have $f_{n+1} = L^* c_{n+1}$.

Computationally

$$c_0 \in \mathbb{R}^n, \quad c_{n+1} = c_n - \eta_n \bar{c}_n$$

where $\bar{c}_n = (K c_n - z) + \rho c_n$.

Gradient descent:

$$0 < \eta_n < \frac{2}{\lambda_{max}(K) + \rho}, \quad \sum_{n=0}^{\infty} \eta_n = \infty.$$

Steepest descent:

$$\eta_n = \frac{\bar{c}_n^T K \bar{c}_n}{\bar{c}_n^T K^2 \bar{c}_n + \rho \bar{c}_n^T K \bar{c}_n}.$$

Parametric vs functional forms.

Stochastic Gradient Methods

Suppose we make new observations at each iteration

$$z_n = L_n f.$$

Define instantaneous, non-negative, functional

$$\hat{J}^{reg}_{n+1}(f_n) = \frac{1}{2}\|L_{n+1} f_n - z_{n+1}\|^2 + \frac{\rho}{2}\|f_n\|^2.$$

Given initial approximation, f_0, method of stochastic gradient descent

$$f_{n+1} = f_n - \eta_{n+1} \nabla \hat{J}^{reg}_{n+1}(f_n)$$

where

$$\nabla \hat{J}^{reg}_{n+1}(f_n) = L^*_{n+1}(L_{n+1} f_n - z_{n+1}) + \rho f_n.$$

Hence

$$f_{n+1} = (1 - \eta_{n+1}\rho) f_n - \eta_{n+1} L^*_{n+1}(L_{n+1} f_n - z_{n+1}).$$

Computational Form (1)

For some constant, $a \in \mathbb{R}$,

$$L_{n+1}^* a = a k_{n+1}$$

and

$$L_{n+1} f_n = f_n(x_{n+1}).$$

Therefore

$$f_{n+1} = (1 - \eta_{n+1}\rho) f_n - \eta_{n+1}[f_n(x_{n+1}) - z_{n+1}] k_{n+1}.$$

Assume model at iteration n is

$$f_n = \sum_{i=1}^{p} c_n^i k_i$$

Then

$$\begin{aligned} f_{n+1} &= (1 - \eta_{n+1}\rho) \sum_{i=1}^{p} c_n^i k_i - \eta_{n+1} e_{n+1} k_{n+1} \\ &= \sum_{i=1}^{p+1} c_{n+1}^i k_i. \end{aligned}$$

Computational Form (2)

Parameters updated as

$$c_{n+1}^i = \begin{cases} (1 - \eta_{n+1}\rho)c_n^i & \text{for } i \leq p \\ -\eta_{n+1}e_{n+1} & \text{for } i = p+1 \end{cases}$$

New parameter equal to -prediction error on new data point weighted by learning rate.

Old parameters decayed by factor $(1-\eta_{n+1}\rho)$.

This is like a forgetting factor (decaying memory).

Insight: consider $\eta_{n+1} = \eta$, then

$$f_{n+1} = \sum_{i=1}^{p+1} (1 - \eta\rho)^{n+1-i} \eta e_i k_i.$$

Regularisation in on-line learning $\rightarrow$ decaying memory.

Conditions on Learning Rate

To ensure monotonicity of instantaneous error function require

$$0 < \eta_{n+1} < \frac{2\|\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2}{\|L_{n+1}\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2 + \|\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2}$$

Can also derive stochastic steepest descent

$$\eta_{n+1} = \frac{\|\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2}{\|L_{n+1}\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2 + \|\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2}$$

Construction of full convergence proof for these learning rates is ongoing.

Computing the Learning Rate

Can be shown that

$$\|\nabla \hat{J}^{reg}_{n+1}(f_n)\|^2 = k(x_{n+1}, x_{n+1})(f_n(x_{n+1}) - z_{n+1}) + \rho^2 c_n^T K_{p,p} c_n + 2\rho f_n(x_{n+1})(f_n(x_{n+1}) - z_{n+1})$$

and

$$\|L_{n+1} \nabla \hat{J}^{reg}_{n+1}(f_n)\|^2 = [k(x_{n+1}, x_{n+1})(f_n(x_{n+1}) - z_{n+1}) + \rho f_n(x_{n+1})]^2$$

where $K_{p,p} \in \mathbb{R}^{p \times p}$ is the kernel matrix.

