JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

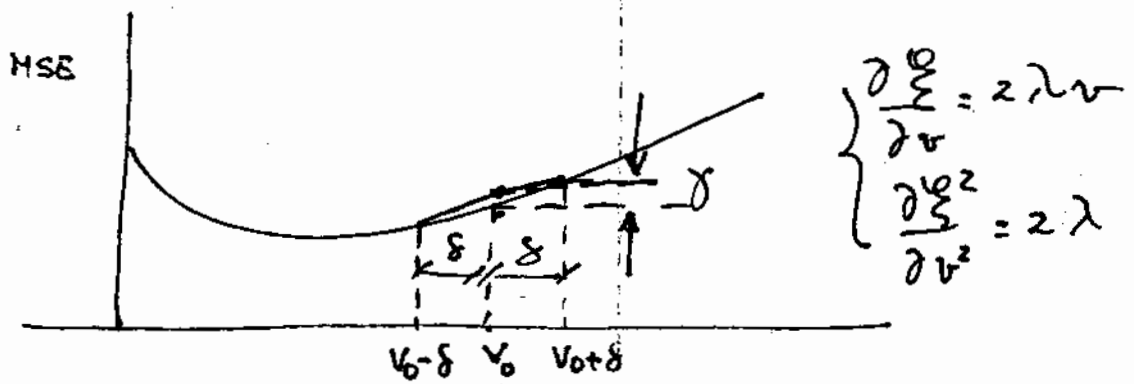
904-392-2662

principe@brain.ee.ufl.edu

GRADIENT ESTIMATION

ESTIMATION BY DERIVATIVE

Assume we know zeta $\xi = \xi_{MIN} + \lambda v^2$



If we estimate numerically by central differences

$$\frac{\partial \xi}{\partial v} \simeq \frac{\xi(v_0+\delta) - \xi(v_0-\delta)}{2\delta}$$

When δ -> 0 approximation becomes exact. (if surface is quadratic even for finite delta this is exact)

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

To implement the estimation we need to compute eta @ $V_o+\delta$ and $V_o\delta$. This quantity will be in error by γ, which is,

$$\gamma = \frac{1}{2}\left[\xi(v_o-\delta) + \xi(v_o+\delta)\right] - \xi(v_o)$$

or for quadratic surfaces,

$$\gamma = \frac{1}{2}\left[2\xi_{MIN} + \lambda(v_o+\delta)^2 + \lambda(v_o-\delta)^2\right] - (\xi_{MIN} + \lambda v_o^2)$$

PERFORMANCE PENALTY is defined as

$$P = \frac{\gamma}{\xi_{MIN}} = \frac{\lambda\delta^2}{\xi_{MIN}}$$

and it translates the average increase in the error normalized to eta min.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ec.ufl.edu

If we have multiple weigths

$$\xi = \xi_{MIN} + V^T R V = \xi_{MIN} + \begin{bmatrix} V_0 & V_1 \end{bmatrix} \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}$$

$$= r_{00} V_0^2 + r_{11} V_1^2 + 2 r_0 V_0 V_1$$

along each axis we have

$$P_0 = \frac{r_{00}\,\delta^2}{\xi_{MIN}} \quad ; \quad P_1 = \frac{r_{11}\,\delta^2}{\xi_{MIN}}$$

and the average perturbation is $P = \frac{1}{2}(P_0 + P_1)$

$$P = \frac{\delta^2}{\xi_{MIN}} \left(\frac{r_{00} + r_{11}}{2}\right)$$

For L+1 weights

$$P = \frac{\delta^2}{\xi_{MIN}} \frac{\text{TRACE}[R]}{L+1}$$

or (since the trace is the sum of eigenvalues)

$$P = \frac{\delta^2}{\xi_{MIN}} \sum_{n=0}^{L} \frac{\lambda_N}{L+1} = \frac{\delta^2}{\xi_{MIN}} \lambda_{AVE.}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

ESTIMATION THEORY

We have assumed that we knew eta. But

$$\xi_s = E\{e_k^2\}$$

$$\xi_t = \sum_{-\infty}^{\infty} e_k^2$$

In each we have finite data sets, i.e. we have to estimate eta, the SECOND moment of the random process. FOR THE FIRST MOMENT,

$$m_x = E\{x_n\} = \int_{-\infty}^{\infty} x\, p_x(x)\, dx$$

$$m_x = A\{x_n\} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{-N}^{N} x_n$$

We assume that the r.p. is ergodic, i.e. $\xi_s = \xi_t$

It is plausible that if N is large the estimate given by the sampled mean will be good. When we use this estimate we are in fact using MAXIMUM LIKELIHOOD estimates. The rth moment is estimated by

$$\hat{\alpha}_r = \frac{1}{N} \sum_{k=1}^{N} (e_k)^{rr}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

The sampled mean for Gaussian r.v. is consistent, i.e. the bias is zero

$$\alpha_n = E[\hat{\alpha}_n]$$

$$Var[\hat{\alpha}_n] = E[(\hat{\alpha}_n - \alpha_n)^2] = \frac{\alpha_{2n} - \hat{\alpha}_n^2}{N}$$

and the variance decreases with N.

