Self-Organizing Maps and Parton Distribution Functions

Katherine Holcomb
University of Virginia
Alliance for Computational
Science and Engineering
2010 May 18

Outline

- Previous work
- Motivation
- SOMs
- Preliminary Results
- Extension to GPDs
- Conclusions and Future Work

Previous and Current Work

- Earlier study by J. Carnahan, H. Honkanen, S. Liuti, Y. Loitiere, and P. Reynolds (D79, PRD, 2009)
  - The first version of the code did not readily extend to studying empirical PDF functions, but used data only
- Current work by D. Perry, K. Holcomb, S. Liuti, S. Taneja
  - Variations of functional forms of structure functions

Motivation

- Neural networks (strictly, artificial neutral networks or [A]NNs) have been widely applied to PDFs (http://sophia.ecm.ub.es/nnpdf/) to study the global properties of systems as they evolve from initial conditions.
- The network makes changes to its connections upon being informed of the "correct" result via a *cost function*. The aim is to minimize the cost.
- NNs are a *nonlinear statistical* data-modeling tool. Among other things they can be used to find patterns in data.

Learning in Neural Networks

- ANNs can learn via *supervised learning*, *unsupervised learning*, or *reinforcement learning.* Supervised learning is the "classical" ANN.
- Supervised learning: a set of examples is given; the goal is to force the data to match the examples as closely as possible. The cost function here implicitly includes knowledge about the domain.
- Unsupervised learning: data and the cost function are given and the network finds the minimum of the cost function without guidance.

Self-Organizing Maps (SOMs)

- A self-organizing map is a type of neural network with *unsupervised learning*. Unsupervised learning is related to the statistical problem of density estimation, i.e. the study of an otherwise unobservable probability density distribution function.
- The SOM reflects an organization in which neighboring locations of the map correspond to similarity in properties of the data.
- Invented by Teuvo Kohonen and sometimes called *Kohonen maps*.

Training and Mapping

- Training: the map is divided into cells. Each cell is populated randomly with a weight vector. The weight vectors are presented with example vectors. The cost function is minimized and the weight vectors are adjusted to be closer to the exemplars.
- Mapping: when presented with new data the algorithm finds the best-matching weight vector for each item and assigns that item to the corresponding cell.

