Maschine Learning Algorithms

Gradient Algorithms

Markus Rupp

3.9.2020

Wiener Solution

- Consider a identification problem:

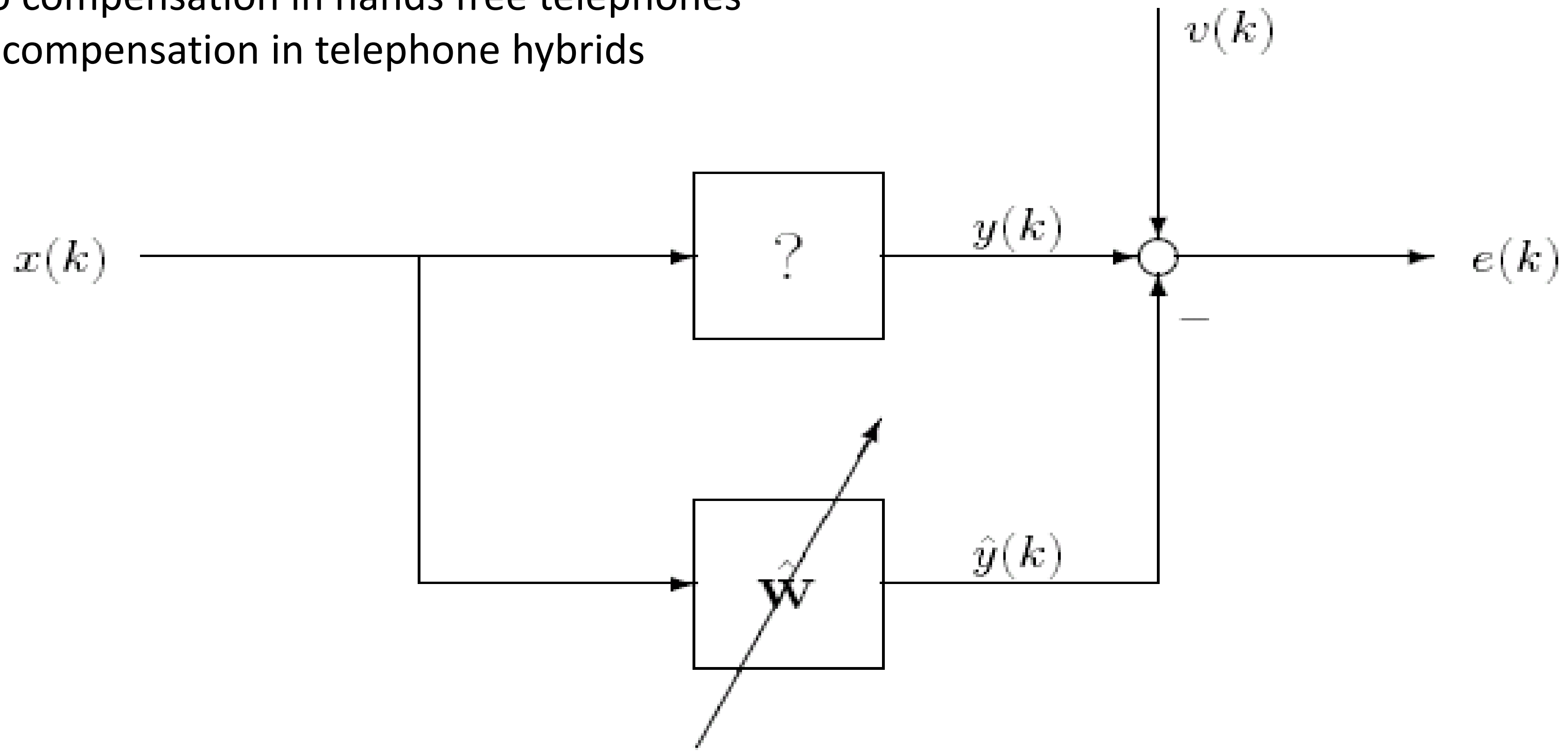
$$\mathbf{d} = \underline{w}^T \underline{\mathbf{x}} + \mathbf{v} = \underline{\mathbf{x}}^T \underline{w} + \mathbf{v}$$

- We observe the random vector $\underline{\mathbf{x}}$ and the output **d** (desired) of the linear system and we wish to estimate $\underline{w}$. The additive noise component **v** is statistically independent of $\underline{\mathbf{x}}$.
- With $r_{\mathbf{xd}}=E[\underline{\mathbf{x}}\mathbf{d}^*]=r_{\mathbf{d}\underline{\mathbf{x}}}^*$, the LS (LLMS) estimator for $\underline{w}$ is given by:

$$\underline{\hat{w}} = \left(R^*_{\underline{\mathbf{xx}}}\right)^{-1} \underline{r}^*_{\underline{\mathbf{x}}\mathbf{d}}$$

System Identification

Acoustic echo compensation in hands free telephones
Electric echo compensation in telephone hybrids