We want to estimate the gradient of the performance surface, so let us see how this theory applies.

Notice that eta is defined as the second moment of e_n. Since the estimator is unbiased (estimated mean equals the true mean), the variance of the estimate is

$$Var[\hat{\xi}] = \frac{\alpha_4 - \alpha_2^2}{N}$$

To compute the power one must know the p.d.f. of e_n. Let us assume that it is zero mean Gaussian, with variance σ^2

Then,

$$Var[\hat{\xi}] = \frac{3\sigma^4 - \sigma^4}{N} = 2\frac{\xi^2}{N}$$

In general 2 is exact or conservatively high.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

Knowing the variance of , eta, we are ready to compute the variance of the gradient estimate. Since

$$\frac{\partial \hat{\xi}}{\partial v} = \frac{1}{2\delta}\hat{\xi}(v+\delta) - \frac{1}{2\delta}\hat{\xi}(v-\delta)$$

and assuming independent estimates

$$\mathrm{Var}\left[\frac{\partial \hat{\xi}}{\partial v}\right] = \frac{1}{4\delta^2}\mathrm{Var}\left[\hat{\xi}(v+\delta)\right] + \frac{1}{4\delta^2}\mathrm{VAR}\left[\hat{\xi}(v-\delta)\right]$$

$$= \frac{1}{2N\delta^2}\left[\xi(v+\delta) + \xi^2(v-\delta)\right]$$

After convergence, and for small perturbation delta,

$$\mathrm{Var}\left[\frac{\delta \hat{\xi}}{\partial v}\right] = \frac{\xi^2_{MIN}}{N\delta^2}$$

so

Assuming independence of ε_k (therefore of the errors),and that N and δ are the same for all componenets of the gradient, the covariance of the estimated gradient is

$$\mathrm{cov}\left[\hat{\nabla}_k\right] = E\left[(\hat{\nabla}_k - \nabla_k)(\hat{\nabla}_k - \nabla_k)^T\right] = \frac{\xi^2_{MIN}}{N\delta^2} I$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

Notice that the covariance is inversely proportional to δ, i.e. the larger the δ the better the estimate (why?).

With this result we can now calculate the effect of the imprecision (noise) in the gradient into the weight solution. Remmember that we are calculating the weights iteratively using the gradient information.

WEIGHT VECTOR COVARIANCE

NEWTON'S METHOD

Let us write the true gradient as

$$\hat{\nabla}_k = \nabla_k + N_k$$

The recursion to calculate the new weight vector is

$$W_{k+1} = W_k + \mu R^{-1} \hat{\nabla}_k$$

Substituting the noisy gradient

$$W_{k+1} = W_k - \mu R^{-1} \nabla_k - \mu R^{-1} N_k$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

System is cross coupled (R^{-1} N). However in the principal coordinate system

$$V_{k+1} = (1-2\mu)V_k - \mu R^{-1} N_k$$

$$\Rightarrow \quad V'_{k+1} = (1-2\mu)\, V'_k - \mu \Lambda^{-1} N'_k \qquad (Q^{-1}N = N')$$

and the equations become decoupled. By induction the solution is

$$V'_{k+1} = (1-2\mu)^k V'_0 - \mu \Lambda^{-1} \sum_{n=0}^{k-1} (1-2\mu)^n N'_{k-n-1}$$

Note that we have the previous term plus a noise component. For very large k when $\mu \sim 1/2$, the first term vanishes and

$$V'_{k+1} = -\mu \Lambda^{-1} \sum_{n=0}^{\infty} (1-2\mu)^n N'_{k-n-1}$$

This is the steady state error after adaptation.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

STEEPEST DESCENT $\left(W_{k+1} = W_k - \mu \nabla_k \right)$

Once again the recusrion can be written as a function of the true gradient plus the noise component.

