PDE-Foam
A probability-density estimation method based on self-adapting phase-space binning

ACAT 2008
Erice, Sicily
3. Nov. 2008

Tancredi Carli[1], Dominik Dannheim[1], Alexander Voigt[1],
Karl-Johan Grahn[2], Peter Speckmayer[3]

[1]CERN

[2]KTH Stockholm

[3]Technische Universität Wien, now with CERN

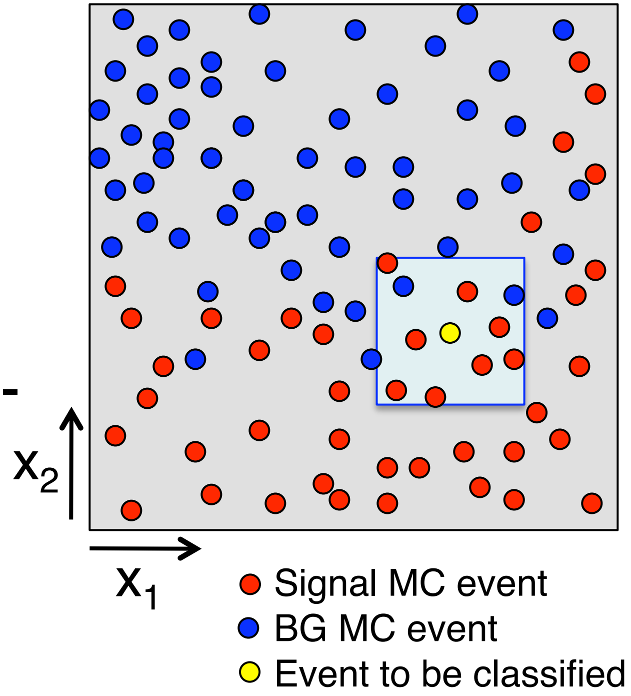
Outline

- Probability-Density Estimation (PDE) for MVA
- PDE-Foam:
  - Concept
  - Implementation in TMVA
  - Parameters
  - Performance
  - Gaussian Kernel Smoothing
  - Event Reconstruction
- Conclusions

Probability-Density Estimation (PDE)

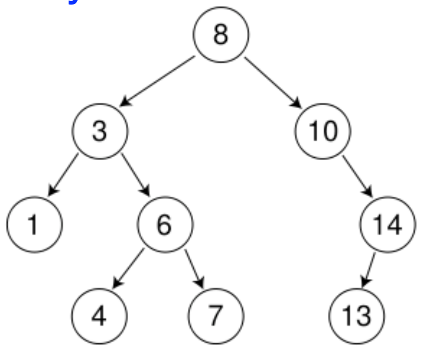
- Aim: distinguish between signal and background, given d measured properties of an event
- Estimate probability for an event x to be of signal class by Discriminant D, based on Monte Carlo simulations sampling the signal- and background event densities:

$$p(x) = \frac{\rho_{sig}(x)}{\rho_{sig}(x) + \rho_{bg}(x)} \approx D(x) := \frac{n_{sig}(x)}{n_{sig}(x) + c \cdot n_{bg}(x)}$$



- n_{sig} and n_{bg} counted in the vicinity of x (d-dimensional box around x)
- PDE-RS implementation in TMVA:
  Use *binary search trees* to store and find the Monte-Carlo sampled events (*Nucl.Instrum.Meth. A501 (2003) 576-588*)

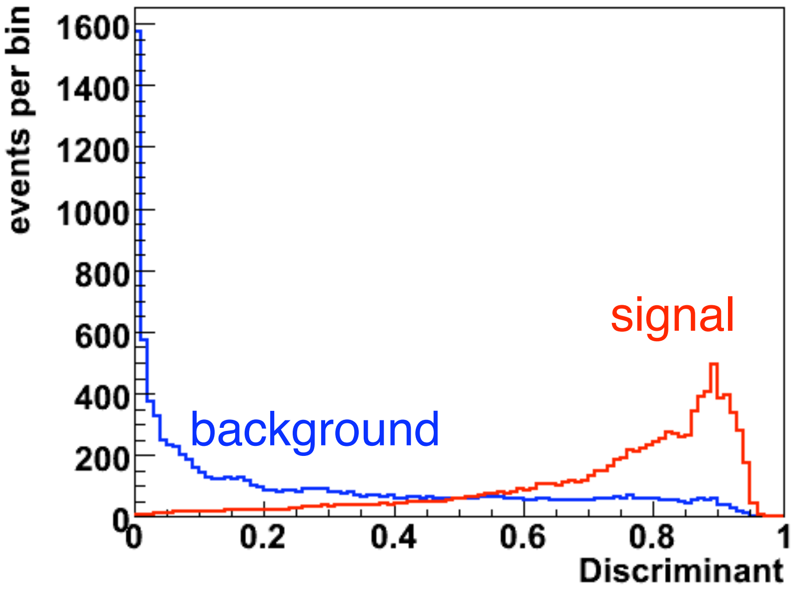
binary search tree:

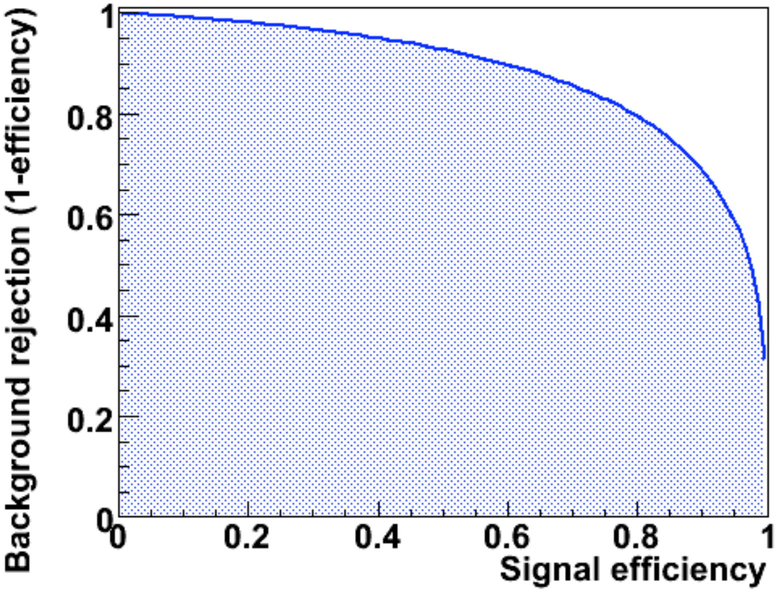


PDE: Application and Performance

- Use MC simulated signal and background events to densely populate the d-dimensional phase space (training samples)

- Use independent MC samples to evaluate the performance (testing samples)





- Most signal events are found at large values of D, most background events at low values of D
- Cut on D defines signal efficiency ε_{sig} and background rejection (=1-ε_{bg})

- *Receiver Operating Characteristic* (ROC) diagram: background rejection vs. signal efficiency
- Area under ROC curve is a measure of estimator performance