Univ.-Prof. Dr.-Ing. Markus Rupp

Wiener Solution

- Note that this solution can also be obtained by minimizing the MSE:

$$\frac{\partial}{\partial \underline{w}} \mathrm{E}\left[\left|\mathbf{d} - \underline{w}^T \underline{\mathbf{x}}\right|^2\right] = \frac{\partial}{\partial \underline{w}} \mathrm{E}\left[\left(\mathbf{d} - \underline{w}^T \underline{\mathbf{x}}\right)\left(\mathbf{d} - \underline{w}^T \underline{\mathbf{x}}\right)^*\right] = \underline{0}$$

$$\frac{\partial}{\partial \underline{w}}\left[\sigma_{\mathbf{d}}^2 - \underline{w}^T \underline{r}_{\underline{\mathbf{x}}\mathbf{d}} - \underline{r}_{\mathbf{d}\underline{\mathbf{x}}}^T \underline{w}^* + \underline{w}^T R_{\underline{\mathbf{xx}}} \underline{w}^*\right] =$$

$$= -\underline{r}_{\underline{\mathbf{x}}\mathbf{d}}^T + \underline{w}^H R_{\underline{\mathbf{xx}}}^* = \underline{0}^T$$

- Assuming a regular matrix $R_{\underline{\mathbf{xx}}}$, the Wiener solution is given by:

$$\underline{w}_o = \left(R_{\underline{\mathbf{xx}}}^*\right)^{-1} \underline{r}_{\underline{\mathbf{x}}\mathbf{d}}^*$$

Wiener Solution

- The corresponding MMSE is obtained by:

$$\min_{\underline{w}} \mathrm{E}\left[\left| \mathbf{d} - \underline{w}^T \underline{\mathbf{x}} \right|^2 \right] = \sigma_{\mathbf{d}}^{\mathbf{2}} - \underline{r}_{\underline{\mathbf{x}}\mathbf{d}}^H \underline{R}_{\underline{\mathbf{x}}\underline{\mathbf{x}}}^{-1} \underline{r}_{\underline{\mathbf{x}}\mathbf{d}}$$

- Note also the orthogonality relation:

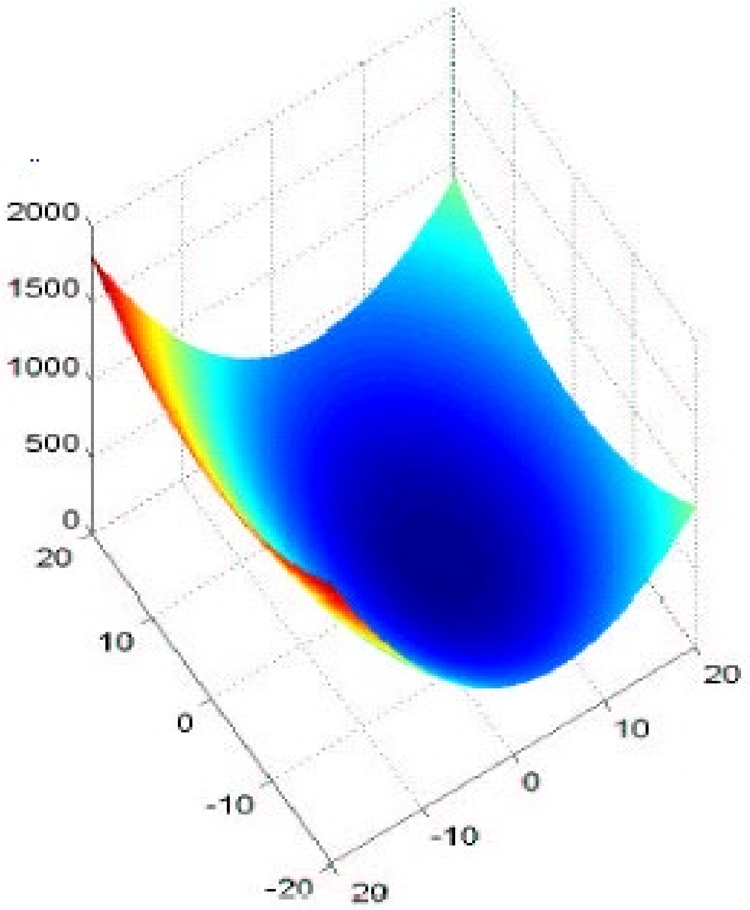
$$\mathrm{E}\left[\left(\mathbf{d} - \underline{w}^T \underline{\mathbf{x}} \right) \underline{\mathbf{x}}^H \right] = \underline{0}^T$$