In the principal coordinate system, $V_{k+1} = (I - 2\mu R) V_k - \mu N_k$

$$V'_{k+1} = (I - 2\mu \Lambda) V'_k - \mu N'_k$$

and by induction the solution can be written,

$$V'_k = (I - 2\mu \Lambda)^k V'_0 - \mu \sum_{n=0}^{k-1} (I - 2\mu \Lambda)^n N'_{k-n-1}$$

If μ is chosen for convergence, for large k the first term vanishes, and the steady state solution becomes $V'_k = -\mu \sum_{n=0}^{\infty} (I - 2\mu \Lambda)^n N$

So the net effect of the noisy gradient estimates is to produce even after adaptation a change in the weigth vector. The question now is how much noise power? Since we have already the covariance of the estimated gradient vector we can calculate the covariance of the weight vector solution.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

NEWTON'S

The covariance is $Cov\left[V_k'\right] = E\left[V_k' V_k'^T\right]$

and substituting we get

$$cov\left[V_k'\right] = \frac{\mu (\Lambda^{-1})^2}{4(1-\mu)} cov\left[N_k'\right]$$

In the STEEPEST DESCENT

$$cov\left[V_k'\right] = \frac{\mu}{4}(\Lambda - \mu\Lambda^2)^{-1} cov\left[N_k'\right]$$

Now substituting the calculated value of the gradient cov.

NEWTON'S

$$cov\left[V_k'\right] = \frac{\mu (\Lambda^{-1})^2 \xi_{MIN}^2}{4N\delta^2(1-\mu)}$$

STEEPEST

$$cov\left[V_k'\right] = \mu \frac{(\Lambda - \mu\Lambda^2)^{-1} \xi_{MIN}^2}{4N\delta^2}$$

$$\left(cov\left[N_k'\right] = E\left[N_k' N_k'^T\right] = Q^{-1} E\left[N_k N_k^T\right] Q = \frac{\xi_{MIN}^2}{N\delta^2} Q^{-1} I Q \right)$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

In the translated coordinate system, noting that

$$cov[V_k] = E[V_k V_k^T] = Q\, E[V'_k V'^T_k] Q^T = Q\, cov[V'_k] Q^T$$

we get

NEWTON'S

$$cov[V_k] = \frac{\mu\, \xi^2_{MIN} (R^{-1})^2}{4N\delta^2(1-\mu)}$$

STEEPEST

$$cov[V_k] = \mu \frac{\xi^2_{MIN} (R - \mu R^2)^{-1}}{4N\delta^2}$$

Note that: in the estimation of the gradient

1. If there is no noise, cov[V] is zero.
2. The larger the # of measurements, the smaller the error
3. the larger the δ, the smaller the error.
4. The smaller the μ the smaller the error.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

EXCESS MEAN SQUARE ERROR

Noise in the gradient estimate implies inaccuracies in the filter coefficients,which mean that even after adaptation, the coefficients will be moving around the bottom of the "bowl".

This means that the error will not be the theoretical minimum, i.e. an excess mean square error will result from the noisy estimation of the gradient.

Notice that this does not include the effect of the perturbation required to estimate the derivative.

Let us characterize the mean of this error. Assume independence in the errors across iterations, stationarity and convergence.

In the principal component system,

$$\text{Excess MSE} = E\left[\xi_k - \xi_{MIN}\right] = E\left[V_k^T R V_k\right]$$

$$\Rightarrow E\left[V_k'^T R V_k'\right]$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

NEWTON'S

Define r=1-2μ

$$ex\ MSE = \mu^2 \sum_n \sum_m r^{n+m} E\left[N'^T_{k-n-1} \Lambda^{-1} N'_{k-n-1} \right]$$

Since independent,

$$ex\ MSE = \mu^2 \sum_{n=0}^{\infty} r^{2n} E\left[N'^T_{k-n-1} \Lambda^{-1} N'_{k-n-1} \right]$$

and stationarity,

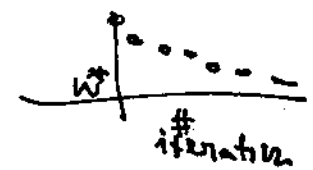
$$ex\ MSE = E\left[N'^T_k \Lambda^{-1} N'_k \right] \frac{\mu^2}{1-r^2}$$

After evaluation of the matrix

$$ex\ MSE = \frac{\xi^2_{min}\, \mu^2}{N\delta^2(1-r^2)} \sum_{m=0}^{L} \frac{1}{\lambda_m}$$