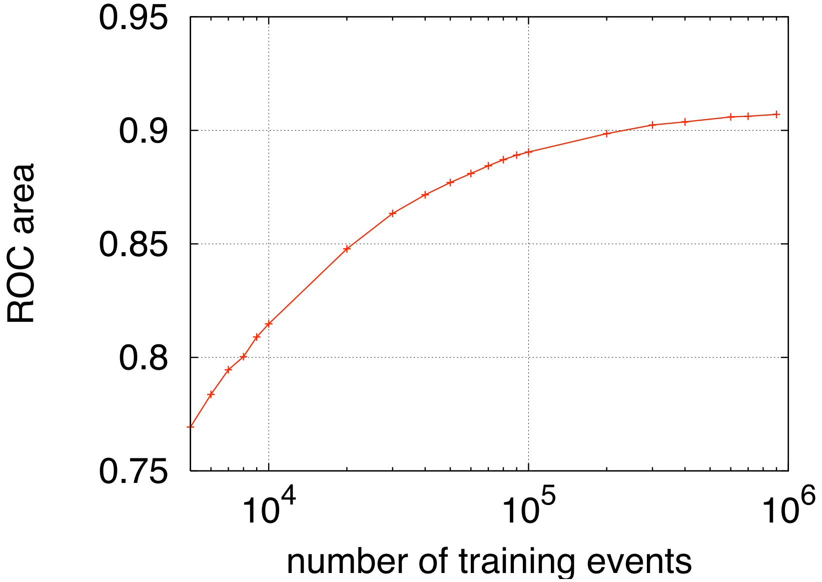
PDE-RS Features

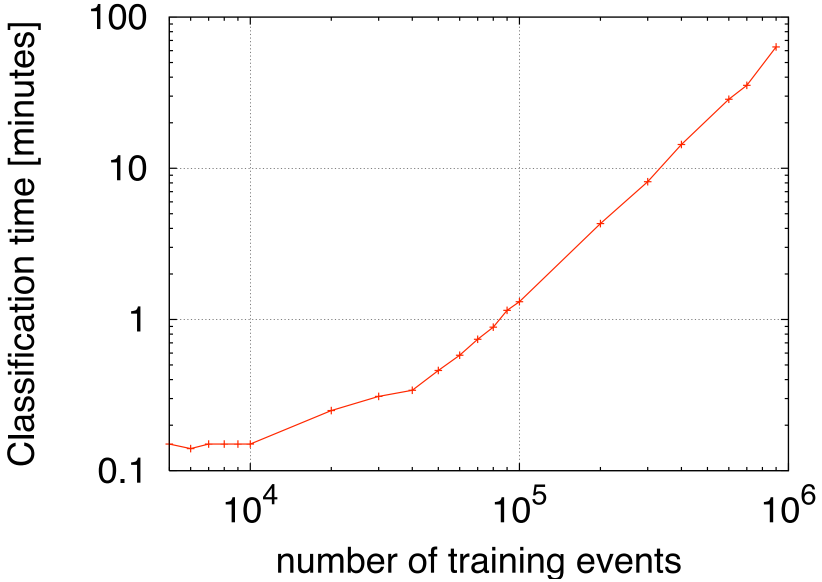
Advantages:

- Very good performance (close to optimal) for sufficient training sample sizes
- Correlations between observables automatically taken into account
- Only one parameter to optimise: box size
- Transparent handling of uncertainties, no hidden parameters
- Short training time (buildup of binary search trees)

Limitations:

- Large training samples required, have to be stored in main memory
- Slowly responding classifier $T_{classif} \sim N_{train} \cdot \log(N_{train})$



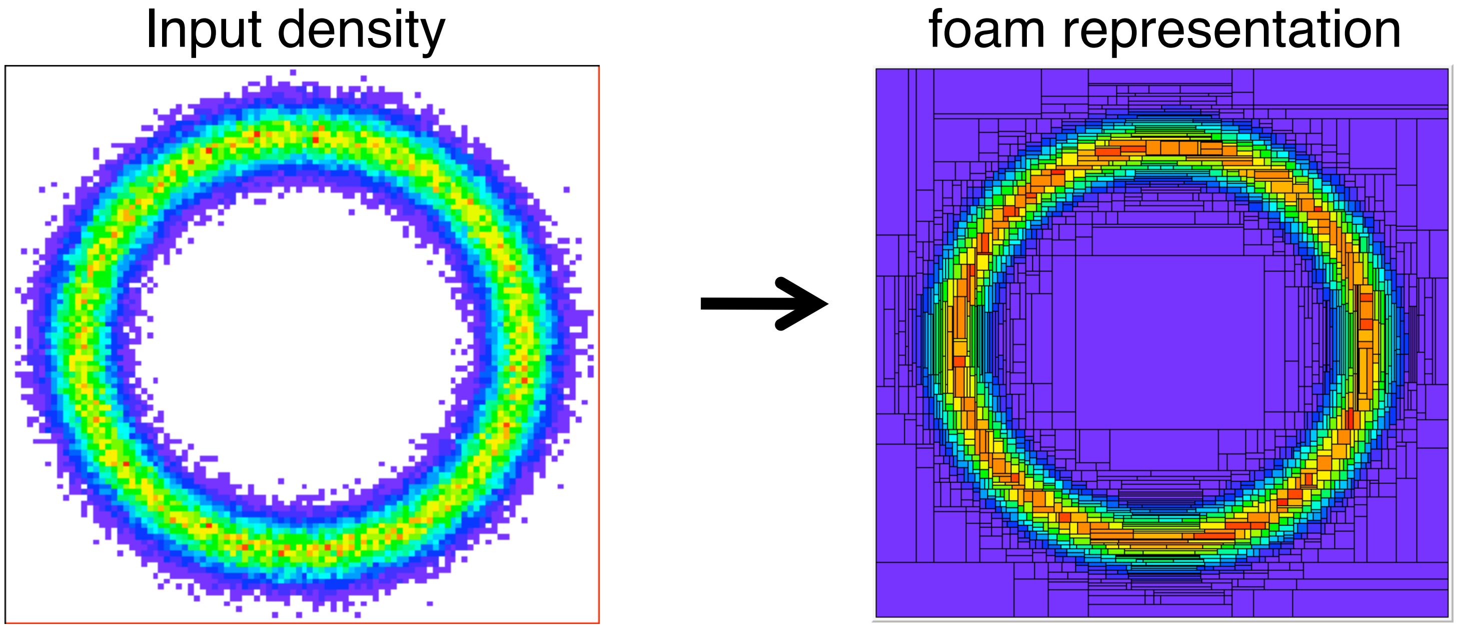


PDE-Foam

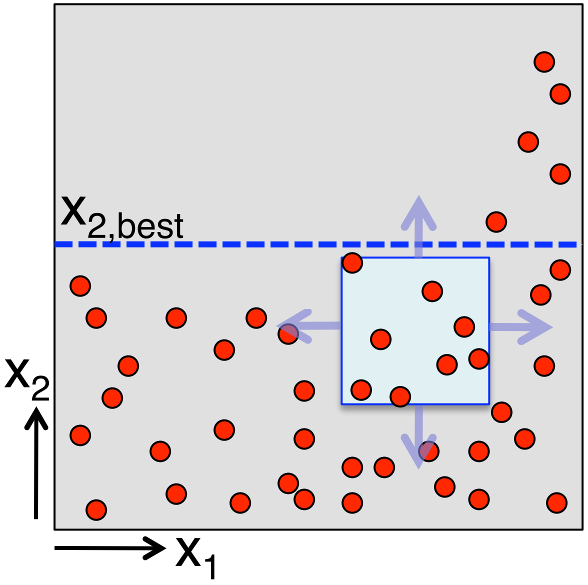
- Self-adapting binning method to divide d-dimensional phase space in finite number of hyper rectangles (cells)
- Algorithm based on Monte-Carlo Generator Package “Foam” by Stanislaw Jadach (*Comput.Phys.Commun. 152 (2003) 55-100*)
- Foam of cells: Few large cells in phase-space regions with constant likelihood density, many small cells in regions with high gradients of likelihood density
- Preserve only binned averaged density information after training phase