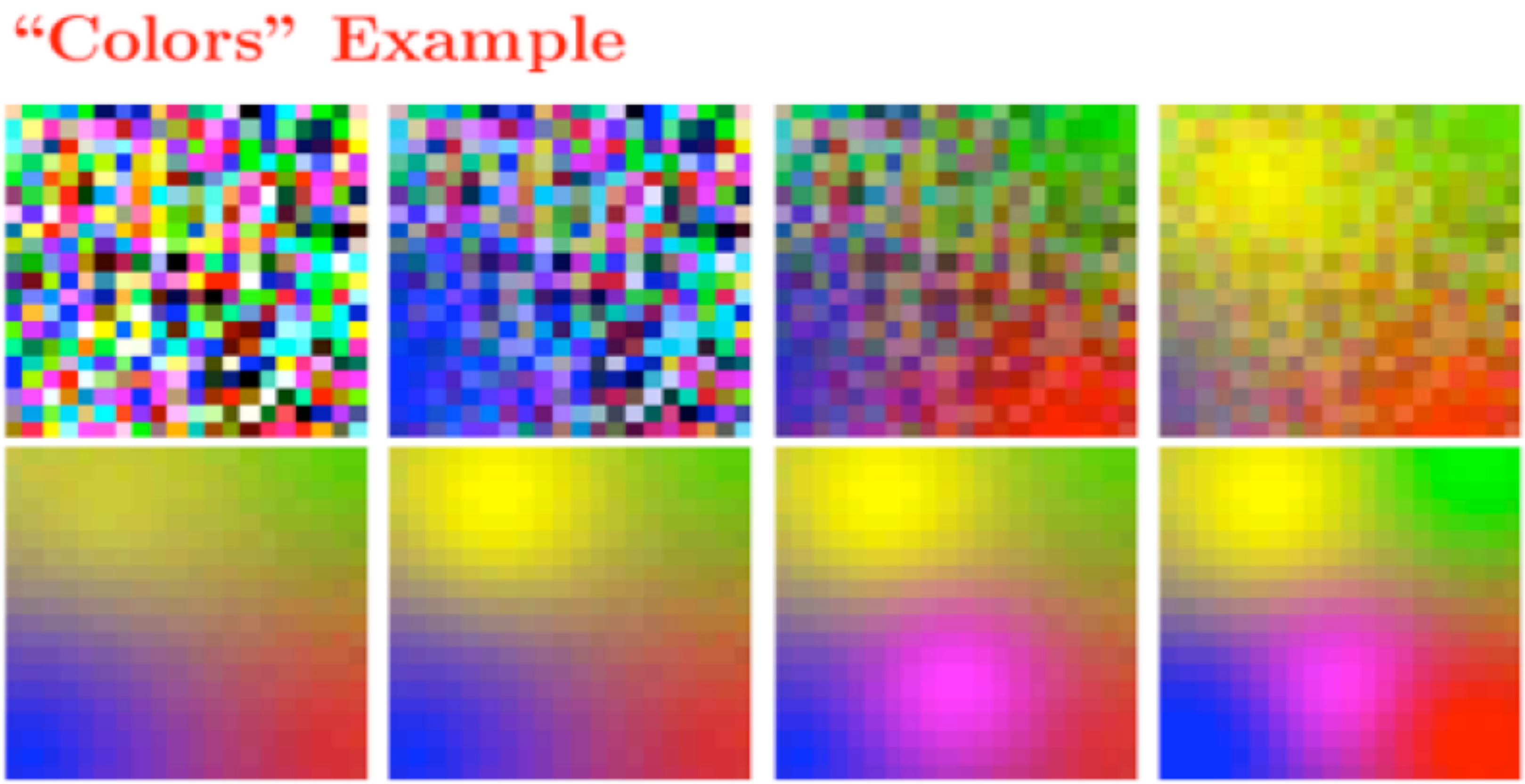
Learning in SOMs

- Competitive learning: for a given example data vector, the cost function is computed for all weight vectors. The most similar weight vector is the *best matching unit* (BMU) and it is adjusted to be closer to the exemplar, but so are its immediate neighbors. The adjustment for node *v* is determined by

$$\vec{W}_v(t+1) = \vec{W}_v(t) + \Theta(v,t)\alpha(t)\left(\vec{D}(t) - \vec{W}_v(t)\right)$$

- The quantity Θ(v,t) is a function of the distance between the BMU and node v and becomes smaller with each iteration ("time"). The quantity α is the learning coefficient and it decreases monotonically.

Example: Organizing Colors (Blue, Red, Yellow, Green, Magenta)



Example Application
Galaxy Classification

- Some work by A. Miller and M. Coe (*MNRAS* 1996)
- Automated plate scanner identified images of objects in the Coma cluster (over 1,000 identified galaxies, 99 Mpc distant)
- SOM trained with images from one region of the cluster, applied to other regions to classify the new images
- SOM required only a few hundred examples to be trained (other classification methods required thousands)