If we express the result in terms of τ, the adaptive time constant,

$$r \approx 1 - \frac{1}{\tau} \Rightarrow \mu \approx \frac{1}{2\tau}$$



we get

$$excess\ MSE = \frac{\xi^2_{min}}{8N\delta^2\tau} \sum_{m=0}^{L} \frac{1}{\lambda_m}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

Defining the average inverse eigenvalue as,

$$\sum_{m=0}^{L} \frac{1}{\lambda_m} = (L+1)\left(\frac{1}{\lambda}\right)_{avg}$$

and substituting P

$$\text{excess MSE} \simeq \frac{(L+1)\,\xi_{min}\,\lambda_{av}\,(1/\lambda)_{av}}{8NP\sigma}$$

Notice that the excess MSE is directly proportional to the number of filter coefficients, inversely proportional to the adaptive time constant and the perturbation.

STEEPEST

The method is the same. Substitute V in the definintion of the excess MSE, we get

$$\text{excess MSE} = \frac{\mu\,\xi_{min}\,\lambda_{av}}{4NP} \sum_{m=0}^{L} \frac{1}{1-\mu\lambda_m}$$

Now the problem is that we do not have a single adaptive time constant.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

TIME CONSTANT OF THE LEARNING CURVE, AND THE TIME CONSTANT OF ADAPTATION.

For the learning curve adaptation, since $r_{mse} = r^2$

$$\exp\left(-\frac{1}{\tau_{mse}}\right) = \exp\left(\frac{-2}{\tau}\right)$$

then $$\boxed{\tau_{mse} = \frac{\tau}{2}}$$

This constant describes the learning time. The unit is the iteration number.

The units for the time constant of adaptation, T is the data sample (can be related to the sampling rate). Since we need 2N samples per gradient component, then 2(L+1)N samples per iteration.

$$T_{mse} = 2(L+1)N\tau_{mse} = N(L+1)\tau$$

$$= \frac{N(L+1)}{2\mu}$$

CANTRADE N FOR μ $\Longrightarrow$ WHAT MATTERS IS # OF POINTS PER TIME CONSTANT

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu.

Returning to the steepest descent, the nth adaptive time constant is $\tau_n = 1 - 2\mu\lambda_n$; $\mu\lambda_n = \frac{1}{2\tau_n}$

If we proceed to calculate the nth time constant of the learning process

$$(\tau_{mse})_n = \frac{\tau_n}{2}$$

and the nth t.c. for the adaptation process

$$(T_{mse})_n = N(L+1)\tau_n \Longrightarrow \frac{N(L+1)}{2\mu\lambda_n}$$

averaging over L we get

$$\mu\lambda_{av} = \frac{N(L+1)}{2}\left(\frac{1}{T_{mse}}\right)_{ave.}$$

and assuming t_n large, we have

$$\text{excess MSE} \approx \frac{(L+1)^2\xi_{min}}{8P}\left(\frac{1}{T_{mse}}\right)_{ave.}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu.

For the Newton's method also

NEWTON $\quad \text{excess MSE} = \dfrac{(L+1)^2 \xi_{min} \lambda_{ave}(1/\lambda_{ave})}{8P\, T_{mse}}$

STEEPEST $\quad \text{excess MSE} = \dfrac{(L+1)^2 \xi_{MIN}}{8P} \left(\dfrac{1}{T_{mse}}\right)_{ave.}$

These expressions show that the excess MSE:

1. decreases with larger perturbations (more accurate gradient).
2. decreases with larger time constants (more data is being used).
3. increases with the square of the weights (try always to live with the minimum # of degrees of freedom).
4. eigenvalues only affect the error if there is a large disparity between them.

Can trade time constant and N. So, what is important is the number of points used per time constant of adaptation.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

MISADJUSTMENT

The excess mean square error is the average MSE less the minimum MSE. $M \triangleq \left(\frac{\text{excess MSE}}{\xi_{MIN}}\right)$

MISADJUSTMENT is the excess MSE normalized by the minum MSE. It is dimensionless, but now it becomes a good characterization of the cost of adaptability.

Notice that misadjustment of the Steepest and Newton is equal if all the eigenvalues of R are equal.

For the same speed of adaptation, they have different misadjustements.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

COMPARISON OF NEWTON'S AND STEEPEST

We have to compare them in terms of misadjustment and speed of adaptation.