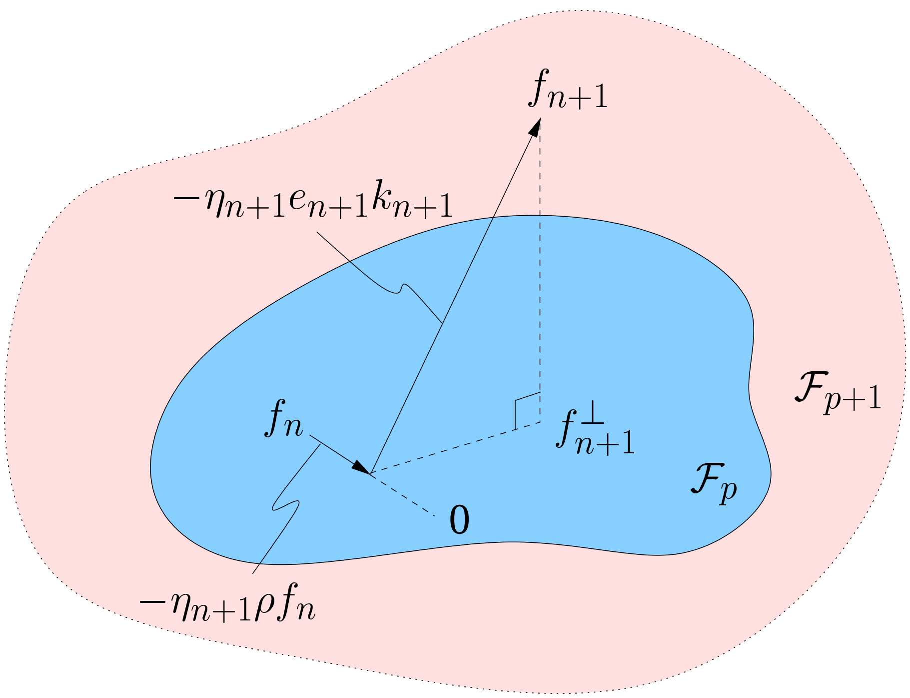
Sparsity

Problem: the number of terms grows without bound.

Solution: restrict model growth by only including "significant" kernels.

Also remove kernels which are no longer important.

But, always include effect of new data points on existing parameters.



Algorithm

1. Choose an initial approximation, f_0, typically zero.
2. Choose thresholds, κ_i, κ_d.
3. For each observation, $\{x_n, y_n\}$, calculate:
   - posterior estimate, f_{n+1};
   - posterior projection, $f_n^{\perp}$; and
   - the error norm, $\|f_{n+1} - f_{n+1}^{\perp}\|^2$.
4. If $\|f_{n+1} - f_{n+1}^{\perp}\| > \kappa_i$ update the function estimate as $f_{n+1}^i = f_{n+1}$ and set $m = p + 1$. Otherwise choose the update as $f_{n+1}^i = f_{n+1}^{\perp}$ and set $m = p$.
5. For all $i = 1, \ldots, m$, calculate
   - the decremental estimates f_{n+1}^{dl}, corresponding to the removal of the lth kernel; and
   - the error norm, $\|f_{n+1}^i - f_{n+1}^{dl}\|$.
6. If the lowest (out of all possible kernels) error norm satisfies $\|f_{n+1}^i - f_{n+1}^{dl}\| < \kappa_d$ update the function estimate. Otherwise do not include a decremental step.
7. Repeat steps 3 to 6 for each new data point.

Relationship to Other Methods

In the unregularised case, stochastic steepest descent is equivalent to:

- Method of potential functions.
- Stochastic approximation.
- Matching pursuit.
- Boosting.
- Resource allocating network.
- Method of $\mathcal{F}$-projections.

Regularised case very similar to:

- On-line Gaussian processes.
- Other on-line kernel algorithms.

