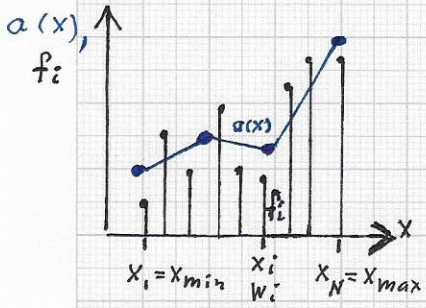
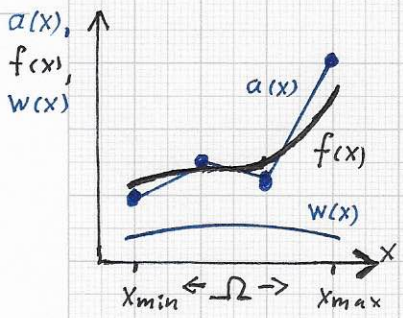
Stratovan

■ GENERATION OF SYNTHETIC DATA – Cont'd.

• Weighted best approximation:



DISCRETE CASE
Input: x_i, f_i, w_i
Output: $a(x)$



CONTINUOUS CASE
Input: $f(x), w(x)$
Output: $a(x)$

The computation of the unknown coefficients c_k of the weighted best approximation $a(x) = \sum_{k=1}^{K} c_k b_k(x)$ involves the solution of a linear system of equations, defined via inner products:

$$\begin{bmatrix} \langle b_1, b_1 \rangle & \dots & \langle b_1, b_K \rangle \\ \vdots & & \vdots \\ \langle b_K, b_1 \rangle & \dots & \langle b_K, b_K \rangle \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_K \end{bmatrix} = \begin{bmatrix} \langle f, b_1 \rangle \\ \vdots \\ \langle f, b_K \rangle \end{bmatrix} . \quad (*)$$

The inner products $\langle \cdot, \cdot \rangle$ must be computed for either the discrete or continuous case.

i) Discrete case:

$$\langle b_I, b_J \rangle = \sum_{i=1}^{N} w_i \, b_I(x_i) \, b_J(x_i) ,$$

$$\langle f, b_I \rangle = \sum_{i=1}^{N} w_i \, f_i \, b_I(x_i) ,$$

$I, J = 1 \dots K.$

ii) Continuous case:

$$\langle b_I, b_J \rangle = \int_{\Omega} w(x) \, b_I(x) \, b_J(x) \, dx ,$$

$$\langle f, b_I \rangle = \int_{\Omega} w(x) \, f(x) \, b_I(x) \, dx ,$$

$I, J = 1 \dots K.$

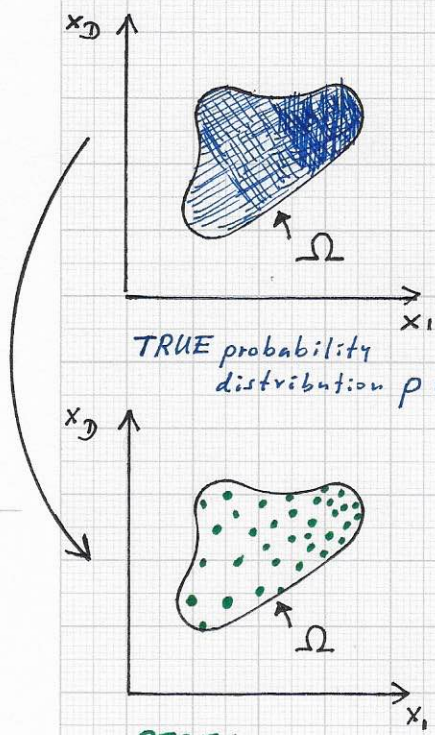
The linear system (*), $B \cdot C = ff$, is ideally defined with basis functions $b_I(x)$ with relatively small, finite domains, for more efficient computation.

The example (left) shows the case of a best approximation being a weighted best linear approximating spline with 3 linear segments (K=4). The linear spline $a(x)$ approximates the data $\{(x_i, f_i, w_i)\}_{i=1}^{N}$ in the discrete case and $f(x)$ in the continuous case.

BH

■ GENERATION OF SYNTHETIC DATA – Cont'd.

• Synthetic data & bootstrap method:



IDEAL, correct probability distribution $p(\mathbb{X})$ of a class' associated allowed feature values $\mathbb{X}$

and

IDEAL, randomly generated discrete sample S exhibiting the distribution of samples as defined by function $p(\mathbb{X})$

The generation of synthetic feature point/vector data is often done via the so-called BOOTSTRAP and MONTE CARLO methods. These methods can be used in a statistically sound way and with computational efficiency. Before describing these methods, some basic underlying assumptions are reviewed.