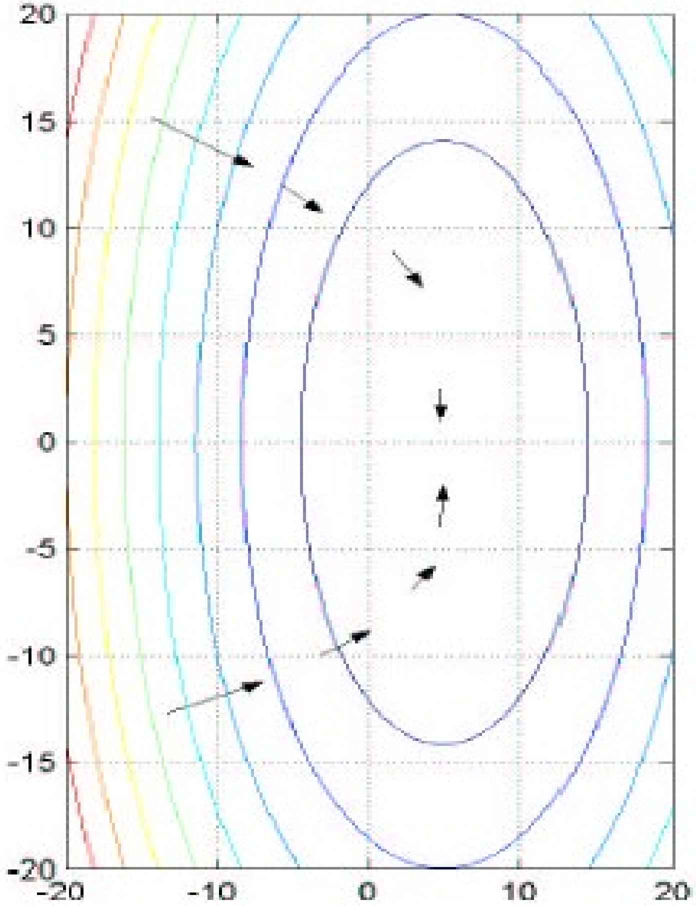
Wiener Solution

- The Wiener solution can be considered a cost function to minimize:

$$
\begin{aligned}
g(\underline{w}) &= \mathrm{E}\left[\left|\mathrm{d} - \underline{w}^T \underline{\mathrm{x}}\right|^2\right] \\
&= \sigma_{\mathrm{d}}^2 - \underline{w}^T \underline{r}_{\underline{\mathrm{x}}\mathrm{d}} - \underline{r}_{\underline{\mathrm{x}}\mathrm{d}}^H \underline{w}^* + \underline{w}^T R_{\underline{\mathrm{xx}}} \underline{w}^* \\
&= g_o + \left(\underline{w} - \underline{w}_o\right)^T R_{\underline{\mathrm{xx}}} \left(\underline{w} - \underline{w}_o\right)^* \\
g_o &= \min_{\underline{w}} \mathrm{E}\left[\left|\mathrm{d} - \underline{w}^T \underline{\mathrm{x}}\right|^2\right] \\
&= \sigma_{\mathrm{d}}^2 - \underline{r}_{\mathrm{d}\underline{\mathrm{x}}}^T R_{\underline{\mathrm{xx}}}^{-1} \underline{r}_{\underline{\mathrm{x}}\mathrm{d}} = \sigma_{\mathrm{d}}^2 - \underline{r}_{\underline{\mathrm{x}}\mathrm{d}}^H R_{\underline{\mathrm{xx}}}^{-1} \underline{r}_{\underline{\mathrm{x}}\mathrm{d}}
\end{aligned}
$$

Wiener Solution





Steepest Descent

- The Wiener solution cannot only be found by inverting a matrix but also by iterative procedures.
- Consider the following iterative procedure:

$$\underline{\hat{w}}_k = \underline{\hat{w}}_{k-1} + \mu_k \underline{z}_k ; k = 1,2,...$$

- The correct choice for the step-size μ_k and the search direction $\underline{z}_k$ would cause the cost function to decrease:

$$g(\underline{\hat{w}}_k) < g(\underline{\hat{w}}_{k-1})$$

Steepest Descent

- The cost function can be expanded into a Taylor series around $\underline{w}_{k-1}$, obtaining:

$$g(\underline{w}) = g(\underline{\hat{w}}_{k-1}) + \nabla g(\underline{\hat{w}}_{k-1})(\underline{w} - \underline{\hat{w}}_{k-1})$$
$$+ (\underline{w} - \underline{\hat{w}}_{k-1})^H \nabla^2 g(\underline{\hat{w}}_{k-1})(\underline{w} - \underline{\hat{w}}_{k-1})$$

- Since the cost function is a quadratic function, the Taylor series is correct with the three given terms. Now, g($\underline{w}$) can be evaluated at the point $\underline{w}_k$ utilizing the previous iterative procedure. We obtain:

$$g(\underline{\hat{w}}_k) = g(\underline{\hat{w}}_{k-1}) + \mu_k \nabla g(\underline{\hat{w}}_{k-1})\underline{z}_k$$
$$+ \mu_k^2 \underline{z}_k{}^H \nabla^2 g(\underline{\hat{w}}_{k-1})\underline{z}_k$$

Steepest Descent

- Gradient and Hessian of the cost function can be evaluated:

$$g(w) = \sigma_{\mathbf{d}}^2 - \underline{w}^T \underline{r}_{\underline{\mathbf{x}}\mathbf{d}} - r_{\underline{\mathbf{x}}\mathbf{d}}^H \underline{w}^* + \underline{w}^T R_{\underline{\mathbf{xx}}} \underline{w}^*$$

$$\nabla g(w) = -\underline{r}_{\underline{\mathbf{x}}\mathbf{d}}^T + \underline{w}^H R_{\underline{\mathbf{xx}}}^* = \left(\underline{w} - \underline{w}_o\right)^H R_{\underline{\mathbf{xx}}}^*$$

$$\nabla^2 g(w) = R_{\underline{\mathbf{xx}}}^*$$

- Note that the Gradient is a row vector
- Leading to the expression

$$g\left(\underline{\hat{w}}_k\right) = g\left(\underline{\hat{w}}_{k-1}\right) +$$
$$+ \mu_k \left(-r_{\underline{\mathbf{x}}\mathbf{d}} + R_{\underline{\mathbf{xx}}} \underline{\hat{w}}_{k-1}^*\right)^T \underline{z}_k + \mu_k^2 \underline{z}_k^H R_{\underline{\mathbf{xx}}}^* \underline{z}_k$$