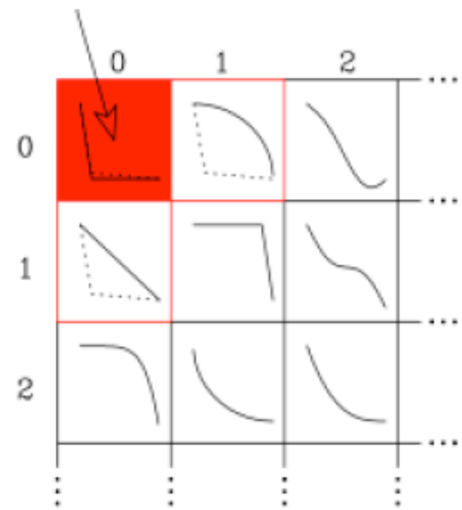
HST Image of Part of the Coma Cluster

SOMPDF

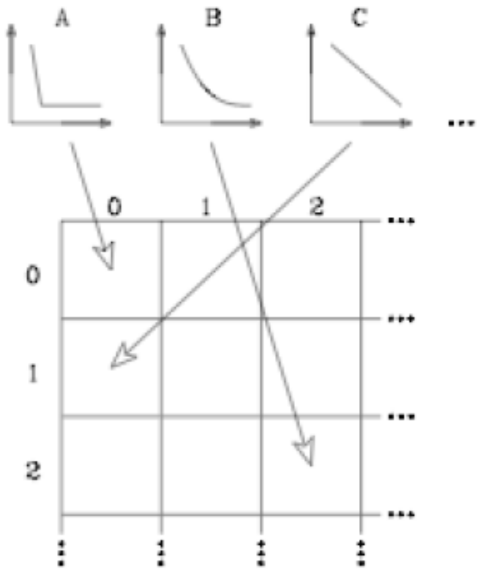
- We have adopted our "neighborhood function" Θ to be

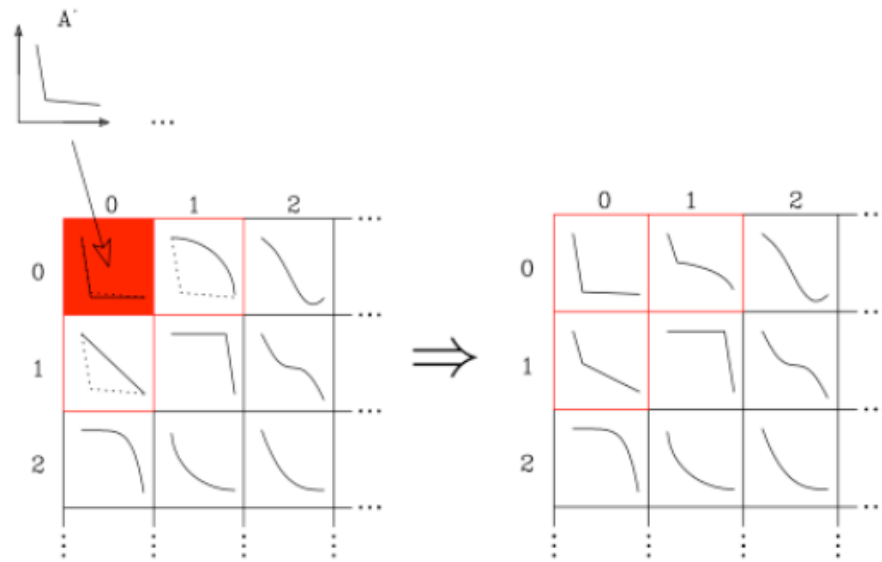
$$R_h(n) = 1.5\left(\frac{n_{train} - n}{n_{train}}\right)R_m$$

R_m is the size of the map, n Is the iteration, n_{train} is the total number of iterations. The initial layout may be represented by the figure to the right.



SOM Algorithm Illustrated





<u>Initialization</u>: functions are placed on map

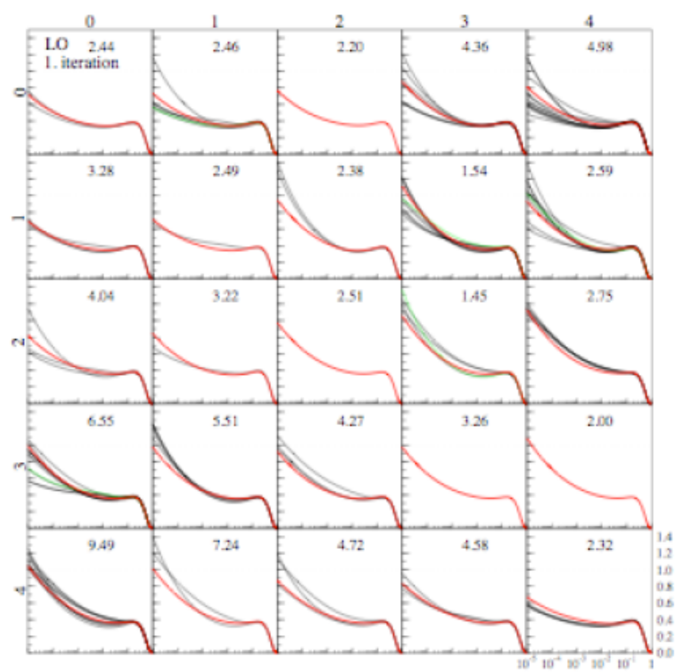
<u>Training</u>: "winner" node is selected,
<u>Learning</u>: adjacent nodes readjust according to similarity criterion

Final Step : clusters of similar functions distribute themselves on the map.