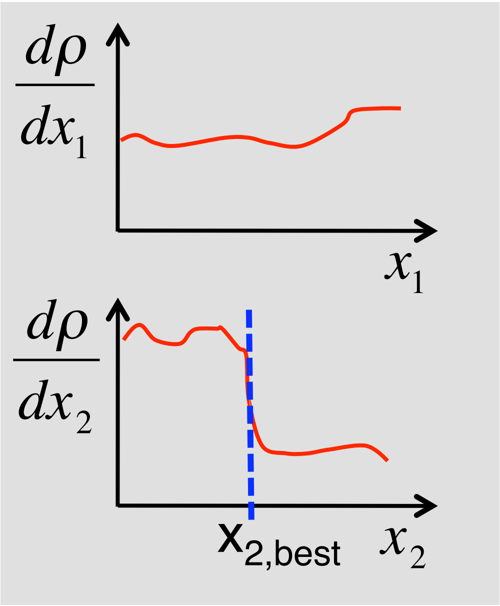
➔Fast and memory efficient classification, independent of training sample size

➔Reduces sensitivity to statistical fluctuations for small training samples



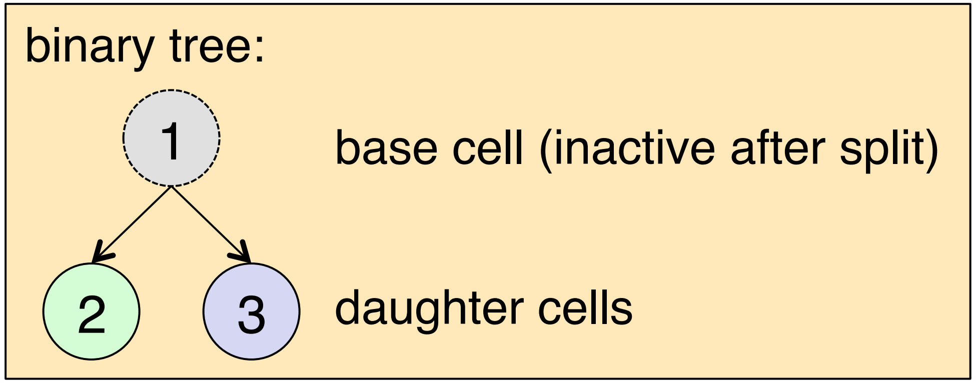
Foam Buildup (I)





Simplified explanation of the Foam buildup:

- Iterative cell-split algorithm, starting from base cell containing all events
- Density of distribution is sampled by counting events in a box of fixed size which is moved randomly across the cell
- All possible split points for dividing the cell across the axes are considered
- Variance of sampled density distributions are calculated for all possible new configurations
- Cell-split plane $x_{i,best}$ with highest gain in variance reduction is selected

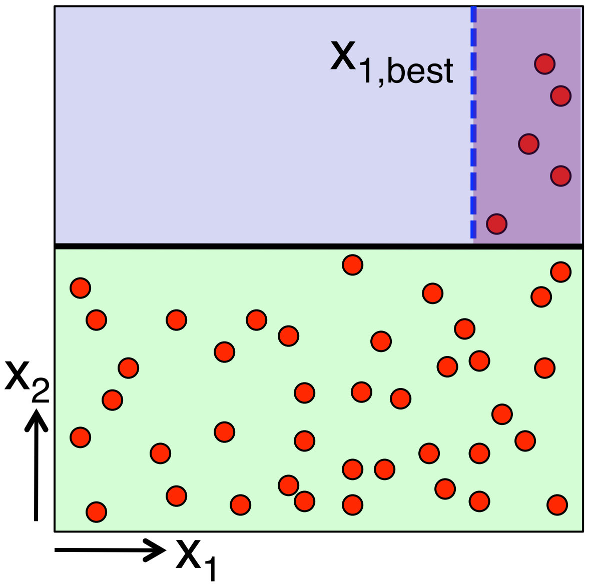


binary tree:

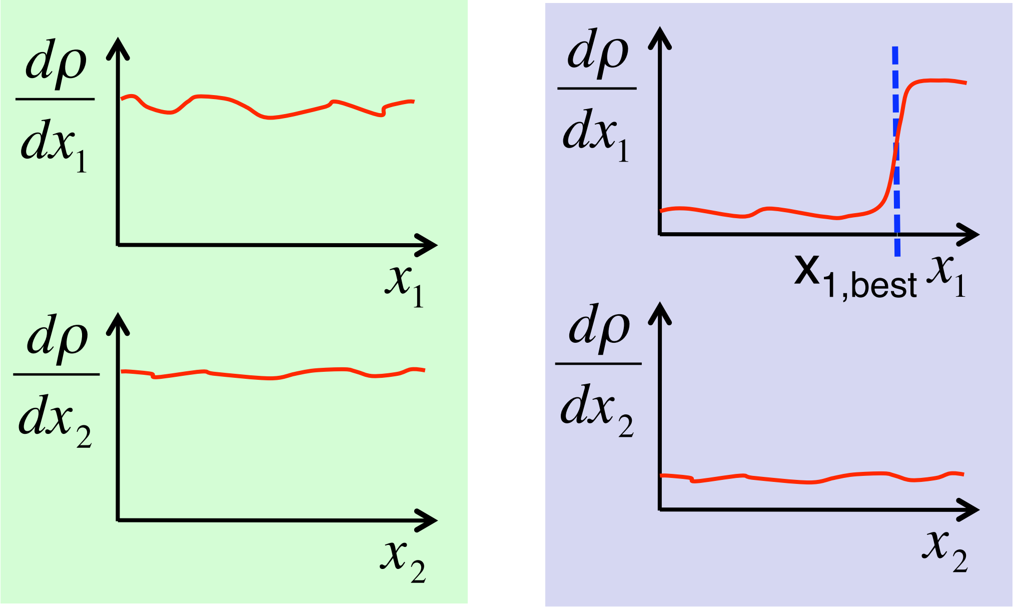
base cell (inactive after split)