Steepest Descent

- For such cost function

$$g\left(\underline{\hat{w}}_k\right) = g\left(\underline{\hat{w}}_{k-1}\right) +$$
$$+\,\mu_k\left(-r_{\underline{\mathbf{x}}\mathbf{d}} + R_{\underline{\mathbf{xx}}}\,\underline{\hat{w}}^*_{k-1}\right)^T \underline{z}_k + \mu_k^2\, \underline{z}_k^H R^*_{\underline{\mathbf{xx}}}\, \underline{z}_k$$

- Since $R_{\underline{\mathbf{xx}}}$ is positive definite for all non-zero $\underline{z}_k$, we require

$$\mu_k\left(-r_{\underline{\mathbf{x}}\mathbf{d}} + R_{\underline{\mathbf{xx}}}\,\underline{\hat{w}}^*_{k-1}\right)^T \underline{z}_k < 0$$

- in order to guarantee

$$g\left(\underline{\hat{w}}_k\right) < g\left(\underline{\hat{w}}_{k-1}\right)$$

Steepest Descent

- In other words the inner product of gradient and search direction must be negative (assuming positive step-sizes only).
- Many search directions are thus possible. The most interesting ones are of the form:

$$\underline{z}_k = -B\nabla g(\underline{\hat{w}}_{k-1})^H$$
$$= -B\left(-r_{\underline{\mathbf{x}}\mathbf{d}} + R_{\underline{\mathbf{xx}}}\,\underline{\hat{w}}^*_{k-1}\right)^*$$

- For any positive definite matrix B, since the inner product becomes then

$$\nabla g(\underline{\hat{w}}_{k-1})\underline{z}_k = -\nabla g(\underline{\hat{w}}_{k-1})B\nabla g(\underline{\hat{w}}_{k-1})^H < 0$$

Steepest Descent

- We can interpret our choice of direction as the direction that points in the opposite direction as the gradient, thus somewhat in direction of the minimum, however, not necessarily exactly.
- For B=I, we thus obtain the most well-known, steepest-descent iteration:

$$\underline{\hat{w}}_k = \underline{\hat{w}}_{k-1} + \mu_k \left[r_{\mathbf{d}\underline{\mathbf{x}}} - R^*_{\underline{\mathbf{x}}\underline{\mathbf{x}}} \underline{\hat{w}}_{k-1} \right]; k = 1,2,\ldots$$

- Or in general form:

  New estimate = old estimate + correction term

Steepest Descent

- Now, let us use a reference approach. The Wiener solution gives us the optimal solution $\underline{w}_o$. Utilizing such, we can rewrite the iterations as:

$$\underline{\hat{w}}_k = \underline{\hat{w}}_{k-1} + \mu_k R^*_{\underline{\mathbf{xx}}} \underbrace{\left(\underline{w}_o - \underline{\hat{w}}_{k-1}\right)}_{\underline{\tilde{w}}_{k-1}}; k = 1, 2, \ldots$$

- Reformulating in terms of the parameter error vector, we obtain:

$$\begin{aligned}\underline{\tilde{w}}_k &= \underline{\tilde{w}}_{k-1} - \mu_k R^*_{\underline{\mathbf{xx}}} \underline{\tilde{w}}_{k-1} \\ &= \left(I - \mu_k R^*_{\underline{\mathbf{xx}}}\right) \underline{\tilde{w}}_{k-1}; k = 1,2,\ldots\end{aligned}$$

Steepest Descent

- Since $R_{\underline{\mathbf{xx}}}$ can be diagonalized using a unitary matrix Q:

$$QR^*_{\underline{\mathbf{xx}}}Q^H = \Lambda$$

- The update iteration can be diagonalized as well:

$$\underline{\tilde{u}}_k = Q\underline{\tilde{w}}_k$$

$$\underline{\tilde{u}}_k = \left(I - \mu_k \Lambda\right)\underline{\tilde{u}}_{k-1}$$

$$\left(\underline{\tilde{u}}_k\right)_i = \left(1 - \mu_k \lambda_i\right)\left(\underline{\tilde{u}}_{k-1}\right)_i$$

- i, indicating the i-th entry of the vector $\underline{u}_k$.

Steepest Descent

- We now have the opportunity to find the convergence condition of the steepest-descent iteration:

$$\left|1-\mu_k\lambda_i\right|<1$$

- Which must be true for all eigenvalues λ_i. Equivalently, the condition can be reformulated for the step size μ_k:

$$0<\mu_k<\frac{2}{\lambda_{\max}}\leq\frac{2}{\lambda_i}$$

Steepest Descent

- The step-size μ_k obviously plays the role of a convergence rate factor. Once μ_k is very small, the term

$$\left|1-\mu_k\lambda_i\right|$$

will be close to one and thus the cost function will decrease only slowly. For larger values of μ_k the convergence rate will be higher and finally for even larger values, the rate will decrease again.

Steepest Descent

- Until now, we only considered quadratic cost functions. The steepest-descent iterations are, however, not limited to such cost functions. Let us consider an arbitrary non-linear cost function g(w):

$$
\begin{aligned}
g(\underline{w}) &= g(\underline{w}_{k-1}) + \\
&+ \nabla g(\underline{w}_{k-1})(\underline{w} - \underline{w}_{k-1}) + (\underline{w} - \underline{w}_{k-1})^H \nabla^2 g(\underline{w}_{k-1})(\underline{w} - \underline{w}_{k-1}) + \ldots
\end{aligned}
$$

- Three terms are not sufficient now to describe the behavior accurately.

