Estimating the covariance matrix

The *covariance matrix* (central moments) can be estimated from an ensemble of observations $\{\mathbf{x}_i\}$:

$$\mathbf{C}_{\mathbf{xx}} = E\{(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})^T\} = \mathbf{R}_{\mathbf{xx}} - \boldsymbol{\mu}_{\mathbf{x}}\boldsymbol{\mu}_{\mathbf{x}}^T$$

estimated value: $\hat{\mathbf{C}}_{\mathbf{xx}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_{\mathbf{x}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_{\mathbf{x}})^T$

and estimated value: $\hat{\boldsymbol{\mu}}_{\mathbf{x}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$

$\hat{\mathbf{C}}_{\mathbf{xx}}$ is calculated from the sum of matrices of degree 1:

$(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_{\mathbf{x}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_{\mathbf{x}})^T$

since in the dyadic product only multiples of the row vector

$(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_{\mathbf{x}})^T$ occur, because:

$$\mathbf{x}\mathbf{y}^T = \begin{bmatrix} x_1 \cdot \mathbf{y}^T \\ x_2 \cdot \mathbf{y}^T \\ \vdots \\ x_N \cdot \mathbf{y}^T \end{bmatrix}$$

Problem of high feature dimensionality

$\hat{\mathbf{C}}$ is thus singular, if less than $n=N$, with $N=\dim(\mathbf{x})$, independent observations of the ensemble are available!!

This is a problem, if the number of features is very high and only few samples of the ensemble are available.

Also $\hat{\mathbf{C}}^{-1}$ is needed!

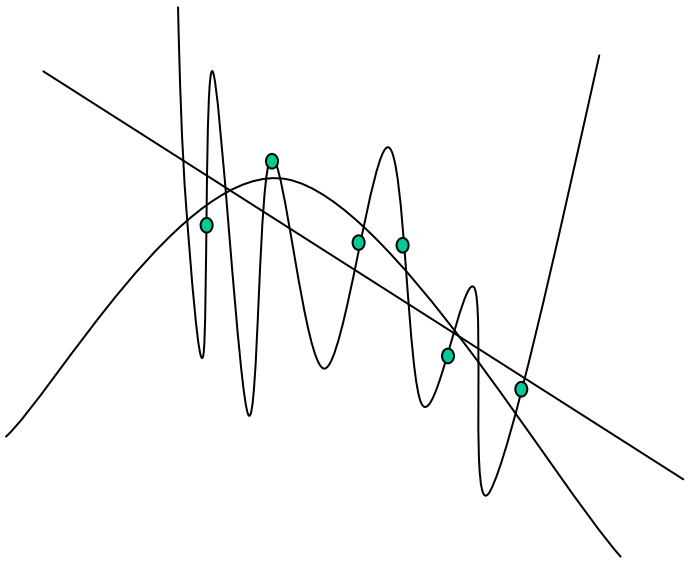
The estimation can be significantly improved with $n>>N$.

What to do, if the number of samples is too little? One could

- reduce the number of feature using a KLT, or
- assuming the features are not correlated all auxiliary diagonal elements are set to zero, which forces invertibality. Even though this procedure is actually incorrect, the results of this heuristics are usually good.

The problem of few samples

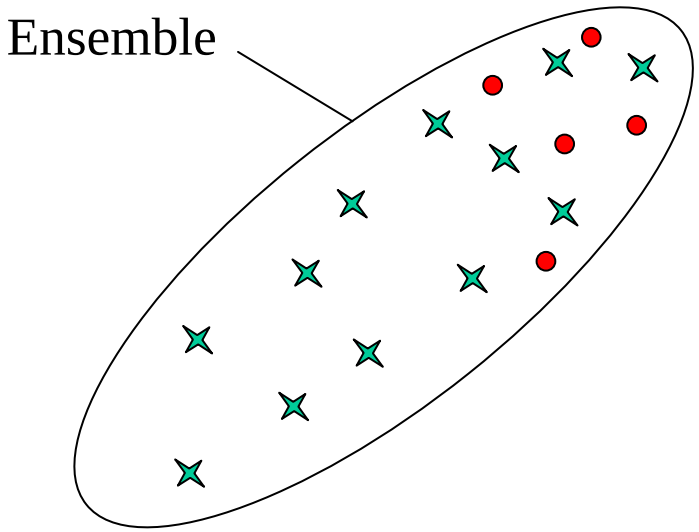
The resulting classifier assuming statistical independency is for sure suboptimal. This relates to the problem of insufficient samples. It can be compared to the problem of curve-fitting. The images shows 6 data points and different polynoms for fitting. The data points were created adding mean value free, independent noise to a ***parabola***. That is why a parabel should result in the best fit, assuming that more samples are added to supplement/complete the 6 points.