Writing $$(T_{mse})_{MAX} = \frac{(L+1)\,\mu}{2\mu\,\lambda_{MIN}}$$

we can express M for the steepest descent as

$$M \simeq \frac{(L+1)^2\,\lambda_{av}}{8P\,\lambda_{MIN}\,(T_{mse})_{MAX}}$$

while for the Newton's is

$$M \simeq \frac{(L+1)^2\,\lambda_{av}\,(1/\lambda_{ave})}{8P\,T_{mse}}$$

We see that if $T_{mse,max} = T_{mse}$ with the other things constant, Newton's method has a smaller misadjustment.

$$\frac{1}{\lambda_{MIN}} \geq \left(\frac{1}{\lambda}\right)_{ave}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

TOTAL MISADJUSTMENT

It may seem that we can tradeoff misadjustment by perturbation. If we add the perturbation to the misadjustment, we get the total misadjustment.

$$M_{TOT} \triangleq M + P$$

This is the measure that we should be interested in. Now the form of the total misadjustment is

$$M_{TOT} \simeq P + \frac{A}{P}$$

So the best perturbation is achieved by setting the derivative =0, which means

$$P_{OPT} \simeq \frac{1}{2} M_{TOT}$$

Total misadjustment is minimzed when the perturbation is half of the total misadjustment.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904 392-2662

principe@brain.ee.ufl.edu

LMS ALGORITHM

LMS is a steepest descent algorithm. Its basic features are:

1. uses a very simple estimate of the performance surface.

$$\xi_k^2 = (d_k - y_k)^2 \qquad \text{ESTIMATE OF } E\{\xi_k^2\}$$

2. Assumes zeta quadratic

DERIVATION

$$\varepsilon_k = d_k - y_k = d_k - \bar{X}_k^T W_k$$

If zeta is quadratic then

$$\hat{\nabla}_k = \begin{bmatrix} \frac{\partial \xi^2}{\partial w_0} \\ \vdots \\ \frac{\partial \xi^2}{\partial w_L} \end{bmatrix} \simeq \begin{bmatrix} \frac{\partial \varepsilon_k^2}{\partial w_0} \\ \\ \frac{\partial \varepsilon_k^2}{\partial w_L} \end{bmatrix} = 2\varepsilon_k \begin{bmatrix} \frac{\partial \varepsilon_k}{\partial w_0} \\ \vdots \\ \frac{\partial \varepsilon_k}{\partial w_L} \end{bmatrix} = -2\varepsilon_k \bar{X}_k$$

Therefore,

$$\bar{w}_{k+1} = \bar{w}_k + 2\mu\varepsilon_k \bar{X}_k = w_k + \mu'(d_k - y_k)\bar{X}_k$$

$$\hat{\nabla}_k$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935 SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

Can see that the update of coefficients involve ONLY a matrix multiplication. So the assymptotic complexity of the algorithm (the big O notation) is

$O(n)$ $N = L + 1$

i.e., linear in the number of weights.

This is VERY EFFICIENT!

The algorithm does not use any averaging, squaring, perturbation, etc. Each component of the gradient is obtained from a SINGLE data point. NO assumptions or knowledge about the statistics of the input/ desired signal are made.

As can be expected the estimation of the gradient is VERY NOISY. What is amazing is that the iterative process works as an effective lowpass filter.....

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

CONVERGENCE OF THE ALGORITHM

$E[\hat{\nabla}_k]$ is UNBIASED. In fact

$$= -2\,E\left[\varepsilon_k \bar{X}_k\right] = -2\,E\left[d_k \bar{X}_k - X_k X_k^T W_k\right]$$

$$= 2\left(R W - P\right) = \nabla$$

So the LMS algorithm can be made conceptually identical to the steepest descent. Just compute the estimate, but instead of updating the weights at each step, only change them after a large number of smaples. These two procedures are equivalent if the estimator is unbiased. In fact,

$$E\left[W_{k+1}\right] = E\left[W_k\right] + 2\mu\, E\left[\varepsilon_k X_k\right] =$$

$$= E\left[W_k\right] + 2\mu\left\{P - R\,E\left[W_k\right]\right\} =$$

$$= E\left[W_k\right]\left(I - 2\mu R\right) + 2\mu R W^* \quad (P = R W^*)$$

This is equvalent to the expression for the steepest descent.