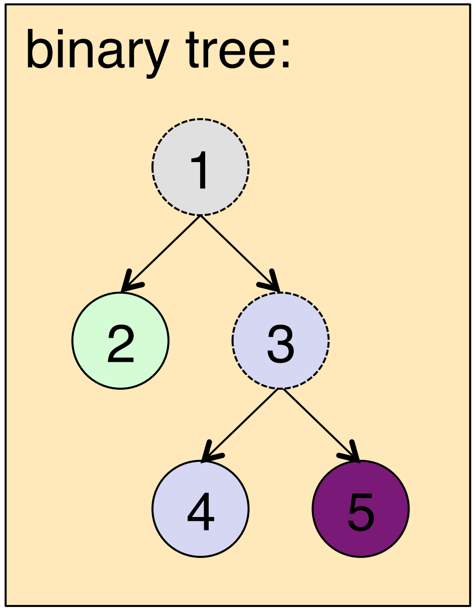
daughter cells

Foam Buildup (II)



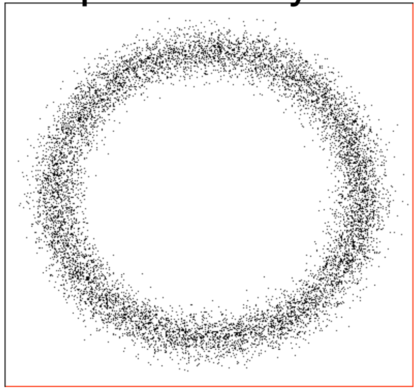
- Sampling of densities is repeated within resulting daughter cells
- next iteration: consider again all possible cell-split planes and chose the one with highest gain in variance
- Repeat cell split until predefined maximum number of cells (or other exit condition) is reached



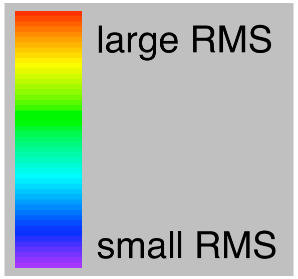


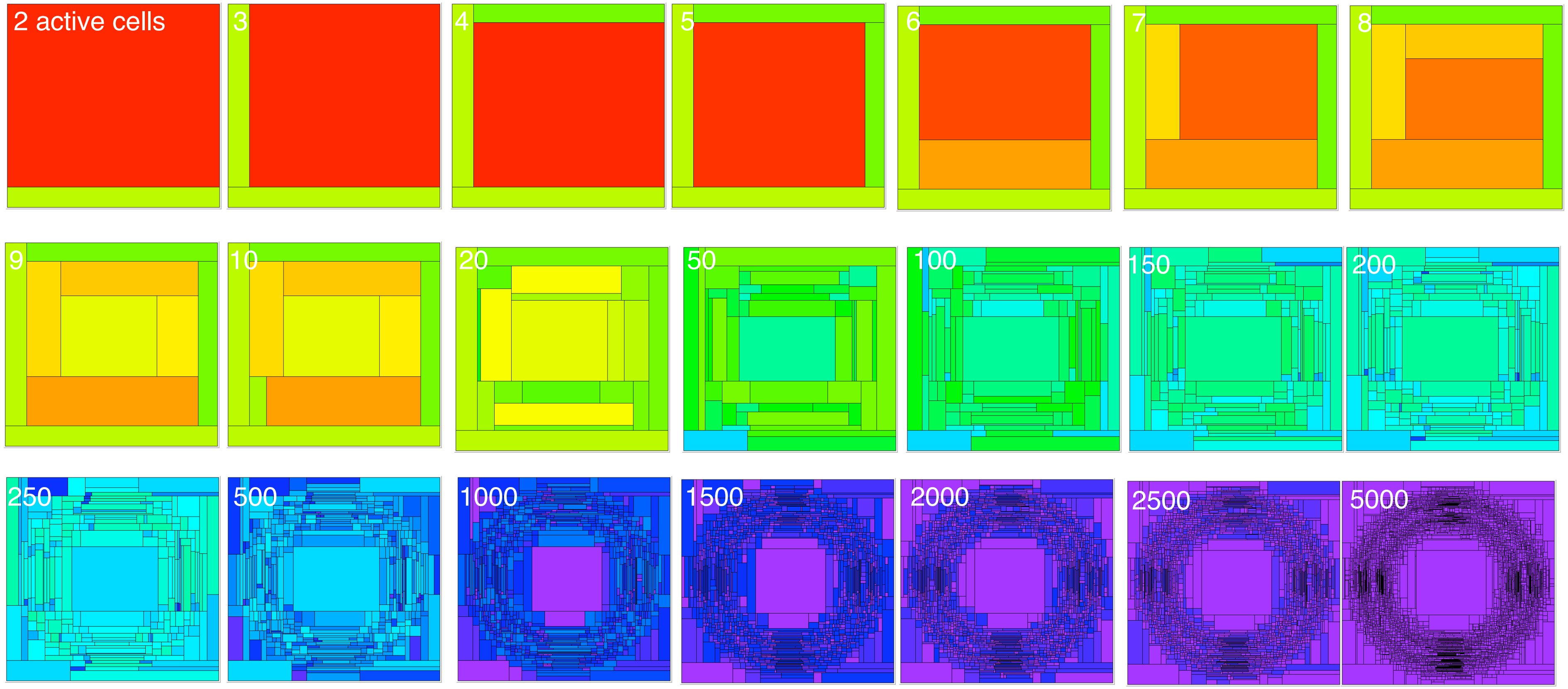
Foam Buildup Example

Input density



- Gauss distribution in radial dimension, uniform distribution in azimuth angle (Gaussian ring)
- Foam buildup based on 500k training events
- Size of sampling box: 0.5% of width of base cell
- 2000 samplings per cell
- Color scale: RMS of event density (log scale)





Foam Application for MVA (I)

1) Separate signal and background foams:

Training phase:

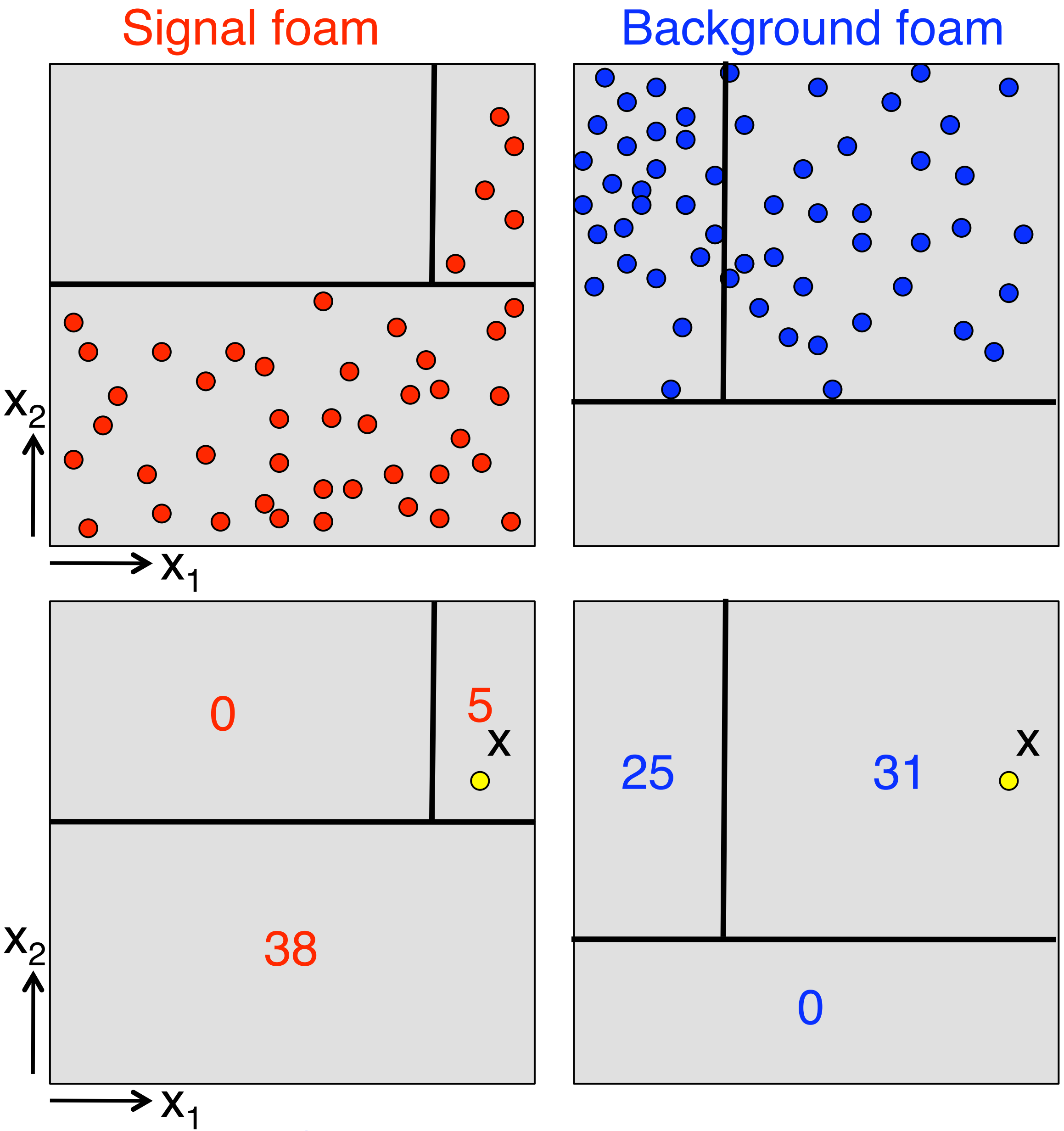
- One foam for signal, one for BG, based on event density
- Store number of events contained in each cell

Testing phase:

- For each event x to be classified: find corresponding cells i, j in signal and background foam containing x
- Calculate discriminant D(x):

$$D(x) = \frac{n_i / V_i}{n_i / V_i + c \cdot n_j / V_j}$$

- Also get statistical error $\sigma_D(x)$



Foam Application for MVA (II)

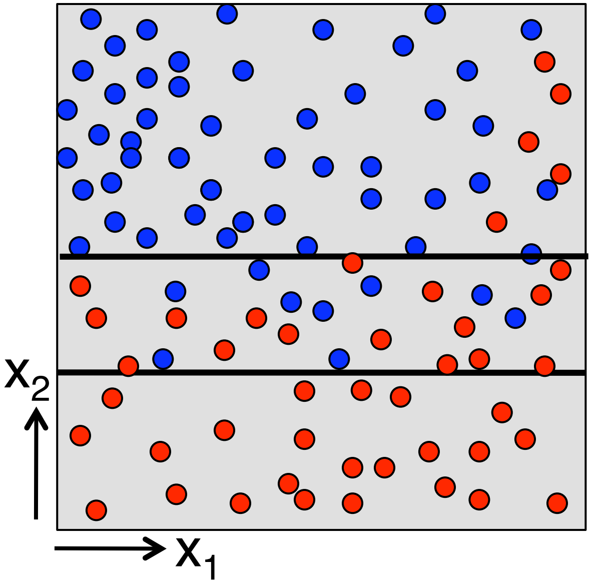
2) One foam for Discriminant distribution:

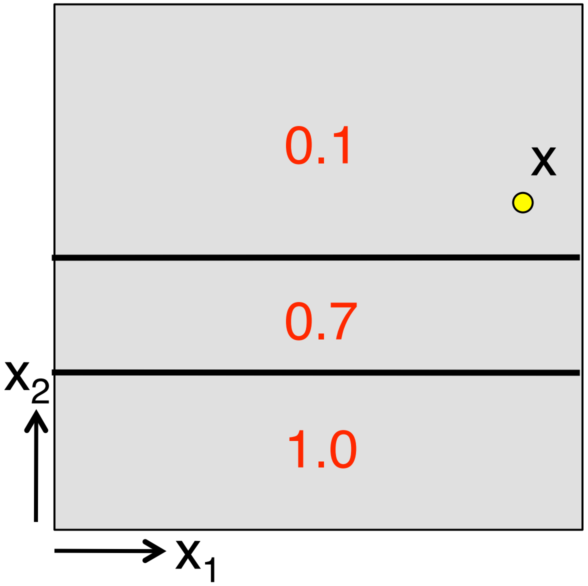
- Training phase:
  - Create one foam based on sampled discriminant calculated according to PDE-RS approach
  - Store discriminant D_i

$$D_i = \frac{n_{sig,i}}{n_{sig,i} + c \cdot n_{bg,i}}$$

  - Store also statistical error σ_i for each cell
- Testing phase:
  - For each event x to be classified: find corresponding cell i containing x and retrieve D_i and $\sigma_{D,i}$

Discriminant foam



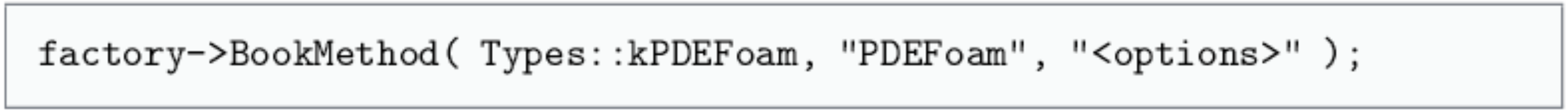


- Similar performance for one- and two-foams
- One-foam option is default

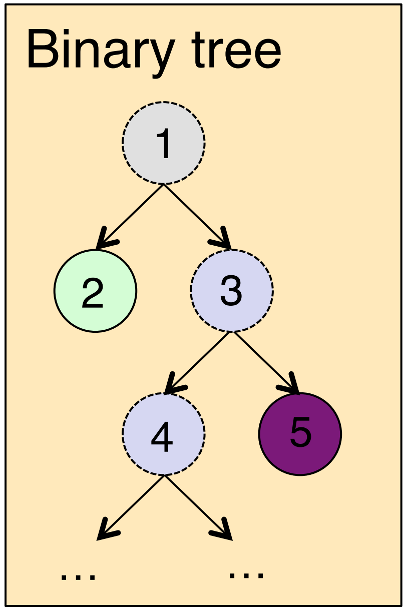
PDEFoam Implementation in TMVA

- Foam Algorithm implemented within the TMVA Framework (*http://tmva.sourceforge.net/*) as Method PDEFoam
- Core Foam functionality inherited from TFoam Class included in ROOT
- Parameter steering, classification output and persistency mechanism follows TMVA standards