SOMPDF Method

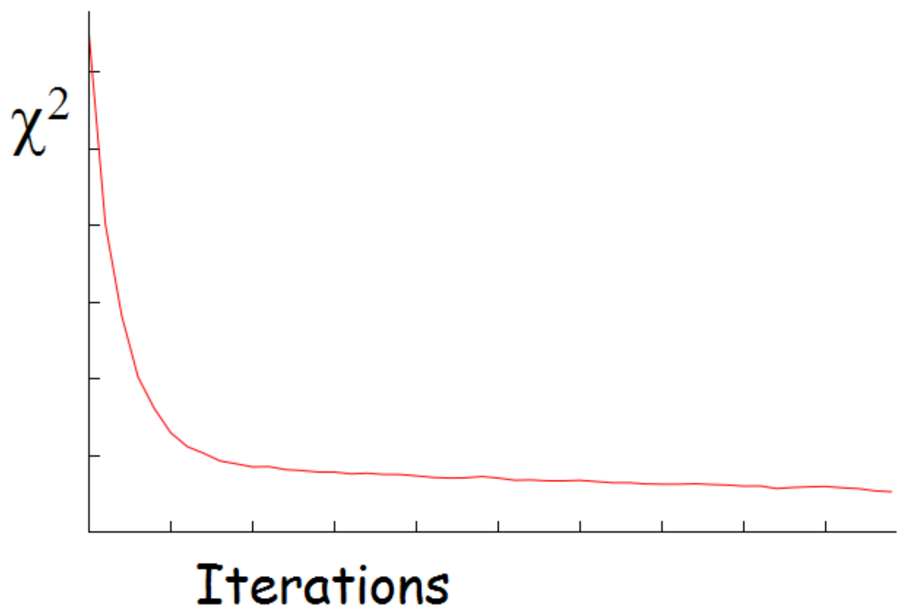
Initialization: a set of “map PDFs” (weight vectors) is formed by selecting at random from existing PDF functions and varying their parameters.
Baryon number and momentum sum rules are imposed at every step.
These input PDFs are used to initialize the map.

Training: Another set of exemplar PDFs is generated and is used to train the map. The similarity is tested by comparing the PDFs at given (x,Q^2) values. The new map PDFs are obtained by the SOM algorithm.