Steepest Descent

- However, the previous condition

$$\mu_k \left(-r_{\underline{\mathbf{x}}\mathbf{d}} + R_{\underline{\mathbf{x}}\underline{\mathbf{x}}} \underline{\hat{w}}_{k-1}^{*} \right)^T \underline{z}_k < 0$$

- cannot guarantee any more the convergence of the iterations.
- In general, the iterations will converge inside a small area to a fixed-point. This is a local minimum. It will not necessarily be identical to the global (desired) minimum.

Eigenvalue-analysis

- **<u>Properties of Hermitian matrices (autocorrelation)</u>**
- 1) $R^k \underline{q} = \lambda^k \underline{q}$
  - If λ is an eigenvalue of R, then λ^k is an eigenvalue of R^k.
  - R^k and R share the same eigenvectors

Eigenvalue-analysis

- Properties:
  - 2) The corresponding eigenvectors $\underline{q}_i$ to two distinct eigenvalues λ_i are linearly independent.
  - <u>Linear Independency</u> requires that there exists factors v_i unequal to zero, so that

$$\sum_{i=1}^{M} v_i \underline{q}_i = \underline{0}$$

Eigenvalue-analysis

- Proof by contradiction: assume, at least one of the v_i is not zero

Multiple Multiplication with R :

$$\sum_{\mathrm{i}=1}^{M} v_i \lambda_i^k \underline{q}_i = \underline{0};\ k = 0,1,\ldots,M-1$$

$$\left[v_1 \underline{q}_1, v_2 \underline{q}_2, \ldots, v_M \underline{q}_M\right] S = \underline{0}$$

$$S = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \ldots & \lambda_1^{M-1} \\ 1 & \lambda_2 & \lambda_2^2 & \ldots & \lambda_2^{M-1} \\ \vdots & & & & \vdots \\ 1 & \lambda_M & \lambda_M^2 & \ldots & \lambda_M^{M-1} \end{bmatrix};\ S^{-1}\textbf{exists}$$

$$\left[v_1 \underline{q}_1, v_2 \underline{q}_2, \ldots, v_M \underline{q}_M\right] = \underline{0} \Rightarrow \textbf{all}\ v_i = 0$$

Eigenvalue-analysis

- Note: Every vector $\underline{w}$ can be formed by a linear combination of eigenvectors, as long as they are of the same dimension:

$$\sum_{i=1}^{M} v_i \underline{q}_i = \underline{w}$$

- Thus:

$$\sum_{i=1}^{M} v_i \lambda_i \underline{q}_i = R\underline{w}$$

- The eigenvectors build a basis of the vectorspace with dimension M.

Eigenvalue-analysis

- Properties
  - 3) An acf matrix is
    - (1) not negative definite and
    - (2) all eigenvalues of a Hermitian matrix R are real-valued und non-negative!
  - **Proof:**

$$\text{Let}: \ \mathrm{y} = \underline{a}^H \underline{\mathrm{x}}$$

$$\mathrm{E}[|\,\mathrm{y}\,|^2] = \mathrm{E}[\underline{a}^H \underline{\mathrm{x}}\underline{\mathrm{x}}^H \underline{a}] = \underline{a}^H R_{\mathrm{xx}} \underline{a} \geq 0$$

Eigenvalue-analysis

- **Proof** (Part 2):

$$R\underline{q}_i = \lambda_i \underline{q}_i$$

$$\underline{q}_i^H R\underline{q}_i = \lambda_i \underline{q}_i^H \underline{q}_i$$

$$\lambda_i = \frac{\underline{q}_i^H R\underline{q}_i}{\underline{q}_i^H \underline{q}_i} \geq 0; \textbf{since } R \textbf{ is non-negative definite}$$

Eigenvalue-analysis

- Properties:
  - 4) If all eigenvalues are different, then all eigenvectors build an orthogonal basis.
  - **Proof:**

$$R\underline{q}_i = \lambda_i \underline{q}_i, \quad R\underline{q}_j = \lambda_j \underline{q}_j$$

$$\underline{q}_j^H R\underline{q}_i = \lambda_i \underline{q}_j^H \underline{q}_i, \quad \underline{q}_i^H R\underline{q}_j = \lambda_j \underline{q}_i^H \underline{q}_j$$

$$0 = (\lambda_j - \lambda_i)\underline{q}_i^H \underline{q}_j$$

  - If the eigenvectors are normalized, they build an orthonormal basis.

