Re-evaluation of the Bjorken sum rule with a MonteCarlo approach

L. Del Debbio[1], A. Guffanti[2], A. P.[3]

[1] Particle Physics Theory Group, School of Physics, University of Edinburgh
[2] Physikalisches Institut, Albert-Ludwigs-Universität, Freiburg
[3] Dipartimento di Fisica, Università di Milano and INFN, Sezione di Milano

from data to sum rules

the problem:

- given a finite set of data points for the asymmetry $A_1^{p,d}(x, Q^2)$;
- we want to evaluate an integral and its error

$$\int_0^1 dx \, [g_1^p(x, Q^2) - g_1^n(x, Q^2)]$$

where

$$g_1(x, Q^2) = A_1(x, Q^2) F_1(x, Q^2) + \gamma^2 g_2(x, Q^2) \, ;$$

- to determine an infinite-dimensional object (a function) from finite set of data points is a mathematically ill-posed problem.

ways out

reduce to a finite dimensional problem:

- take a general enough parametrization;
- determine the parameters by comparing to data and minimizing a given figure of merit.

open issues:

- how do we propagate errors from data to the parametrization?
- how do we propagate errors from the parametrization to an integral?
- how does the parametrization affects the minimization?
- how do we access the error associated to the parametrization?

ways out

reduce to a finite dimensional problem:

- take a general enough parametrization;
- determine the parameters by comparing to data and minimizing a given figure of merit.

open issues:

- how do we propagate errors from data to the parametrization?
- how do we propagate errors from the parametrization to an integral?
- how does the parametrization affects the minimization?
- how do we access the error associated to the parametrization?

our recipe

- use a montecarlo sample of data;
- use a redundant parametrization;
- use an efficient minimization.

references:

- $F_2^p(x,Q^2), F_2^d(x,Q^2), F_2^{NS}(x,Q^2)$ [Forte, Latorre, Garrido, A.P. - 2002]
- $F_2^p(x,Q^2)$ [Del Debbio, Forte, Latorre, A.P., Rojo - 2005]
- non-singlet PDF [Del Debbio, Forte, Latorre, A.P., Rojo - 2007]
- PDF global set [see Rojo's talk in SF session]

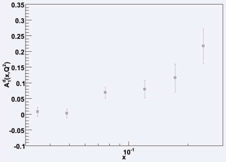
our recipe

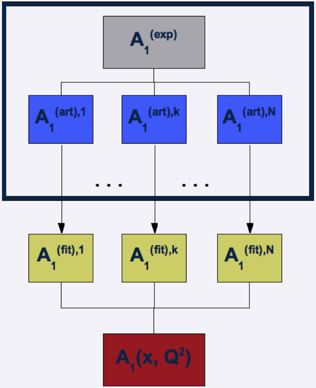
- use a montecarlo sample of data;
- use a redundant parametrization;
- use an efficient minimization.

references:

- $F_2^p(x, Q^2), F_2^d(x, Q^2), F_2^{NS}(x, Q^2)$ [Forte, Latorre, Garrido, A.P. - 2002]
- $F_2^p(x, Q^2)$ [Del Debbio, Forte, Latorre, A.P., Rojo - 2005]
- non-singlet PDF [Del Debbio, Forte, Latorre, A.P., Rojo - 2007]
- PDF global set [see Rojo's talk in SF session]

the montecarlo sample





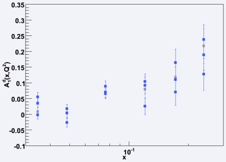
from data to the parametrization:

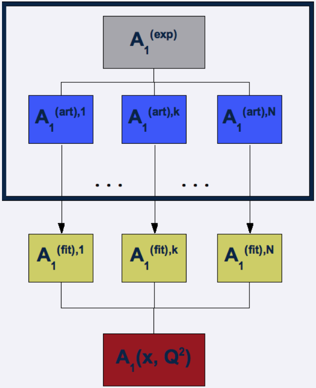
$$A_1^{(art),k}(x,Q^2) = (1 + r_{k,N}\sigma_N)\left(A_1^{(exp)}(x,Q^2) + r_{k,t}\sigma_t(x,Q^2)\right)$$

from the parametrization to an integral:

$$\int_0^1 [g_1^p(x,Q^2) - g_1^n(x,Q^2)] \simeq \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} \int_0^1 [g_1^{p,k}(x,Q^2) - g_1^{n,k}(x,Q^2)]$$

the montecarlo sample





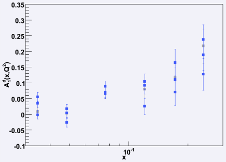
from data to the parametrization:

$$A_1^{(art),k}(x,Q^2) = (1 + r_{k,N}\sigma_N)\left(A_1^{(exp)}(x,Q^2) + r_{k,t}\sigma_t(x,Q^2)\right)$$

from the parametrization to an integral:

$$\int_0^1 [g_1^p(x,Q^2) - g_1^n(x,Q^2)] \simeq \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} \int_0^1 [g_1^{p,k}(x,Q^2) - g_1^{n,k}(x,Q^2)]$$

the montecarlo sample

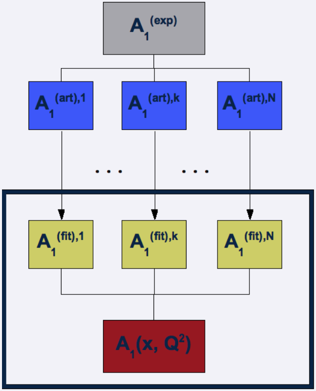


from data to the parametrization:

$$A_1^{(art),k}(x,Q^2) = (1 + r_{k,N}\sigma_N)\left(A_1^{(exp)}(x,Q^2) + r_{k,t}\sigma_t(x,Q^2)\right)$$

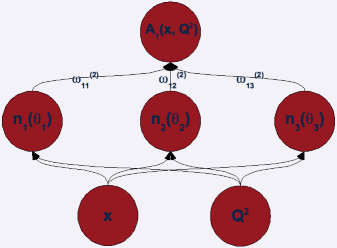
from the parametrization to an integral:

$$\int_0^1 [g_1^p(x,Q^2) - g_1^n(x,Q^2)] \simeq \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} \int_0^1 [g_1^{p,k}(x,Q^2) - g_1^{n,k}(x,Q^2)]$$



the parametrizations

we use neural networks since they are redundant and smooth.

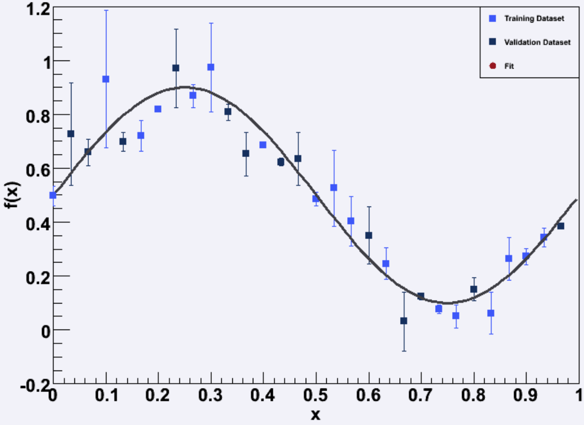


as an example, if $A_1(x, Q^2) \sim A_1(x)$, in a simple case (1-2-1) we have

$$A_1(x) = \frac{1}{1 + e^{\theta_1^{(3)} - \frac{\omega_{11}^{(2)}}{1+e^{\theta_1^{(2)} - x\omega_{11}^{(1)}}} - \frac{\omega_{12}^{(2)}}{1+e^{\theta_2^{(2)} - x\omega_{21}^{(1)}}}}}$$