→Allows to use it like other Methods and compare performance



```
factory->BookMethod( Types::kPDEFoam, "PDEFoam", "<options>" );
```

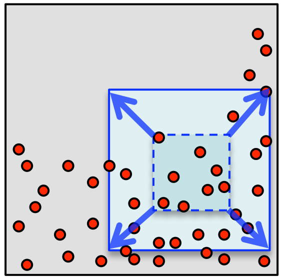


PDEFoam Object		1.4 kB
C 1	inactive	160 bytes
C 2	active	160 bytes
C 3	inactive	160 bytes
C 4	active	160 bytes
C 5	active	160 bytes
…	…	…

- Memory consumption on a 64-bit architecture

- 2 short Integer numbers per cell to store foam geometry: split coordinate + split point

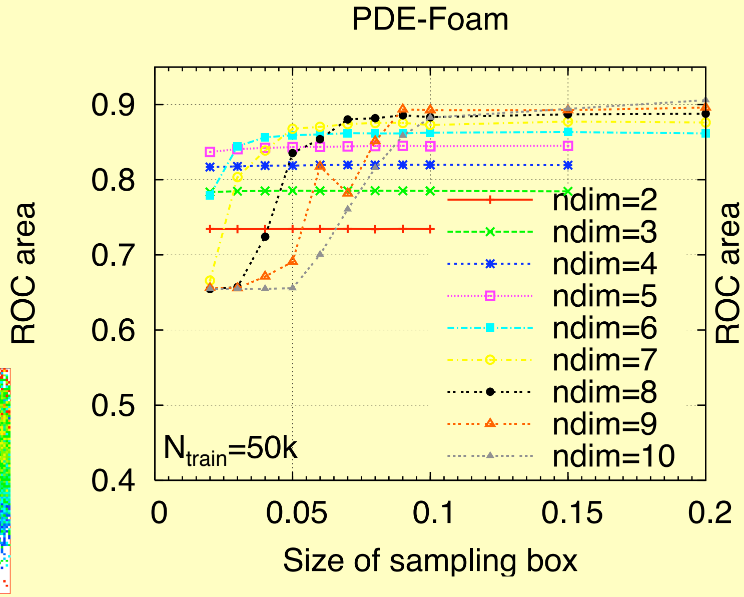
PDE Parameters: Size of Sampling Box

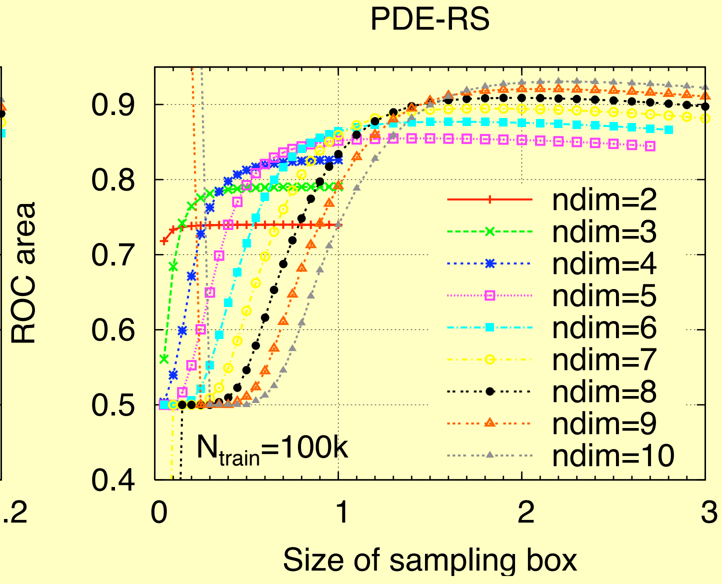


- Size of box used for sampling affects performance:
    - Larger box ➔ reduced statistical error, smoother sampling, slower sampling
    - Smaller box ➔ increased sensitivity to statistical fluctuations, more precise local estimate, faster sampling
- In general: higher dimensional problems require larger box

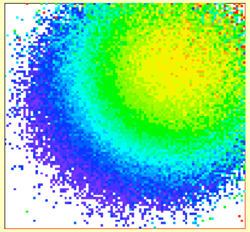
Toy-model “n-dim”

n-dim. Gauss distributions, shifted means for signal and BG





Input signal probability p_{sig} (ndim=2):



- Reduced sensitivity to size of sampling box for PDE-Foam in comparison with standard PDE-RS method

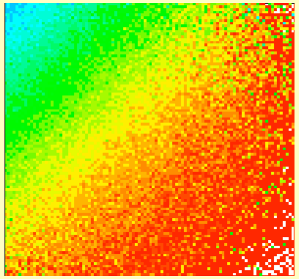
PDE-Foam Parameters: Number of Cells

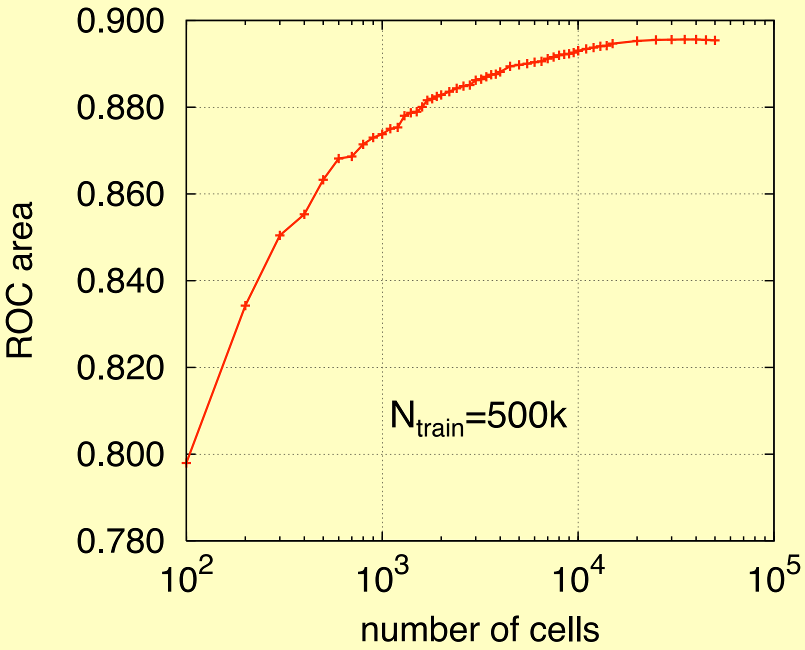
- Increasing the number of cells leads to:
    - improved performance for large training samples
    - danger of overtraining for small training samples (more vulnerable to statistical fluctuations in training samples)
    - more memory consumption
    - increased training time

Toy-model “5 observables”

5 moderately correlated observables constructed from Gaussian distributions for signal and background

Input signal probability p_{sig} (first two variables):