Eigenvalue-analysis

- Properties: Unitary Transformation
  - 5) Build a matrix Q out of the eigenvectors. This matrix can be diagonalized: $Q^H R Q=\Lambda$.
  - **Proof:**

$$R\left[\underline{q}_1,\underline{q}_2,\ldots,\underline{q}_M\right]=\left[\lambda_1\underline{q}_1,\lambda_2\underline{q}_2,\ldots,\lambda_M\underline{q}_M\right]$$

$$=\left[\underline{q}_1,\underline{q}_2,\ldots,\underline{q}_M\right]\Lambda=Q\Lambda$$

$$RQ=Q\Lambda$$

Eigenvalue-analysis

- If this matrix Q stems from normalized eigenvectors, then Q is „unitary“, i.e., $Q^H Q = I$.
  - **Proof:** by orthononal and orthonormal property of eigenvectors

$$\underline{q}_i^H \underline{q}_j = \begin{cases} 1 & ; i = j \\ 0 & ; \textbf{else} \end{cases}$$

Eigenvalue-analysis

- Properties
  - 6) The trace of a Hermitian matrix equals the sum of its eigenvalues
  - **Proof:**

$$
\begin{aligned}
\sum_{i=1}^{M} \lambda_i &= \mathrm{trace}(\Lambda) \\
&= \mathrm{trace}\left(Q^H R Q\right) \\
&= \mathrm{trace}\left(R Q Q^H\right) = \mathrm{trace}(R)
\end{aligned}
$$

Eigenvalue-analysis

- **Cayley-Hamilton Theorem:** every matrix R satisfies its own characteristic equation:

$$\det(R-\lambda I)=0$$

$$\lambda^M + a_1\lambda^{M-1} + \ldots + a_{M-1}\lambda + a_M = 0$$

$$\Lambda^M + a_1\Lambda^{M-1} + \ldots + a_{M-1}\Lambda + a_M I = 0$$

$$R^M + a_1 R^{M-1} + \ldots + a_{M-1}R + a_M I = 0$$

Eigenvalue-analysis

- **Proof:** Diagonalize the matrix equation:

$$R^M + a_1 R^{M-1} + \ldots + a_{M-1} R + a_M I = \mathbf{0}$$

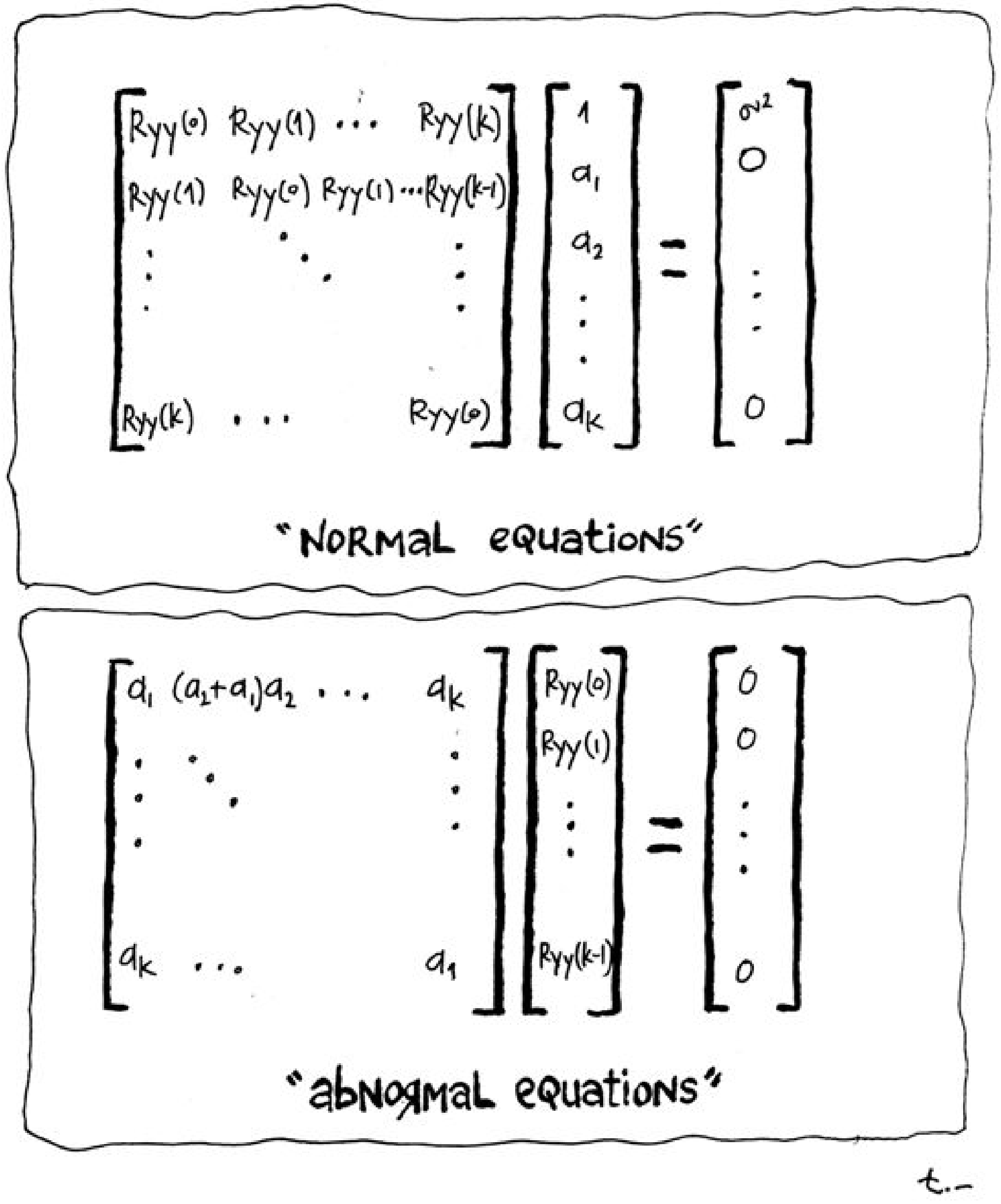
$$Q^H \left[R^M + a_1 R^{M-1} + \ldots + a_{M-1} R + a_M I \right] Q = \mathbf{0}$$

$$\Lambda^M + a_1 \Lambda^{M-1} + \ldots + a_{M-1} \Lambda + a_M I = \mathbf{0}$$

- Interesting add-on:

$$R^M = -a_1 R^{M-1} - \ldots - a_{M-1} R - a_M I$$

Polynomials of degree $\geq M$ can be described as polynomials of lesser degree.