Example: Channel Equalisation

Consider a communication channel

$$x_n = \sum_{i=0}^{L_C} h_i u_{n-i} + q_n$$

where

$\{u_n \in \mathcal{A}\}$ is a discrete-valued input sequence,

$\{x_n \in \mathbb{R}\}$ is the channel output sequence,

$\{h_i \in \mathbb{R}\}$ are the channel coefficients,

$\{q_n\}$ is a noise sequence, and

L_C is the channel order.

Equalisation problem: recover an estimate of $\{u_n\}$ given $\{x_n\}$.

Channel Equalisation (cont.)

For a symbol-decision equaliser

$$\hat{u}_{n-\tau} = d(y_n)$$

and

$$y_n = f(n, \bar{x}_n)$$

$$\bar{x}_n = [x_n, x_{n-1}, \ldots, x_{n-L_E}]^T.$$

The L_E order equaliser, with delay, τ, is given by $f(n, \cdot)$, and $d(\cdot)$ is a decision function with range $\mathcal{A}$.

Problem: estimate the function $f(\cdot)$.

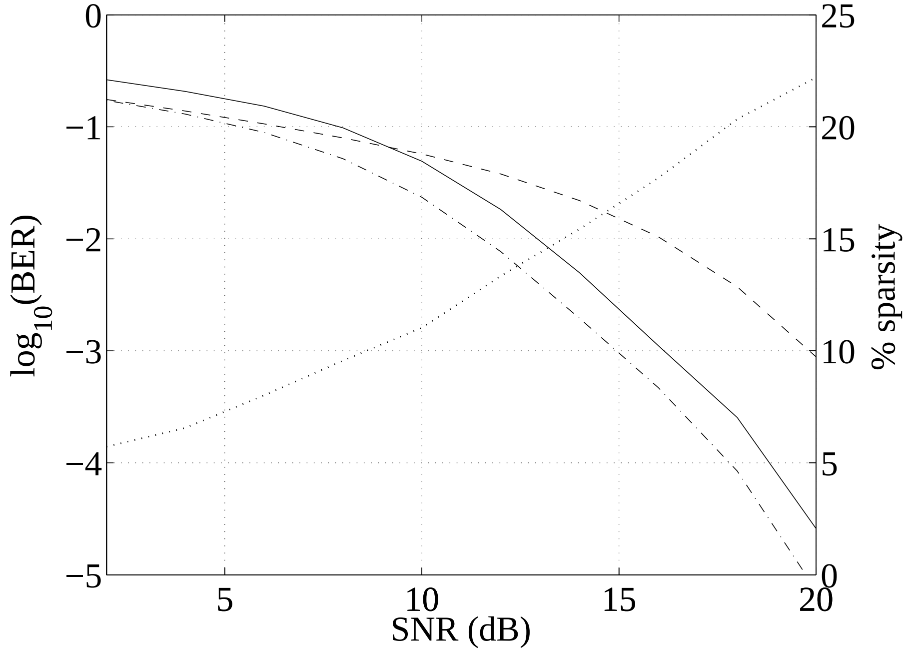
Choose:

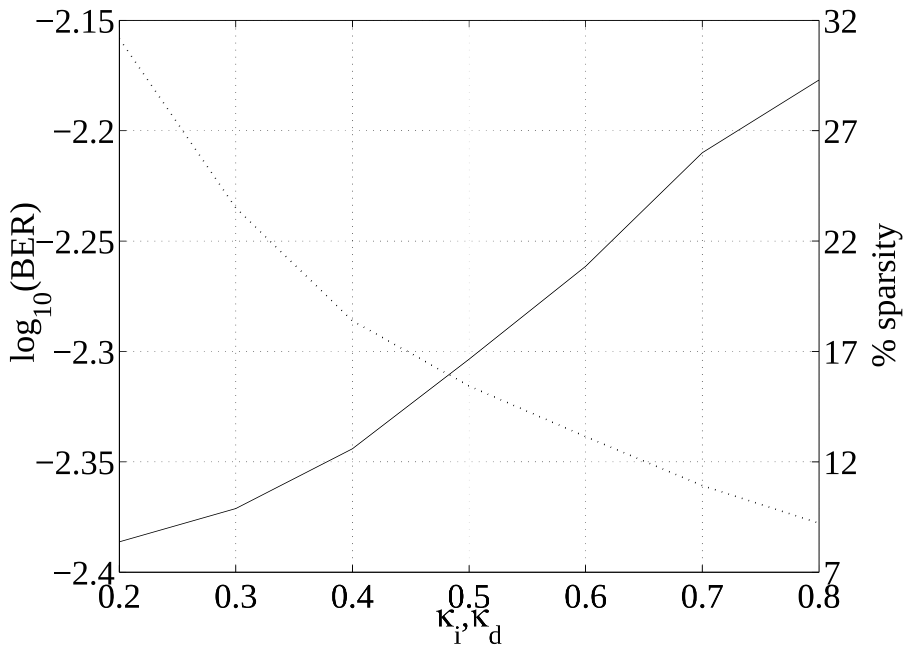
$$[h_0, h_1, h_2] = [0.3482, 0.8704, 0.3482]$$

and $L_E = 3, \tau = 1, u_n \in \{\pm 1\}$ and $q_n \sim N(0, \sigma_q^2)$.

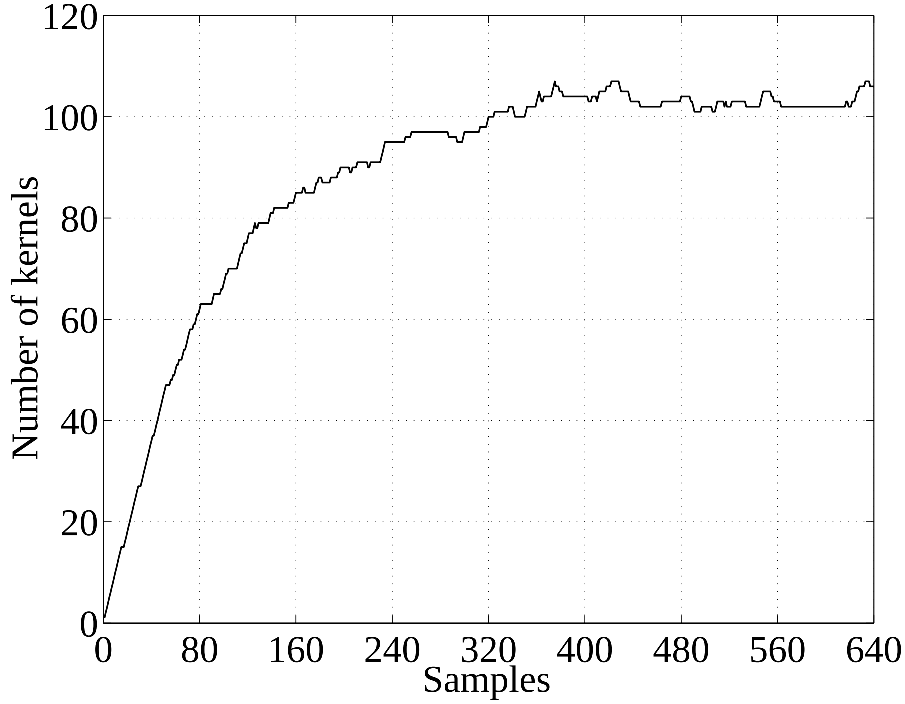
Use Gaussian kernel with $\sigma = 2\sigma_q$.

Results


0
−1
−2
−3
−4
−5
log10(BER)
5
10
15
20
SNR (dB)
25
20
15
10
5
0
% sparsity


−2.15
−2.2
−2.25
−2.3
−2.35
−2.4
log10(BER)
0.2
0.3
0.4
0.5
0.6
0.7
0.8
κi,κd
32
27
22
17
12
7
% sparsity

Results


120
100
80
60
40
20
0
Number of kernels
0
80
160
240
320
400
480
560
640
Samples

Learning Rates

A number of approaches have been proposed for the learning rates.