Minimization Through Genetic Algorithm

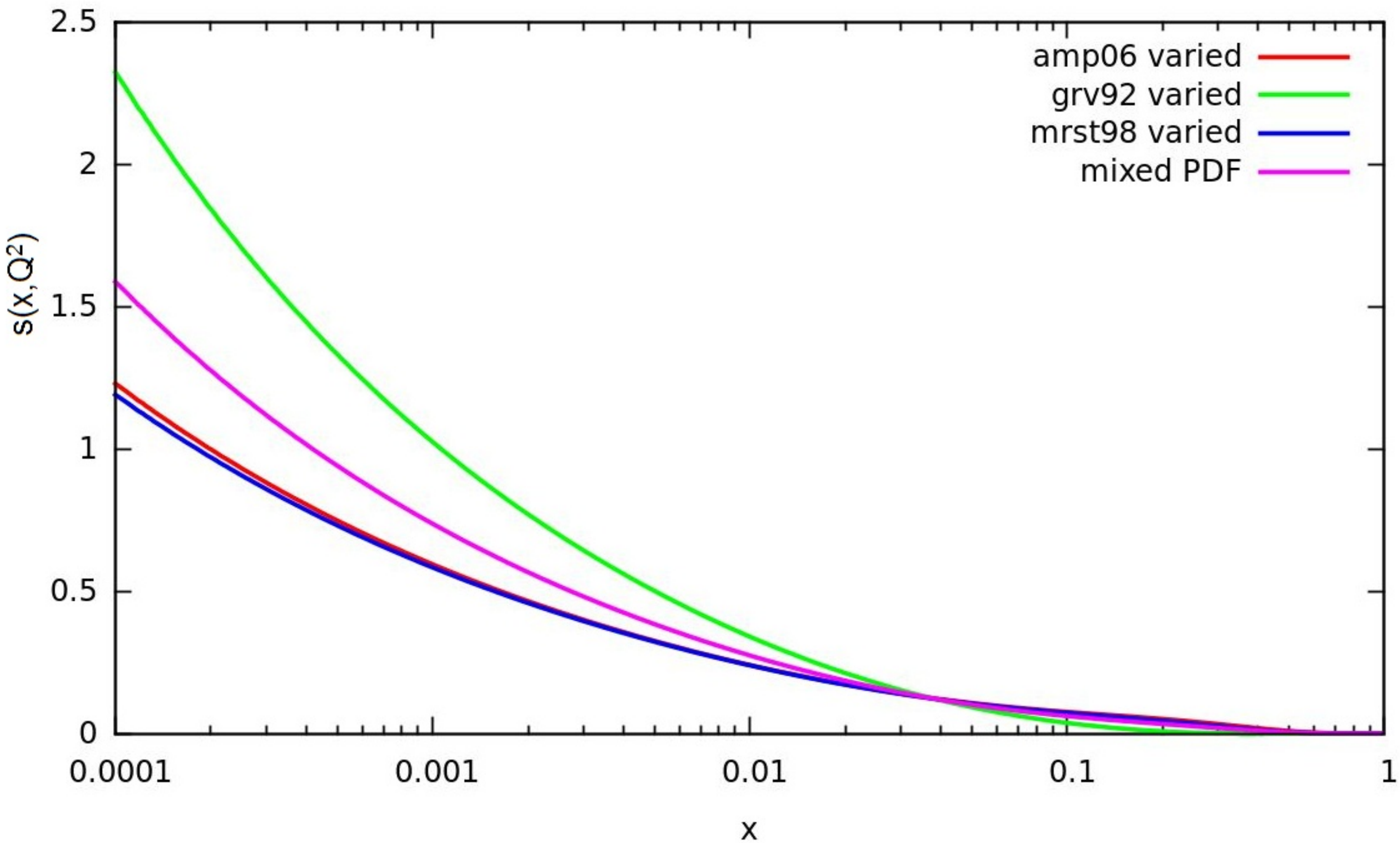
- Once the map is trained, we select a subset of the training PDFs with the best values of χ^2. These are used to generate a new set of training vectors. The cycle is repeated to convergence.



Mixing

- In generating the PDFs (for the map and for the training) we need to avoid introducing a functional bias
- Thus we mix together variations of different structure functions
  - Random perturbations are used to generate a variant of a standard set of structure functions—currently based on GRV, MRST, AMP. We select some number of these varied functions, then combine them in a weighted-average linear combination to obtain a final candidate PDF.
  - Sum rules are enforced on each candidate “mixed” PDF