In the principal coordinate system

$$E\left[W_k'\right] = \left(I - 2\mu\Lambda\right)^k V_0'$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

and we know that this convergences to W* if

$$0 < \mu < \frac{1}{\lambda_{MAX}}$$

As

$$\lambda_{MAX} < \sum \lambda_i = tr[R]$$

and

$$0 < \mu < \frac{1}{tr[R]}$$

$$tr[R] = (L+1)\, E\{x_k^2\}$$

we can say

$$\boxed{0 < \mu_{THEO} < \frac{1}{(L+1)\,(\text{SIGNAL POWER})}}$$

Practically,

$$\mu \simeq 0.1\, \mu_{THEORY}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

LEARNING CURVE

In the steepest descent wee have defined the time constant of each mode as

$$\tau_n = \frac{1}{2\mu\lambda_n} \qquad n = 0, 1, \ldots L$$

and for the time constant of the learning curve

$$\tau_{mse} = \frac{\tau}{2} \Rightarrow (\tau_{mse})_n = \frac{1}{4\mu\lambda_n}$$

Due to the fact that the LMS only uses 1 data sample per iteration,

$$(T_{mse})_n = (\tau_{mse})_n$$

This result is an optimistic approximation since the error may be very different from its expected value.

Notice that $T_{mse} = \max_i \left\{ \frac{1}{4\mu\lambda_i} \right\} = \frac{1}{4\mu\lambda_{MIN}}$

But as $\mu < 1/\lambda_{max}$, if say $\mu = \frac{\alpha}{\lambda_{MAX}} \qquad 0 < \alpha < 1$

then $$\boxed{T_{mse} = \frac{1}{4\alpha} \frac{\lambda_{MAX}}{\lambda_{MIN}}}$$

So, time constant of adaptation is limited by the eigenvalue spread.

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

NOISE IN WEIGTH SOLUTION

Need another expression because LMS does not use perturbation. Write as before

$$\hat{\nabla}_k = \bar{\nabla}_k + \bar{N}_k$$

Near adaptation ($\bar{\nabla}_k = 0$) $\quad \hat{\nabla}_k = \bar{N}_k = -2\,\varepsilon_k X_k$

So $\mathrm{Cov}[N_k] = 4\,E\left[\varepsilon_k^2 X_k X_k^T\right]$

Assuming uncorrelated noise and signal vectors

$$= 4\,E\left[\varepsilon_k^2\right] \cdot E\left[X_k X_k^T\right] = 4\,\xi_{MIN} R$$

In the principal coordinate system

$$\mathrm{cov}\left[N'_k\right] = 4\,\xi_{MIN}\,\Lambda$$

So the covariance of the coefficient vector is

$$\mathrm{Cov}\left[v'_k\right] = \mu\,\xi_{MIN}\left(\Lambda - \mu\Lambda^2\right)^{-1}\Lambda$$

JOSE C. PRINCIPE
UNIVERSITY OF FLORIDA
EEL 6935- SPRING 90
904-392-2662
principe@brain.ee.ufl.edu

If $\mu \Lambda^2 << \Lambda$

$$cov\left[V_k'\right] = \mu \xi_{min} I$$

The the covariance of V_k is proportional to μ.

MISADJUSTMENT

Now can write

$$\text{excess MSE} = E\left[V_k'^T \Lambda V_k'\right] = \sum_{n=0}^{L} \lambda_n E\left[v_{nk}'^2\right]$$

Assuming that $E[v_{nk}']$ is a component of the covariance (the analysis is done after transients died out),

$$\text{excess MSE} = \sum_{n=0}^{L} \lambda_n \cdot \left[\mu \xi_{min} I\right] = \mu \xi_{min} tr[R]$$

So

$$\boxed{M = \mu tr[R]}$$

i.e. the misadjustment in the LMS is proportional to μ.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

From

$$\begin{cases} M = \mu \, tr[R] \\ \tau_{mse} = \dfrac{1}{4\mu\lambda_n} \end{cases}$$

we see that there is a trade-off between speed of adaptation and misadjustment. As

$$tr[R] = \sum_0^L \lambda_n = \frac{1}{4\mu} \sum \frac{1}{(\tau_{mse})_n} = \frac{L+1}{4\mu} \left(\frac{1}{\tau_{mse}}\right)_{AVG}$$

so

$$M = \frac{L+1}{4} \cdot \left(\frac{1}{\tau_{mse}}\right)_{AVG}$$

This is a good approximation. Since after 4 time constants the adaptation transients have died out, we can say that in the LMS MISADJUSTMENT IS EQUAL TO THE NUMBER OF WEIGHTS DIVIDED BY SETTLING TIME (for equal weights).

$$M = \frac{L+1}{4\,\tau_{mse}}$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

PERFORMANCE LMS - STEEPEST DESCENT

LMS is better (less computation, faster to converge):

1. LMS misadjustment increases linearly with weights
2. For same misadjustment LMS is much faster
3. LMS shares with steepest descent the problem of adaptation time dependency on eigenvalue spread.