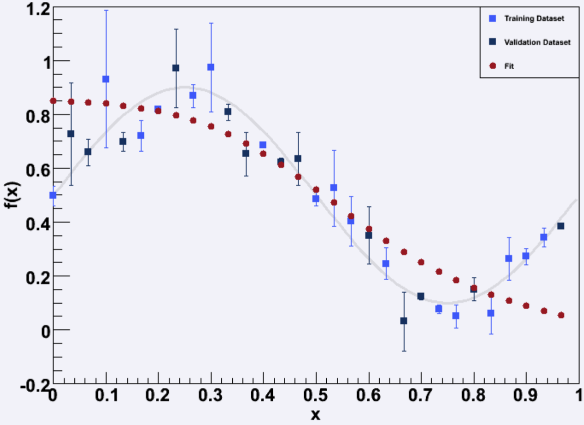
the minimization - a toy model

artificial data with noise



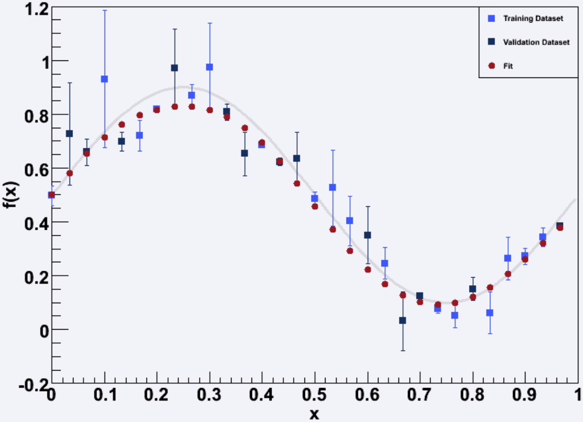
the minimization - a toy model

bad generalization - bad representation



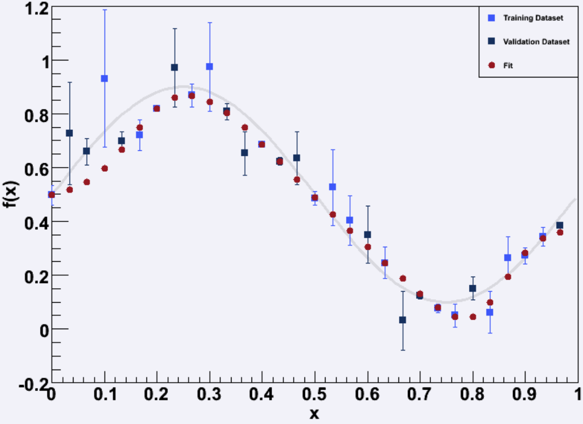
the minimization - a toy model

good generalization - good representation



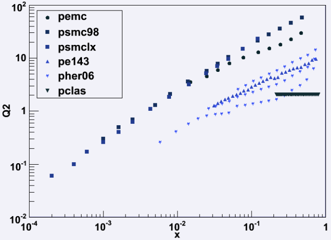
the minimization - a toy model

bad generalization - excellent representation

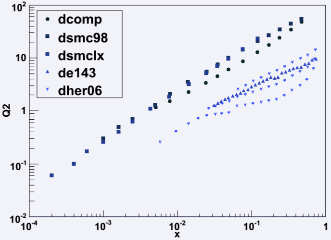


the experimental data

Proton	x range	$Q^2(GeV^2)$ range	N_{dat}
EMC	0.0150 - 0.466	3.50 - 29.5	10
SMC	0.0010 - 0.480	0.30 - 58.0	15
SMC low-x	0.0001 - 0.121	0.02 - 23.1	15
E143	0.0310 - 0.75	1.27 - 9.52	28
HERMES06	0.0058 - 0.7311	0.26 - 14.29	45