Example Mixed PDF

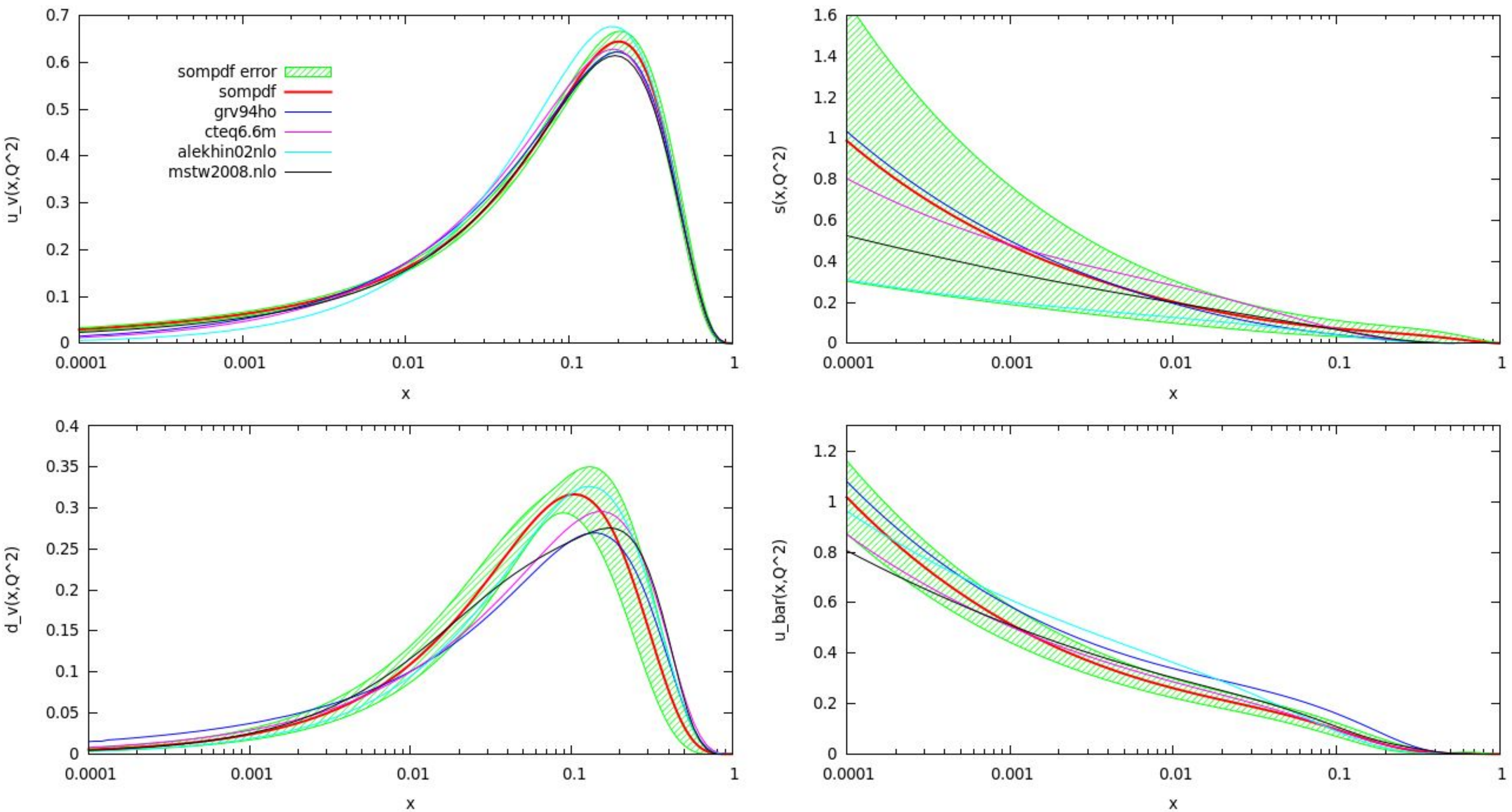


Error Analysis

- Treatment of experimental error is complicated by the incompatibility of different experiments
- Treatment of theoretical error is complicated because the theoretical errors and their correlations are poorly understood
- We have defined a statistical error on an ensemble of SOMPDF runs
- More detailed error analysis using a Lagrange multiplier technique is underway

Preliminary Results

- Up valence, strange, down valence, and up-bar quark distributions at Q^2=7.5 GeV^2. The size of the map was 5x5 with 43 runs. The error bands are the statistical error from these runs.



Extension to GPDs

- SOMs find similarities in the input data without a training target.
- They have been used in theoretical physics approaches to critical phenomena, to the study of complex networks, and in general for the study of high dimensional non-linear data (see e.g. Der, Hermann, Phys.Rev.E (1994), Guimera et al., Phys. Rev.E (2003))

- SOMs have been used in theoretical physics approaches to critical phenomena, to the study of complex networks, and in general for the study high dimensional **non linear** data

(see e.g. Der, Hermann, *Phys. Rev. E* (1994), Guimera et al., *Phys. Rev.E* (2003))