(i.e. Misadjustment controlled by fast modes

Time constant controlled by slowest modes)

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

OTHER FORMS OF LMS ALGORITHMS

COMPLEX LMS

If data is complex, then complex LMS

$$w_{k+1} = w_k + 2\mu\, \varepsilon_k\, x_k^*$$

NORMALIZED LMS

Speed of converge depends on signal power

$$\lambda_{MAX} = \sum_{i=0}^{L-1} \lambda_i = E[x_k x_k^T] \cdot (L+1)$$

$$0 < \mu < \frac{1}{\lambda_{MAX}}$$

Alternatively, if signal power changes, a fixed μ can lead to divergence.

To cope with this, NORMALIZE stepsize by the input power.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

SIGN ERROR LMS

In the other end of things (simplicit) we can try to avoid the multiplications. (NOTE: this is not an issue anymore with DSP chips).

We saw that to update the coefficients we need L+1 multiplications, and L+1 multiplications to implement the filter. So total ~2L.

One idea is just use the sign of the error for adaptation.

$$\varepsilon_k = d_k - y_k \Longrightarrow \varepsilon'_k = \text{SGN}\left[d_k - y_k\right]$$

$$\text{SGN}\left[d_k - y_k\right] = \begin{cases} 1 & \text{if } d_k > y_k \\ 0 & \text{if } d_k = y_k \\ -1 & \text{if } d_k < y_k \end{cases}$$

There is a penalty paid, because the estimate of the gradient became even noisier. To obtain a similar performance to the straight LMS must use smaller mus (which imply slower adaptation).

The method changes the scale of the update vector.

$$\bar{w}_{k+1} = \bar{w}_k + \mu\, \varepsilon'_k\, \bar{x}_k$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

With this scheme we still need L+1 multiplications.

There are 2 ways to cut down on this:

1. First is to approximate mu by negative powers of 2 (left RIGHT shifts). To update the coefficients just left shift the input and add. This may be a significant speed improvement.

2. Take only the sign of the input.

$$\bar{x}_k' \rightarrow \text{SGN}[x_k]$$

$$\bar{w}_{k+1} = \bar{w}_k + \mu\, \varepsilon_k\, \bar{X}_k'$$

This modification is more drastic because the DIRECTION of the LMS update is changed. It can lead to instability. Combining the two simplifications

$$w_{k+1} = w_k + \mu\, \underbrace{\varepsilon_k'}\, \underbrace{x_k'}$$

In the CCITT standard for ADPCM transmission this was the adopted model.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

We can define

$$\mu(k) = \frac{\alpha}{X_k^T X_k} \qquad 0 < \alpha < 1$$

With this μ,

$$W_{k+1} = W_k + \frac{2\alpha}{X_k^T X_k} \varepsilon_k X_k$$

Experimentally it was verified that $X_k X_k^T$ can become too small, so modify to (heuristics)

$$\bar{W}_{k+1} = \bar{W}_k + 2\alpha \varepsilon_k \frac{\bar{X}_k}{\gamma + X_k^T X_k}$$

where gamma is a small positive constant (can be 1).

It may seem that we made the algorithm much more complex (a vector multiplication). However, this is not so.

Remmember that

$$X_k^T X_k = \sum_{i=0}^{L-1} x^2(k-i)$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

FIR so if we use L+1 storage locations, can compute $X_{k+1} X_{k+1}^T$ by adding x^2 (k+1) and subtracting x^2(k-L), so just need squaring and an extra division.

IIR Can implement this operation recursively, through an exponentially weighted time average of the input square. Notice that we can compute recursively an estimate of the power by

$$\sigma_{k+1}^2 = \alpha \sigma_k^2 + (1-\alpha) x_k^2 \qquad \left(H(z) = \frac{(1-\alpha) z^{-1}}{z - \alpha} \right)$$

If you think of it, this is the output of a lowpass filter with a pole at alpha, when the input is the signal squared. For alpha close to 1 (.9 to .99) we are smoothing the input square (the envelope). Now the limit of this iteration for large k is

$$\sigma_{k-}^2 = (1-\alpha) \sum_{j=0}^{\infty} \alpha^j x^2(k-j)$$

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

LMS in non stationary environments

Figures of merit of the LMS have been derived assuming stationarity conditions.

Nonstationarity may arise at least in 2 important cases:

- The desired signal is time varying (system identification). Here the autocorrelation function is time invariant, but the crosscorrelation function is time varying.