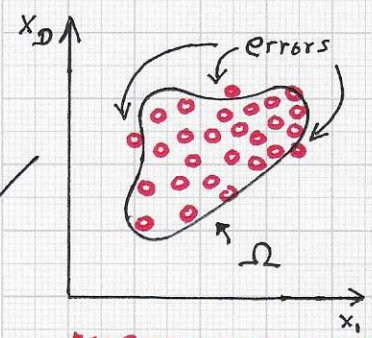
We can think of the 3D, volumetric image/scan of a specific material (i.e., a set of voxels that constitute the material, with each voxel having a particular density value).

First, a (material) class has an associated TRUE analytically defined probability distribution function $p(\mathbb{X})$, where a D-dimensional tuple $(x_1, \ldots, x_D) = \mathbb{X}$ is an allowable feature point/vector for this class, occurring with probability $p(\mathbb{X})$. The individual components x_d, $d=1 \ldots D$, of a tuple $\mathbb{X}$ can represent a specific numerical attribute of the class – either given as an original datum or derived from original data. The finite domain Ω of $p(\mathbb{X})$ can be of arbitrary shape (and topology).

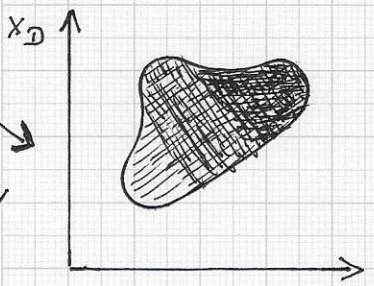
Second, when randomly sampling the feature domain Ω, with probability $p(\mathbb{X})$, one obtains an ideal, error-free, perfect discrete sample set S.

■ GENERATION OF SYNTHETIC DATA – Cont'd.

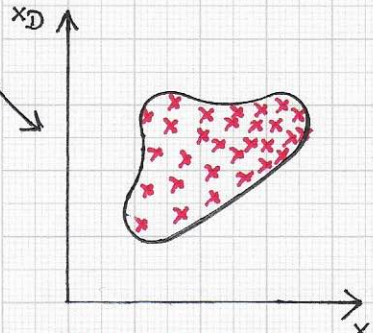
• Synthetic data & bootstrap method:



IMPERFECT, experimental sample S_E with errors



WEIGHTED BEST APPROXIMATION of imperfect probability distribution $p_E(\mathbb{X})$



SYNTHETIC feature sample set, obtained by randomly generating points from p_E

Third, when experimentally generating discrete feature point/vector data for this class, with some feature values possibly generated computationally as derived data (e.g., gradients), the resulting sample set S_E is no longer a perfect one (due to experimental and computational errors).

Fourth, assuming that the experimental sample S_E is of "sufficiently high resolution," one can perform weighted best approximation (e.g., χ^2 minimization) to define a MODEL, i.e., an analytical representation of the experiment-induced feature value probability distribution $p_E(\mathbb{X})$ – for an/the appropriate domain Ω.

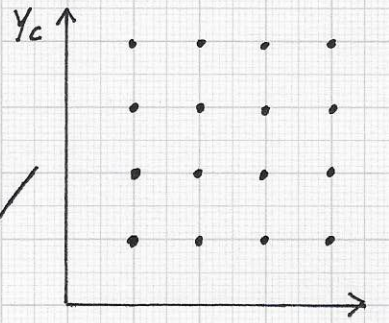
Fifth, once an analytical model (weighted best approximation, χ^2 minimization) is known, one can apply the Monte Carlo method to $p_E(\mathbb{X})$ to generate synthetic discrete sample sets. Without loss of generality, one can assume that $0 \leq p_E(\mathbb{X}) \leq 1$, $\mathbb{X} \in \Omega$. The Monte Carlo random number generator would first generate a tuple $\mathbb{X}$ in Ω (with uniform probability) and subsequently generate a value $\hat{p} \in [0,1]$; the tuple $\mathbb{X}$ would be accepted as a sample when $\hat{p} \leq p_E(\mathbb{X})$.

THIS APPROACH IS COMPUTATIONALLY INEFFICIENT!

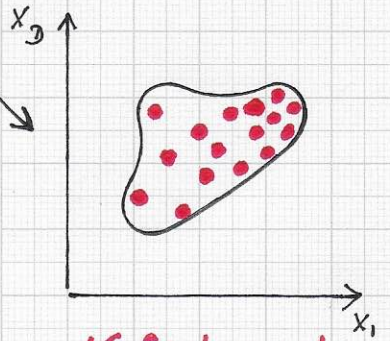
Stratovan

■ GENERATION OF SYNTHETIC DATA – Cont'd.