curve approximation to a set of points

- The *line* results in a *feasible* estimation.
- The *parabel* results in a *better* approximation, but in question is still, whether the sample was appropiate to locate the parabel. The parabel for a higher number of samples could be somewhere different and in the inspected interval the line could be a better estimation.
- The polynomial of 10th degree results in a perfect fit. But such an underdetermined estimation cannot be expected to approximate new samples in a good way. Many more samples are needed to get a similarly good approximation of a polynom of 10th degree compared to a parabel fit, despite the fact, that the latter case is a special case (n=2) of first.

Rule: the fewer the samples the simpler the model



- sample 1 (representative)
- sample 2 (not representative)

Generally applies: Reliable inter- and extrapolation can only be expected for highly over-determined solutions (sufficiently high number of samples).

If an exact statistical model would have been given, our problem could be solved with the MAP-approach. In practice there is the problem of finding a good classifier from a finite number of samples.

The problem of generalizing capability of a classifier

How does a classifier, which is built on a finite set of samples, react to new experiments (problem of inter- and extrapolation)?

That is why we need to distinguish between a *training- (learning-)* and a *testing set.*

The checking of the capability only based on the learning set is called *reclassification* (which can lead to an ideal fit) and the checking based on an independent test data set is called *generalization* (inter- and extrapolation capability).

The higher the number of parameter of the estimation function used in the classification, the higher the number of the samples of the training set has to be.

Recursive estimation of statistical parameter

In case during a recognition task new samples are added, *recursive* estimation of the paramaters can be beneficial. This produces much less cost than calculating the basic equations over and over again using the enhanced samples (learnig or adaptive approach, *batch* estimate versus *recursive* estimate).

For estimation of expected value applies:

$$\hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k = \frac{1}{n}\left(\sum_{k=1}^{n-1}\mathbf{x}_k + \mathbf{x}_n\right)$$
$$= \left(1-\frac{1}{n}\right)\hat{\mu}_{n-1} + \frac{1}{n}\mathbf{x}_n = \hat{\mu}_{n-1} + \frac{1}{n}\left(\mathbf{x}_n - \hat{\mu}_{n-1}\right)$$

The estimate is altered proportionally to deviation between present estimate and present observation with each step.