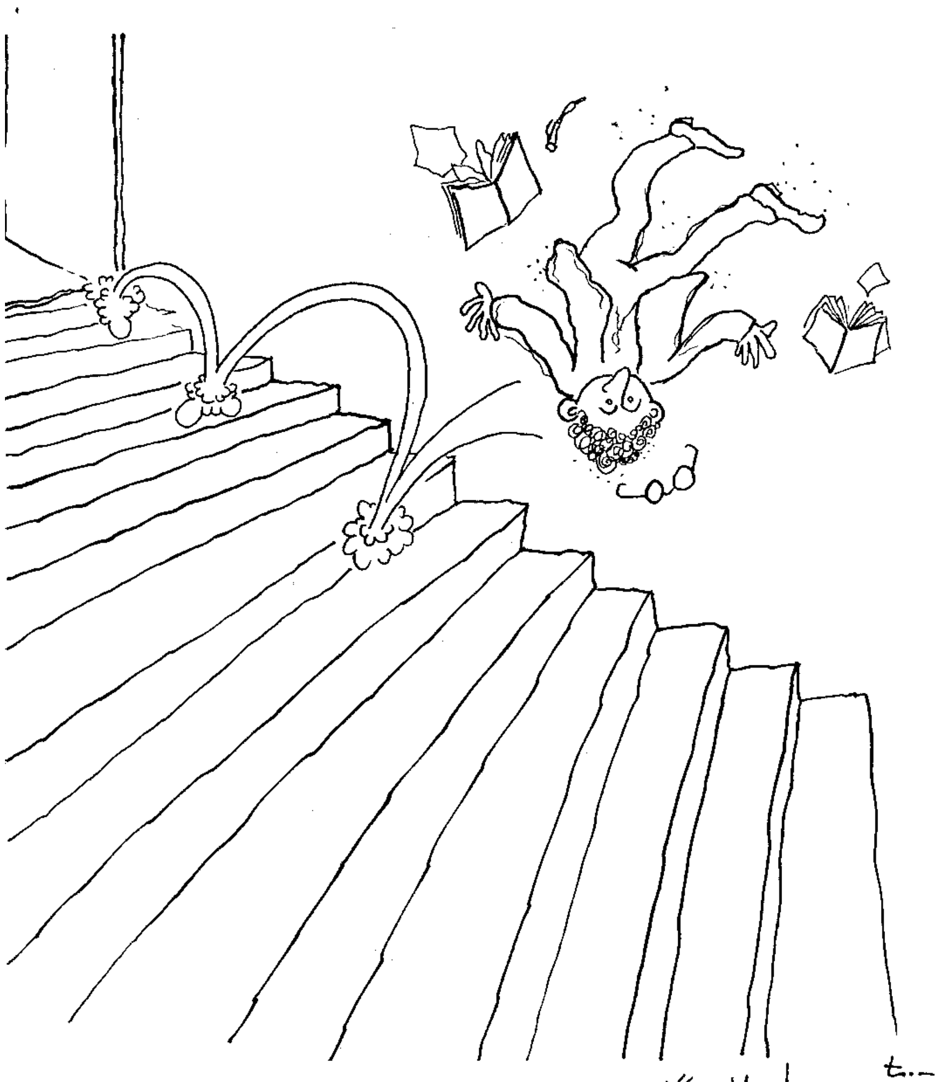


"NORMAL EQUATIONS"
"abNORMAL EQUATIONS"
t.-

Steepest Descent

Humor in DSP:

www.eurasip.org



Just after learning the "Steepest Descent" method in optimization class...

Further Gradient Approaches

- We have mostly only concentrated on the first (gradient) term but note

$$g\left(\underline{w}\right)=g\left(\underline{w}_{k-1}\right)+\nabla g\left(\underline{w}_{k-1}\right)\left(\underline{w}-\underline{w}_{k-1}\right)+$$
$$+\left(\underline{w}-\underline{w}_{k-1}\right)^{H}\nabla^{2}g\left(\underline{w}_{k-1}\right)\left(\underline{w}-\underline{w}_{k-1}\right)+\ldots$$

- Including the second (quadratic) term, results in the so called Newton type gradient algorithm, which
  - Is much faster in convergence speed
  - Can find a solution of a quadratic problem in a single step
  - Is of higher complexity
  - Approximates complicated error functions by a quadratic one…
  - Does not resolve the issue of local minima.