- *Study of particle shape and size*, N. Laitinen et al. 2001

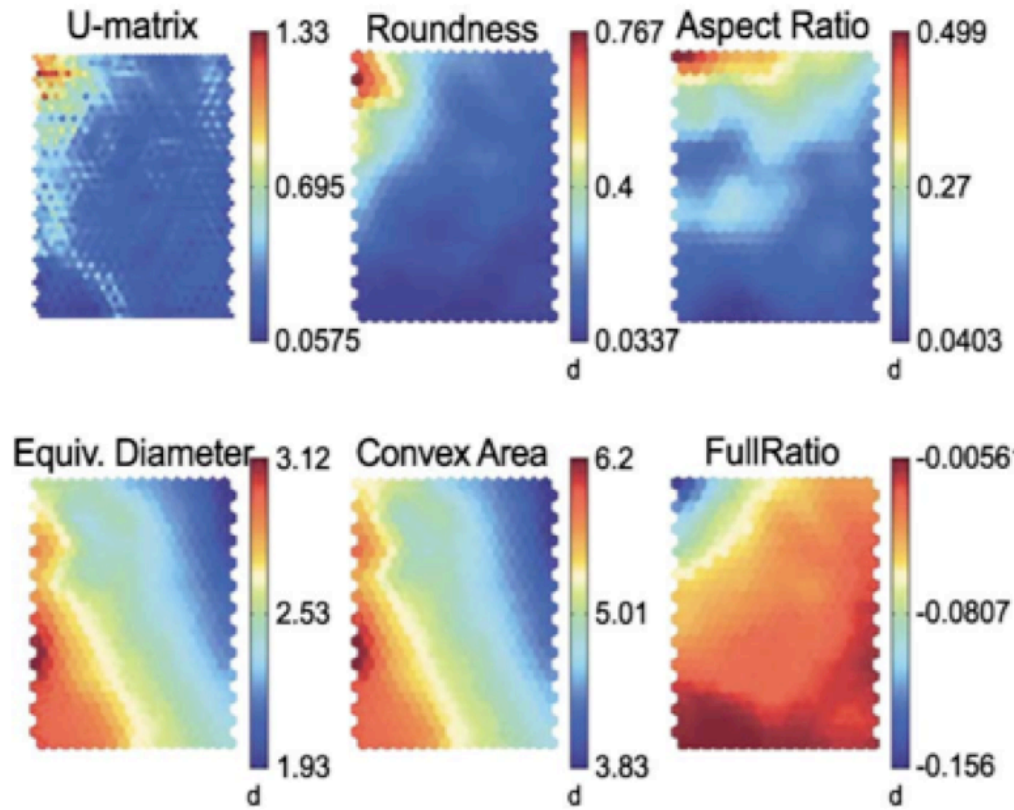


Fig. 8. The U-matrix and the variable information for the model particles.

We are studying similar characteristics of SOMs to devise a fitting procedure for GPDs: our new code has been made flexible for this use

Main question: Which experiments, observables, and with what precision are they relevant for which GPD components?

From Guidal and Moutarde, and Moutarde analyses (2009)

$$H_{Re} = P\int_0 dx\,[H(x,\xi,t) - H(-x,\xi,t)]\,C^{+}(x,\xi) \quad (1$$

$$E_{Re} = P\int_0^1 dx\,[E(x,\xi,t) - E(-x,\xi,t)]\,C^{+}(x,\xi), \quad (2$$

$$\tilde{H}_{Re} = P\int_0^1 dx\left[\tilde{H}(x,\xi,t) + \tilde{H}(-x,\xi,t)\right]C^{-}(x,\xi) \quad (3$$

$$\tilde{E}_{Re} = P\int_0^1 dx\left[\tilde{E}(x,\xi,t) + \tilde{E}(-x,\xi,t)\right]C^{-}(x,\xi) \quad (4$$

$$H_{Im} = H(\xi,\xi,t) - H(-\xi,\xi,t), \quad (5$$

$$E_{Im} = E(\xi,\xi,t) - E(-\xi,\xi,t), \quad (6,$$

$$\tilde{H}_{Im} = \tilde{H}(\xi,\xi,t) + \tilde{H}(-\xi,\xi,t) \quad \text{and} \quad (7)$$