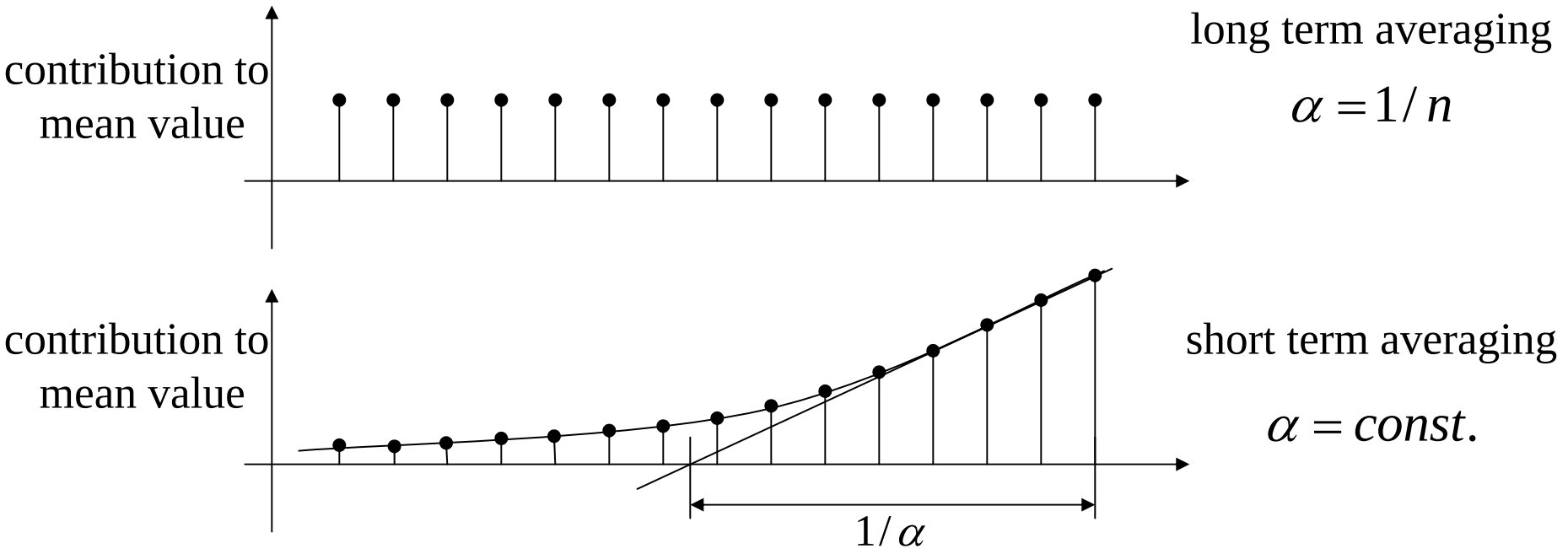
Generalizing the recursion above results in:

$$\hat{\mu}_n = \hat{\mu}_{n-1} + \alpha\left(\mathbf{x}_n - \hat{\mu}_{n-1}\right) = \left(1-\alpha\right)\hat{\mu}_{n-1} + \alpha\mathbf{x}_n$$

with: $\alpha = \begin{cases} 1/n & \text{stationary} \\ const. & \text{quasi stationary} \end{cases}$

With $\alpha=1/n$ *stationary* relations are assumed, i.e. all observations have the same weight independent on the time of occurence, which means that later observations are as important as first observations.

At $\alpha=const.$ a *fluctuation* is accepted, i.e. the newer observations have a higher weight than the old ones (exponential smoothing). The observation frame is approximately given through $1/\alpha$ with $\alpha=const.$



Recursive estimate of the covariance matrix

For *correlation matrix* (2nd moments) this recursion results:

$$\boxed{\hat{\mathbf{R}}_n = (1-\alpha)\hat{\mathbf{R}}_{n-1} + \alpha \mathbf{x}_n \mathbf{x}_n^T}$$

For recursive calculation of *covariance matrix* $\hat{\mu}$ is needed, which can be calculated by a second recursion

$$\begin{aligned}
\hat{\mathbf{C}}_n &= \hat{\mathbf{R}}_n - \hat{\mu}_n \hat{\mu}_n^T \\
&= [(1-\alpha)\hat{\mathbf{R}}_{n-1} + \alpha \mathbf{x}_n \mathbf{x}_n^T] - [(1-\alpha)\hat{\mu}_{n-1} + \alpha \mathbf{x}_n][(1-\alpha)\hat{\mu}_{n-1} + \alpha \mathbf{x}_n]^T \\
&= (1-\alpha)\hat{\mathbf{R}}_{n-1} + \alpha \mathbf{x}_n \mathbf{x}_n^T - (1-\alpha)^2 \hat{\mu}_{n-1}\hat{\mu}_{n-1}^T - \alpha(1-\alpha)[\hat{\mu}_{n-1}\mathbf{x}_n^T + \mathbf{x}_n \hat{\mu}_{n-1}^T] - \alpha^2 \mathbf{x}_n \mathbf{x}_n^T \\
&= (1-\alpha)[\hat{\mathbf{R}}_{n-1} - \hat{\mu}_n \hat{\mu}_n^T + \alpha(\mathbf{x}_n \mathbf{x}_n^T - \hat{\mu}_{n-1}\mathbf{x}_n^T - \mathbf{x}_n \hat{\mu}_{n-1}^T + \hat{\mu}_{n-1}\hat{\mu}_{n-1}^T] \\
&= (1-\alpha)[\hat{\mathbf{C}}_{n-1} + \alpha(\mathbf{x}_n - \hat{\mu}_{n-1})(\mathbf{x}_n - \hat{\mu}_{n-1})^T]
\end{aligned}$$