- The input is non-stationary (equalization, noise cancellation). Here both the autocorrelation and crosscorrelation are non stationary.

In non-stat environments, the optimum set of weights change with time. Therefore, the LMS must not only find the minimum, but also track the change of the minimum of the performace surface.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

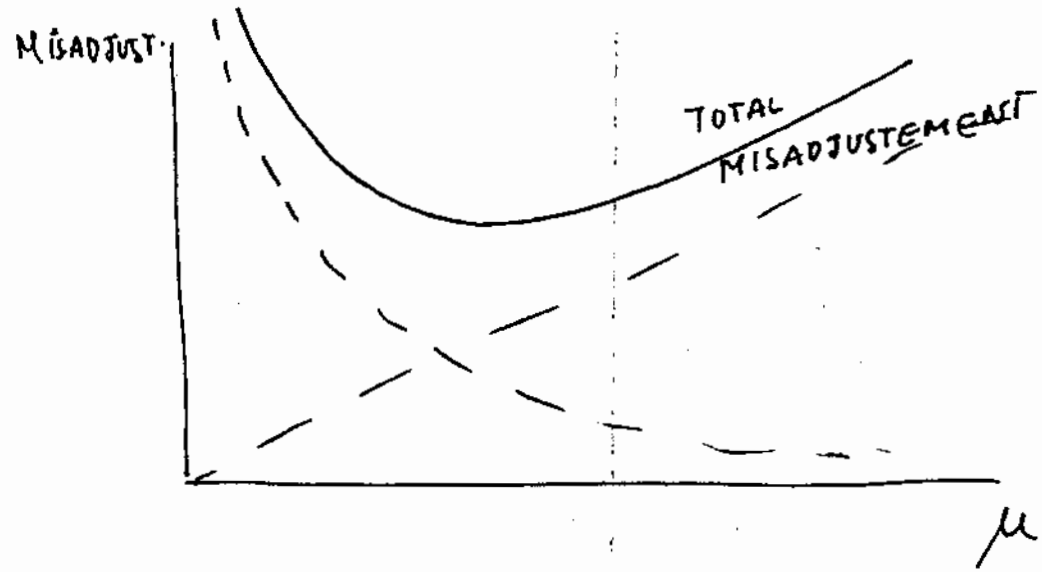
EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

In this case we have two error sources for the actual weights. One that is related to the errors of the estimate of the gradient. The other is related to the lag in the adaptive process.

It can be shown (Widrow and al 1976) that the first contribution is directly proportinal to mu. And that the second contribution is inversely proportional to mu. We can assume them addtive, so the total misadjustment has a single minimum.



JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

IMPLEMENTATION ISSUES- ROUNDOFF NOISE.

It is a known fact that the finite precision of the fixed point computer arithmetic causes errors in digital filter implementations. This error can be modelled as additive white noise, with a variance equal to weight of the least significant bit kept in the calculations.

This is difficult to model, but for fixed coefficients digital filters, we know how to estimate the roundoff noise level at the output. This noise will degradate the output dynamic range.

What I would like to address is the effect of fixed point arithmatic in the adaptation process.

In the LMS we have

Therefore in the weight update equation, we have a product of the input by the error, which must be further reduced by mu. Mu is generally small, which means that we may get a truncated value of zero to update the weights..., i.e. no update.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

We say that the weight stalls. This is responsible for excess mean square error.

So the theoretical result that the smaller the mu, the smaller the misadjustment is no longer valid.

This effect is dependent on the wordlength of the computation of the update equation. The longer the wordlength of the accumulator to keep the results (and the weigths) the better we are.

This means practically two things:

- The precision of the coefficients for the update equation should be larger than the one used for the filtration. An example is to use double precision for the weight computations and only use the MSB to implement the filter.
- Block computation minimizes this problem.

JOSE C. PRINCIPE

UNIVERSITY OF FLORIDA

EEL 6935- SPRING 90

904-392-2662

principe@brain.ee.ufl.edu

PROBLEMS WITH LMS

1. It is slow to converge for large eigenvalue spread.
2. It uses a noisy gradient estimate (stochastic gradient).
3. It "rattles".
4. The dependency on signal power is bothersome.

How can we improve on these aspects?

1. Speed - go directly to W* (LMS Newton)
2. Rattle - Uncouple error from the coefficients (orthogonalize error at each step, not only after adaptation)--- Lattice structures.
3. Compute the Optimum, but instead of doing it algebraicly, do it recursively (Recursive Least Squares).