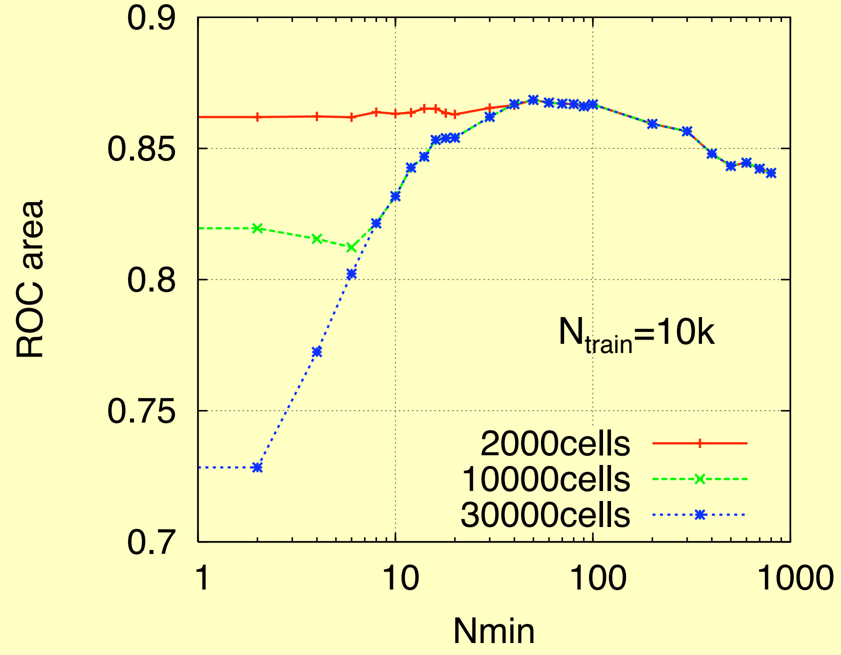


PDE-Foam Parameters: N_{min}

- Cut on the minimum number of events per cell:
  - Cell is not considered for splitting, if $N<N_{min}$ events are contained
  - Splitting iteration stops if no more cells with $N>N_{min}$

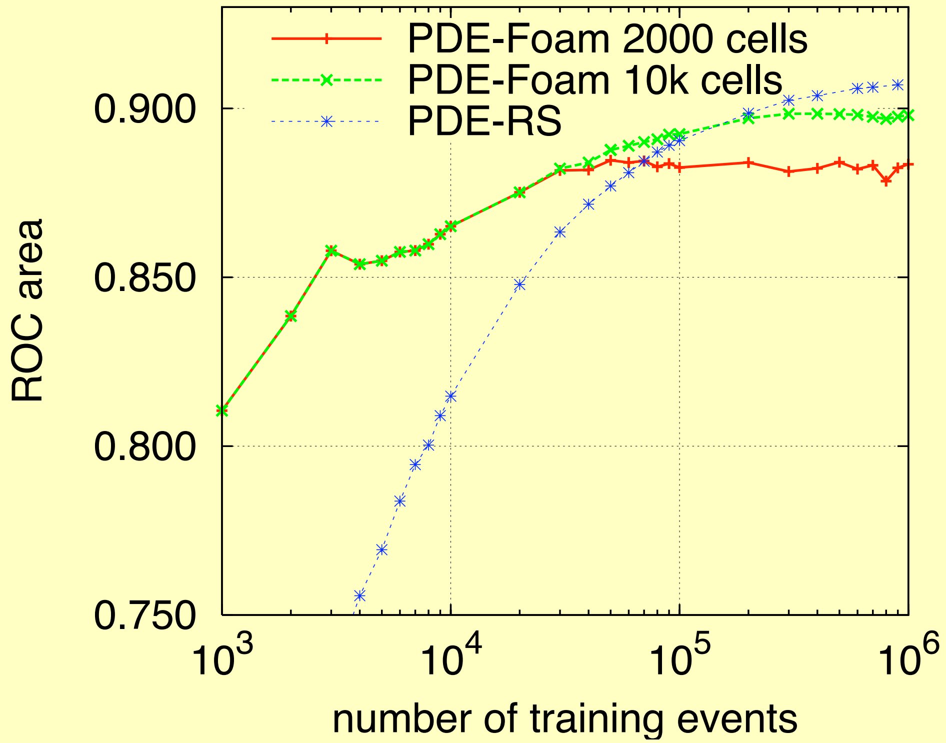
  → cut on N_{min} can limit the effective number of cells
- Reduces sensitivity to statistical fluctuation in training sample
- Improves performance drastically for small N_{train}
- N_{min}=100 (default) leads to good performance for most cases studied

Toy-model
“5 observables”



Foam Results: Performance Comparison

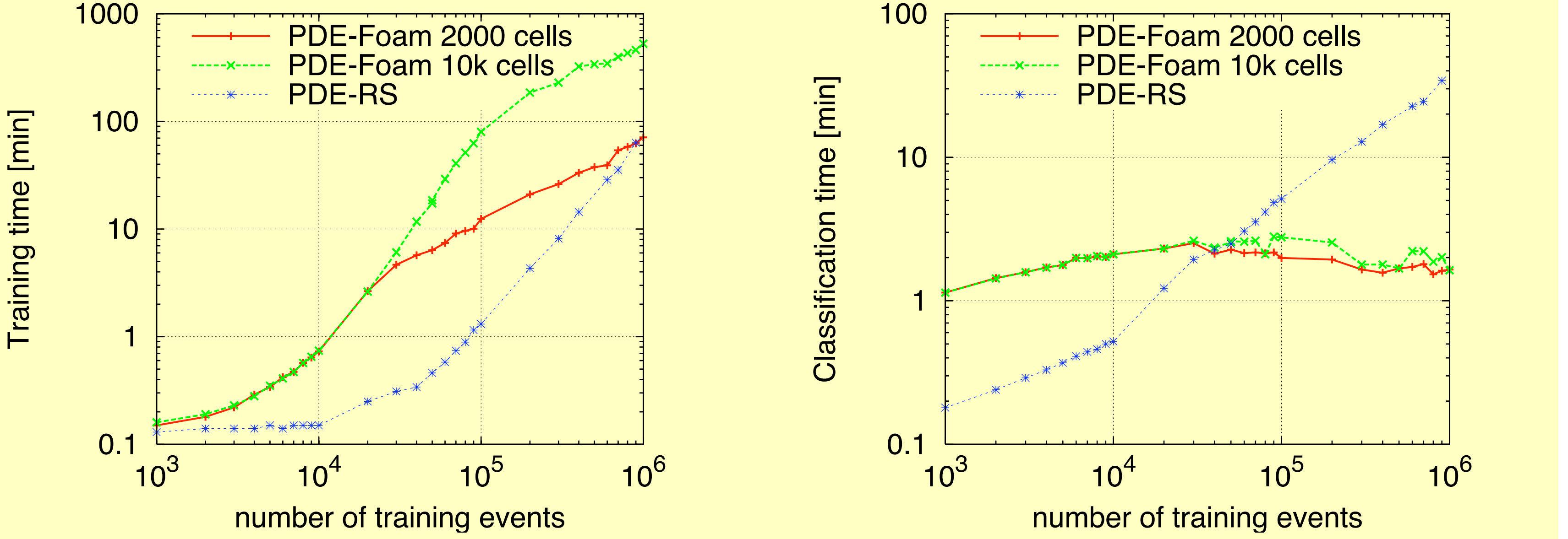
Toy-model
“5 observables”



- •Foam performance exceeds PDE-RS for small training samples
- •PDE-RS performance not reached for very large training samples
- •Foam performance increases with number of cells
(for large training samples)