Recursive estimate of the covariance matrix

Both recursions together:

$$\boxed{\begin{aligned}\hat{\mathbf{C}}_n &= (1-\alpha)[\hat{\mathbf{C}}_{n-1} + \alpha(\mathbf{x}_n - \hat{\mu}_{n-1})(\mathbf{x}_n - \hat{\mu}_{n-1})^T] \\ \hat{\mu}_n &= (1-\alpha)\hat{\mu}_{n-1} + \alpha\mathbf{x}_n\end{aligned}}$$

Recursive estimate of the inverse correlation matrix

For calculating the Mahalanobis-distance a recursion for the inverse covariance matrix is needed, without inverting the matrix additionally!

With the following statement about matrix inversion:

$$(\mathbf{I}+\mathbf{A}\mathbf{B}^T)^{-1}=\mathbf{I}-\mathbf{A}(\mathbf{I}+\mathbf{B}^T\mathbf{A})^{-1}\mathbf{B}^T$$

a recursion for the inverse correlation matrix results:

$$\begin{aligned}\hat{\mathbf{R}}^{-1} &= [(1-\alpha)\hat{\mathbf{R}}_{n-1}+\alpha\mathbf{x}_n\mathbf{x}_n^T]^{-1}\\ &= \frac{1}{(1-\alpha)}\hat{\mathbf{R}}_{n-1}^{-1}-\frac{1}{(1-\alpha)^2}\hat{\mathbf{R}}_{n-1}^{-1}\mathbf{x}_n\left(\frac{1}{\alpha}+\frac{1}{(1-\alpha)}\mathbf{x}_n^T\hat{\mathbf{R}}_{n-1}^{-1}\mathbf{x}_n\right)^{-1}\mathbf{x}_n^T\hat{\mathbf{R}}_{n-1}^{-1}\\ &= \frac{1}{(1-\alpha)}\left(\hat{\mathbf{R}}_{n-1}^{-1}-\alpha\frac{\hat{\mathbf{R}}_{n-1}^{-1}\mathbf{x}_n\mathbf{x}_n^T\hat{\mathbf{R}}_{n-1}^{-1}}{1+\alpha(\mathbf{x}_n^T\hat{\mathbf{R}}_{n-1}^{-1}\mathbf{x}_n-1)}\right)\end{aligned}$$

Recursive estimate of the inverse correlation matrix

And for inverse covariance matrix:

$$\begin{aligned}\hat{\mathbf{C}}_n^{-1} &= [(1-\alpha)\hat{\mathbf{R}}_{n-1} + \alpha \mathbf{x}_n \mathbf{x}_n^T]^{-1} \\ &= \frac{1}{(1-\alpha)}[\hat{\mathbf{C}}_{n-1} + \alpha(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{n-1})(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{n-1})^T]^{-1} \\ &= \frac{1}{(1-\alpha)}\left(\hat{\mathbf{C}}_{n-1}^{-1} - \alpha \frac{\hat{\mathbf{C}}_{n-1}^{-1}(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{n-1})(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{n-1})^T \hat{\mathbf{C}}_{n-1}^{-1}}{1 + \alpha(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{n-1})^T \hat{\mathbf{C}}_{n-1}^{-1}(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{n-1})}\right)\end{aligned}$$

Recursive learning can of course be combined with pattern classification. The system is being improved for additional samples. This implies „*labelling*“ for classes though (*supervised learning*), i.e. the human observer decides on the class.