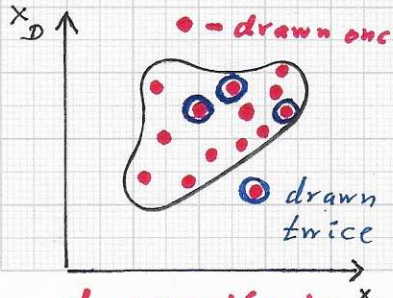
• Synthetic data & bootstrap method:



C-dimensional control parameter space, sampled uniformly with $\mathbb{y}$ tuples for "experiments" (e.g., scans)



16 feature pnts. resulting from 16 "experiments" in D-dimensional feature space



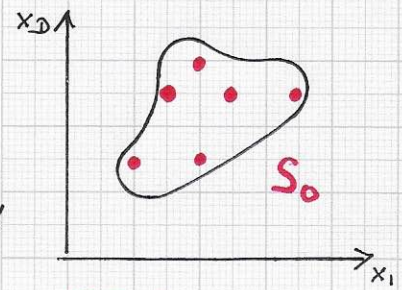
drawing 16 pnts. from feature point set, with replacement

Feature points/vectors $\mathbb{x} = (x_1, \ldots, x_D)$ can be viewed as results obtained via experiments that are "controlled" via C "control variables" $\mathbb{y} = (y_1, \ldots, y_C)$. One should assume that the $\mathbb{y}$-space, i.e., the control parameter space, is uniformly sampled in each of the C y-directions. (In other words, a feature point $\mathbb{x}$ is the multi-valued response value of a multivariate control variable $\mathbb{y}$, i.e., $\mathbb{x} = \mathbb{x}(\mathbb{y})$.) The left images show the case of experimentally (and/or computationally) generating 16 feature points from 16 "control settings." For example, the experiments could be 16 scans performed with 16 different control settings.

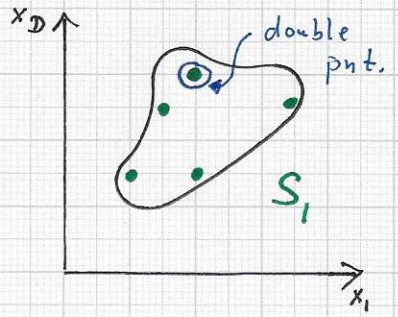
The (probability) distribution of the resulting (generally HIGH-RESOLUTION) feature point/vector data set can be used directly by the BOOTSTRAP method. The bootstrap method uses the feature point data set (with N points) as input and generates a single **synthetic** data set (with N points) by randomly drawing data from the given feature point data set – with "replacement", i.e., possibly drawing a given data point multiple times.

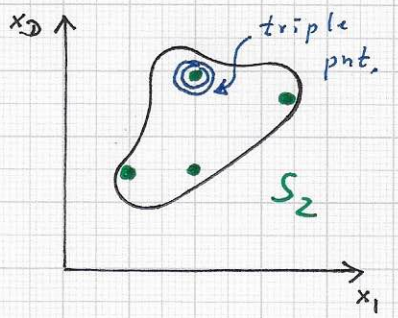
GENERATION OF SYNTHETIC DATA - Cont'd.

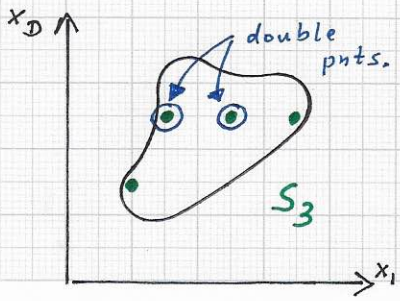
• Bootstrap method:



6 feature pnts. obtained via orig. "experiment"







3 sets of feature pnts; 6 pnts. drawn with replacement from S_0

The bootstrapping method uses a given feature point/vector data set (discrete) as input (S_0). (This data set can also be represented analytically via some probability distribution function $p_0(x)$, e.g., via χ^2 minimization or weighted best approximation.) Additional synthetic feature point/vector data sets $S_1, S_2, S_3, \ldots$ (discrete) are generated by randomly drawing N points from S_0, where $N = |S_0|$ (= cardinality of S_0). The same feature point can be drawn from S_0 multiple times ("drawing with replacement"). (Each synthetic data set S_i can also be represented via a probability distribution function $p_i(x)$.)

One can also interpret the discrete feature point data sets $S_0, S_1, S_2, S_3, \ldots$ as "delta functions" with "Dirac pulses" at the locations of the feature points (having multiplicities $1, 2, 3, \ldots$).

The percentage of feature points in the original feature point set "replaced by duplicated points" should be around $37\% \approx \frac{1}{e}$ (see statistics books...). "In the limit," if one performed a 'large' number of experiments, then the resulting feature point distributions would be statistically highly similar to those of the synthetic ones. ☞BH