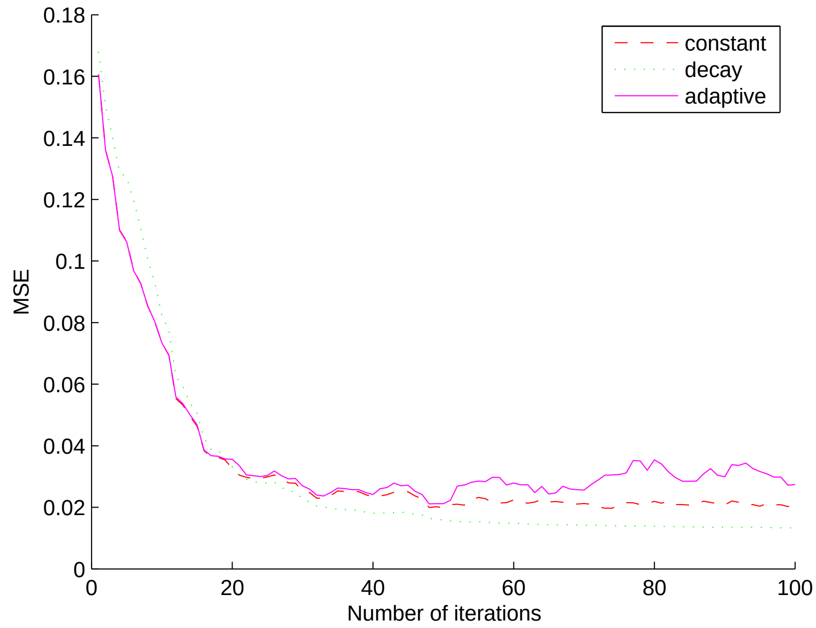
Three considered:

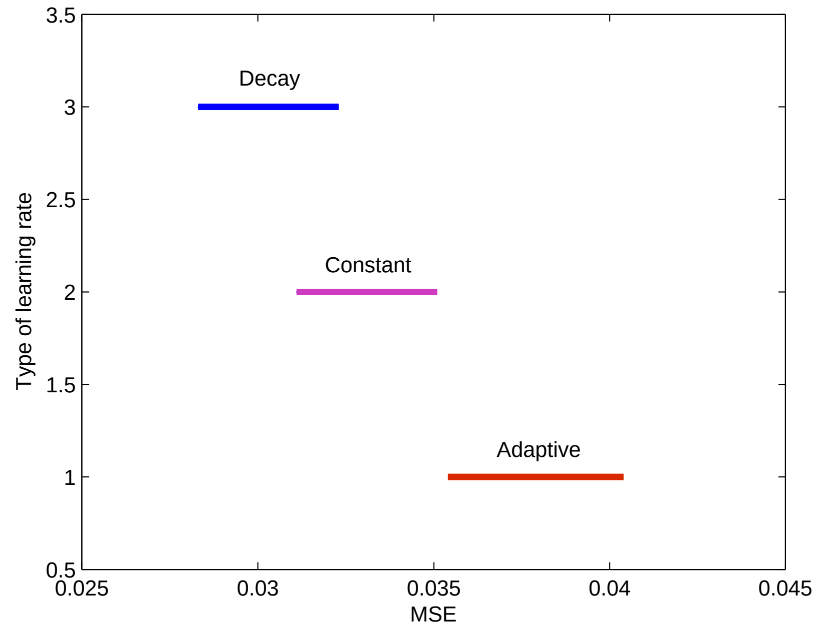
- Constant - $\eta_n = \eta$.
- Decay - $\eta_n = \eta_0 n^{-\lambda}$.
- Adaptive - η_n as previous.

Which is the most appropriate:

- Convergence.
- Non-stationarity.

Comparing Learning Rates





Challenges and Open Questions

- Completion of convergence proofs (large data).
- What is the best learning rate?
- Second order methods.
- Efficient model computation.
- Hyperparameters.
- Uncertainty.
- Stochastic conjugate gradient methods.
- What about non-stationary processes.
- What about correlated data - time series.
- Recurrent - non-Gaussian.