Newton‘s Approach

- In order to take advantage of the quadratic form

$$g\left(\underline{w}\right)=g\left(\underline{w}_{k-1}\right)+\nabla g\left(\underline{w}_{k-1}\right)\left(\underline{w}-\underline{w}_{k-1}\right)+$$
$$+\left(\underline{w}-\underline{w}_{k-1}\right)^{H}\nabla^{2}g\left(\underline{w}_{k-1}\right)\left(\underline{w}-\underline{w}_{k-1}\right)+\ldots$$

- An iterative update would look like

$$\underline{\hat{w}}_{k}=\underline{\hat{w}}_{k}-\left[\nabla^{2}g\left(\underline{\hat{w}}_{k-1}\right)\right]^{\#}\nabla g\left(\underline{\hat{w}}_{k-1}\right)$$

- With # denoting the pseudo inverse if for some reasons the inverse does not exist.
- Often an additional step-size (μ<1) is applied.

Newton's Approach

- The update then reads

$$\underline{\hat{w}}_k = \underline{\hat{w}}_{k-1} + \mu_k \left(R^*_{\underline{x}\,\underline{x}} \right)^{-1} \left[r_{d\underline{x}} - R^*_{\underline{x}\,\underline{x}} \underline{\hat{w}}_{k-1} \right]; k = 1, 2, \ldots$$

- Applying the same analysis as before, we now find

$$\begin{aligned} \underline{\tilde{w}}_k &= \underline{\tilde{w}}_{k-1} - \mu_k \left(R^*_{\underline{xx}} \right)^{-1} R^*_{\underline{xx}} \underline{\tilde{w}}_{k-1} \\ &= \left(1 - \mu_k \right) \underline{\tilde{w}}_{k-1}; k = 1, 2, \ldots \end{aligned}$$

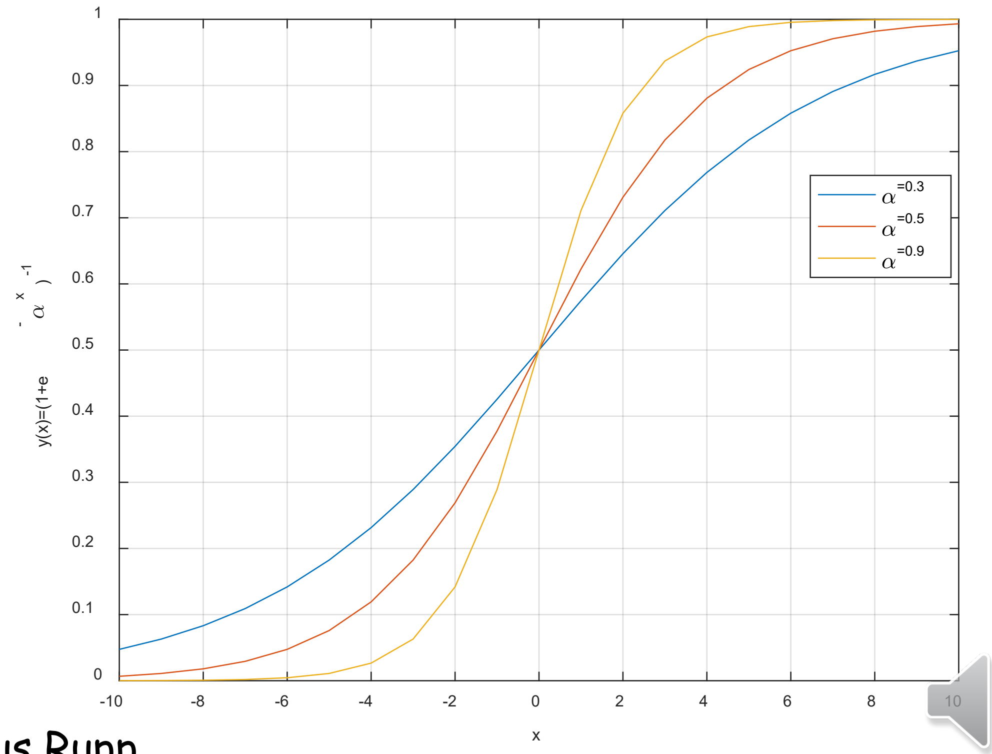
- which converges for $0<\mu_k<2$ and shows fastest learning for $\mu_k=1$.

Further Gradient Approaches

- If in general, an arbitrary cost function g($\underline{w}$) is given, we can always compute its derivative and use a gradient descent approach.
- Example: sigmoid function

$$g(\underline{w}) = \frac{1}{1+e^{-\alpha \underline{x}^H \underline{w}}}$$

$$\nabla g(\underline{w}) = \frac{\partial g(\underline{w})}{\partial \underline{w}} = \alpha g(\underline{w})\left(1-g(\underline{w})\right)$$



Further Gradient Approaches

- To avoid local minima(maxima), a very helpful property of a cost function is convexity (concavity)
- A twice differentiable function g(x) is convex, if and only if

$$g''(x) \geq 0$$

- Or, equivalently,

$$\nabla^2 g(x) \geq \mathbf{0}$$

- has non-negative eigenvalues for all arguments

Further Gradient Approaches

- Convex functions:

$$g(x) = x^2 \quad \rightarrow \quad g''(x) = 2 \geq 0$$

$$g(x) = e^x \quad \rightarrow \quad g''(x) = e^x \geq 0$$

$$g(x) = -\log(x)\,\mathrm{U}(x) \quad \rightarrow \quad g''(x) = \frac{1}{x^2}\,\mathrm{U}(x) \geq 0$$

- Note that the sigmoid function is non convex