$$\tilde{E}_{Im} = \tilde{E}(\xi,\xi,t) + \tilde{E}(-\xi,\xi,t) \quad (8)$$

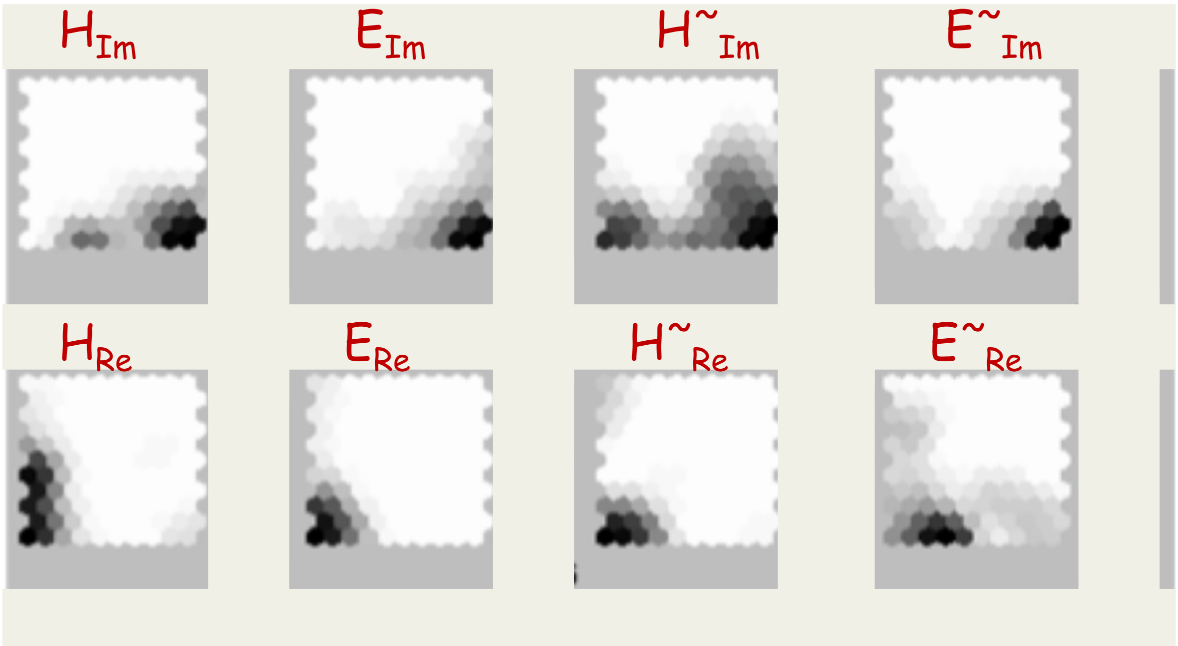
8 GPD-related functions

$$\begin{aligned}
&\boldsymbol{A_{\{C\}}},\ A_{\{C\}}^{\sin\phi},\ \boldsymbol{A_{\{C\}}^{\cos\phi}},\ A_{\{C\}}^{\cos 2\phi},\ A_{\{C\}}^{\cos 3\phi}\\
&A_{\{LU,DVCS\}},\ A_{\{LU,DVCS\}}^{\sin\phi},\ A_{\{LU,DVCS\}}^{\cos\phi},\ A_{\{LU,DVCS\}}^{\sin 2\phi}\\
&A_{\{LU,I\}},\ \boldsymbol{A_{\{LU,I\}}^{\sin\phi}},\ A_{\{LU,I\}}^{\cos\phi},\ A_{\{LU,I\}}^{\sin 2\phi}\\
&\boldsymbol{A_{\{Ux,I\}}^{\sin\phi}},\\
&A_{\{Uy,DVCS\}},\\
&\boldsymbol{A_{\{Uy,I\}}} \quad \text{and} \quad \boldsymbol{A_{\{Uy,I\}}^{\cos\phi}} \qquad (13)
\end{aligned}$$

17 observables (6 LO) from HERMES + Jlab data

“a challenge for phenomenology…” (Moutarde) + “theoretical bias”

The 8 GPDs are the dimensions in our analysis



Conclusions and Future Work

- We have developed a method to find in an unbiased manner PDFs with minimum χ^2 and have performed a preliminary analysis of the error.
- Next steps: larger maps (will require parallel runs), effects of varying parameters, applications to more varied datasets (polarized scattering)
- Next: onward to GPDs