Foam Results: CPU-Time Comparison

Toy-model “5 observables”

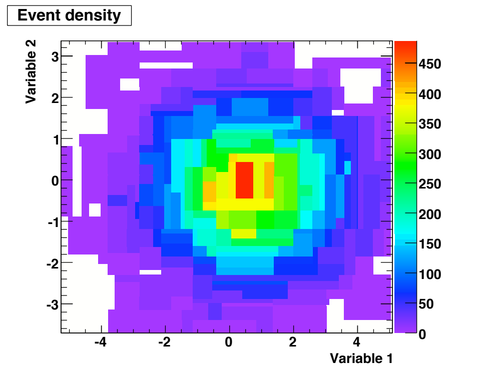


- Training time for Foam longer than for PDE-RS (due to time-consuming sampling and cell-splitting procedure)
- Training time increases with number of cells
- Classification time shorter than for PDE-RS
- Classification time independent of number of training events
- Classification time largely independent of number of cells

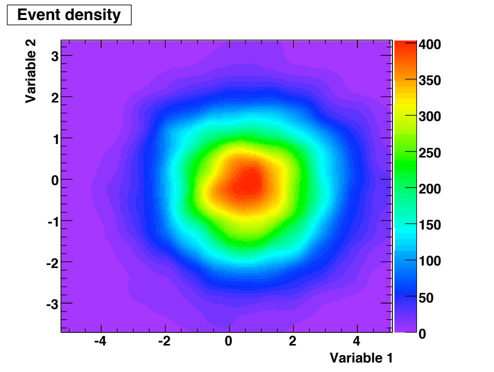
Foam Option: Gaussian Kernel Smoothing

- Gaussian Kernel smoothing can be applied during classification phase:
  - All cells contribute to discriminant calculation, convoluted with gauss-smeared euclidian distance to the event

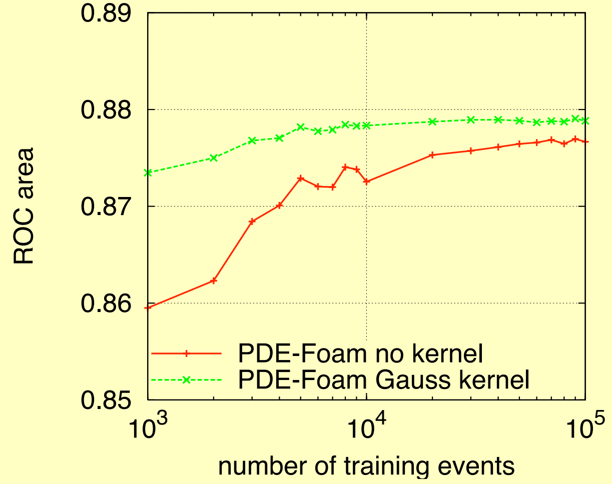
Example:
2-dimensional
Gauss distribution;
500 foam cells,
5000 training events



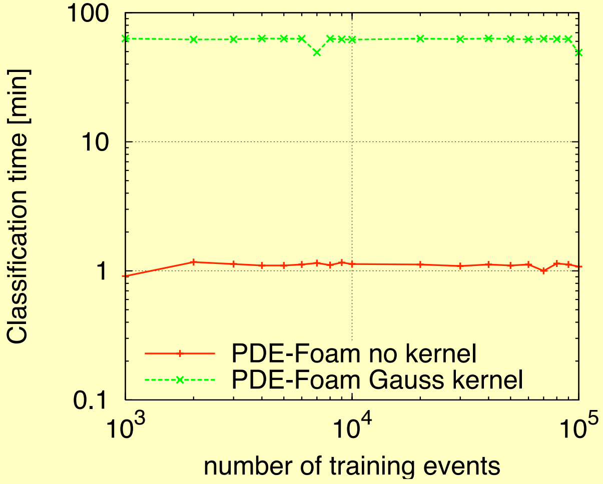
Foam



Density after smoothing



Improved performance if N_{train} and N_{cell} small



Increased classification time

Reconstruction of Event Quantities

- PDEFoam can also be used to reconstruct event quantities (regression)
- Event counting is replaced by averaging over event quantities inside the sampling box
- Density used for Foam buildup is given as: $\rho = \frac{\langle t \rangle}{V_s} \equiv \frac{\sum_{i=1}^{N_s} t^{(i)}}{N_s \cdot V_s}$

  N_s: events inside the sampling volume V_s
  $t^{(i)}$: target value of the ith event

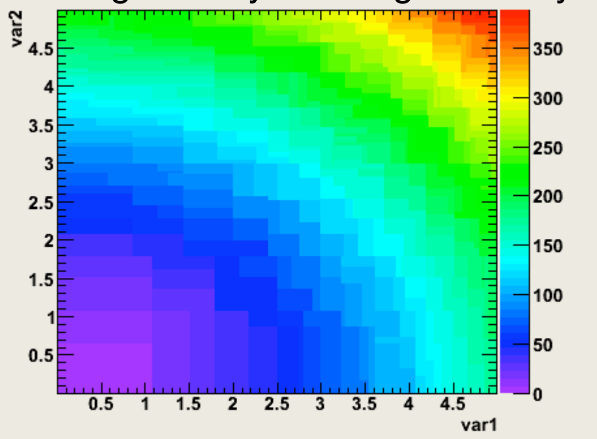
- Mean target density <t> is stored in each cell of the final foam
- Also multi-target regression is implemented: using higher Foam dimensions for the additional targets

Mono-target example:

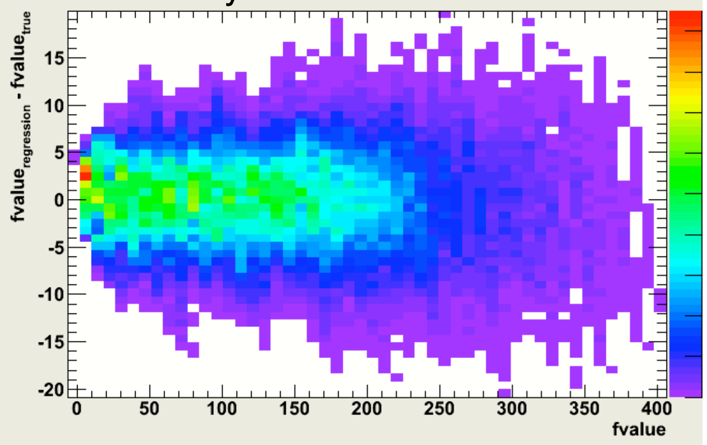
$t=a\bullet var1^2+b\bullet var2^2 +c+\Delta t$

Δt : small random number

Foam geometry and target density:



Accuracy of event reconstruction:



Conclusions

- Developed PDE-Foam Method for multivariate analysis
  - Combined TFoam adaptive binning method with PDE-RS probability-density estimation method
  - Implemented PDE-Foam in TMVA package
- Compared performance for various toy models and parameter settings
  - Performance exceeds PDE-RS for small training samples
  - Reduced classification time, independent of size of training sample
  - Reduced memory consumption, independent of size of training sample
- Adapted PDE-Foam for reconstruction of event variables