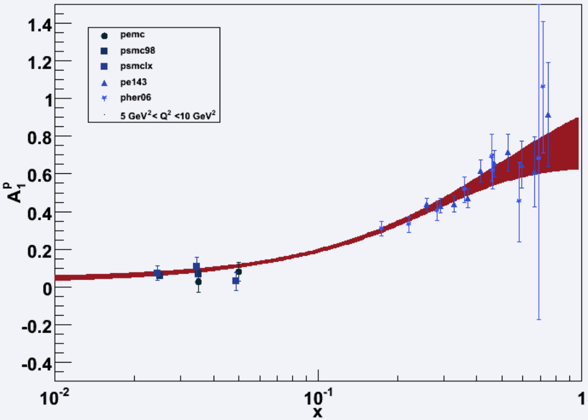


Deuteron	x range	$Q^2(GeV^2)$ range	N_{dat}
COMPASS	0.0051 - 0.474	1.18 - 47.5	12
SMC	0.0010 - 0.480	0.30 - 58.0	15
SMC low-x	0.0001 - 0.121	0.02 - 23.1	15
E143	0.0310 - 0.75	1.27 - 9.52	28
HERMES06	0.0058 - 0.7311	0.26 - 14.29	45



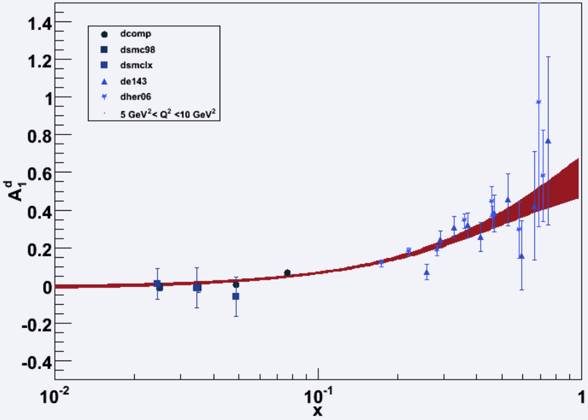
A_1 - proton

Exp.	χ^2
EMC	0.36
SMC98	0.39
SMC low x	0.99
E143	0.68
HERMES06	0.40



A_1 - deuteron

Exp.	χ^2
COMPASS	0.94
SMC98	1.08
SMC low-x	0.80
E143	1.51
HERMES06	0.97



a first summary

- we presented a parametrization of $A_1^{p,d}$ which takes into account all experimental information and which minimized the theoretical bias;
- one can use this parametrization to extract phenomenological quantities and assess the impact of the chosen assumptions.

applications

since

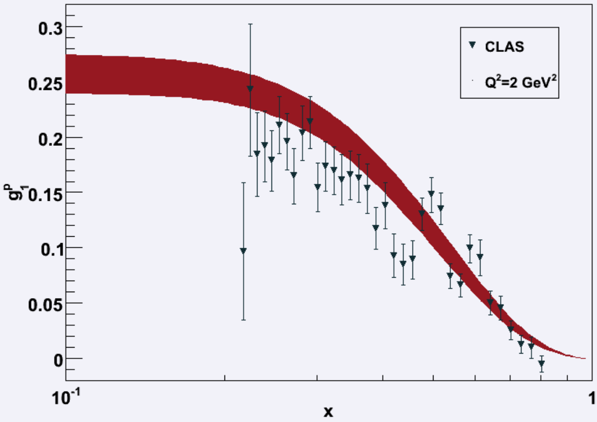
$$g_1(x, Q^2) = A_1(x, Q^2)F_1(x, Q^2) + \gamma^2 g_2(x, Q^2)\,;$$

$$F_1(x, Q^2) = F_2(x, Q^2)\frac{1+\gamma^2}{2x[1+R(x, Q^2)]}$$

we will supplement our parametrization with the following assumptions:

- R_{SLAC} [Whitlow et al. - 1990, Abe et al. - 1998]
- NNF2 [Forte, Latorre, Garrido, A.P. - 2002, Del Debbio, Forte, Latorre, A.P., Rojo - 2005]
- $g_2 = 0$ (g_2^{WW} is coming soon)

exercise 1: predictions and quark-hadron duality



exercise 2: integrals and errors

$\int_{0.021}^{0.9} dx g_1(x, Q^2 = 2.5\ GeV^2)$	DGP	HERMES
p	0.1182 ± 0.0069	0.1201 ± 0.0090
d	0.0454 ± 0.0049	0.0428 ± 0.0035
NS	0.1384 ± 0.0171	0.1477 ± 0.0167
$\int_{0.021}^{0.9} dx g_1(x, Q^2 = 5\ GeV^2)$	DGP	HERMES
p	0.1182 ± 0.0061	0.1211 ± 0.0092
d	0.0426 ± 0.0036	0.0436 ± 0.0035
NS	0.1444 ± 0.0143	0.1479 ± 0.0169

parametrization error estimation:

	$\int_{x_{min}}^{0.9} dx g_1^p(x, Q^2 = 5\ GeV^2)$
F_2	0.1182 ± 0.0052
x_{min}=0.00001	0.1897 ± 0.0318

exercise 2: integrals and errors

$\int_{0.021}^{0.9} dx g_1(x, Q^2 = 2.5\ GeV^2)$	DGP	HERMES
p	0.1182 ± 0.0069	0.1201 ± 0.0090
d	0.0454 ± 0.0049	0.0428 ± 0.0035
NS	0.1384 ± 0.0171	0.1477 ± 0.0167
$\int_{0.021}^{0.9} dx g_1(x, Q^2 = 5\ GeV^2)$	DGP	HERMES
p	0.1182 ± 0.0061	0.1211 ± 0.0092
d	0.0426 ± 0.0036	0.0436 ± 0.0035
NS	0.1444 ± 0.0143	0.1479 ± 0.0169

parametrization error estimation:

	$\int_{x_{min}}^{0.9} dx g_1^p(x, Q^2 = 5\ GeV^2)$
F_2	0.1182 ± 0.0052
x_{min}=0.00001	0.1897 ± 0.0318

exercise 3: sum rules [Abbate, Forte - 2005]

the bjorken sum rule:

- in order to evaluate

$$\Gamma_1^{NS}(Q^2) = 2\int_0^1 \left[g_1^p(x,Q^2) - \frac{g_1^d(x,Q^2)}{1-\frac{3}{2}\omega_D} \right]$$

we assume a low-x behaviour $A_1^{sx}(x,Q^2) \sim Kx^b$

- if we want to extract g_A, we need additional tools such as a coefficient function, HT models and TMC [A.P., Ridolfi - 1998, Bluemlein, Tkabladze - 1999]

$$\Gamma_1^{NS}(Q^2) = \frac{1}{6} g_A \Delta C_{NS}(\alpha_s(Q^2)) + \delta_{HT} + \delta_{TMC}\,;$$

- a first rough estimate (only one Q^2 bin, no HT, no TMC) gives

$$g_A = 1.593 \pm 0.208\,.$$

exercise 3: sum rules [Abbate, Forte - 2005]

the bjorken sum rule:

- in order to evaluate

$$\Gamma_1^{NS}(Q^2) = 2\int_0^1 \left[g_1^p(x,Q^2) - \frac{g_1^d(x,Q^2)}{1-\frac{3}{2}\omega_D} \right]$$

we assume a low-x behaviour $A_1^{sx}(x,Q^2) \sim Kx^b$

- if we want to extract g_A, we need additional tools such as a coefficient function, HT models and TMC [A.P., Ridolfi - 1998, Bluemlein, Tkabladze - 1999]

$$\Gamma_1^{NS}(Q^2) = \frac{1}{6} g_A \Delta C_{NS}(\alpha_s(Q^2)) + \delta_{HT} + \delta_{TMC}\,;$$

- a first rough estimate (only one Q^2 bin, no HT, no TMC) gives

$$g_A = 1.593 \pm 0.208\,.$$

delivery and outlook

the tool box contains:

- 1000 sets of parameters for both A_1^p and A_1^d;
- a driver code in F77, C++ and Python.

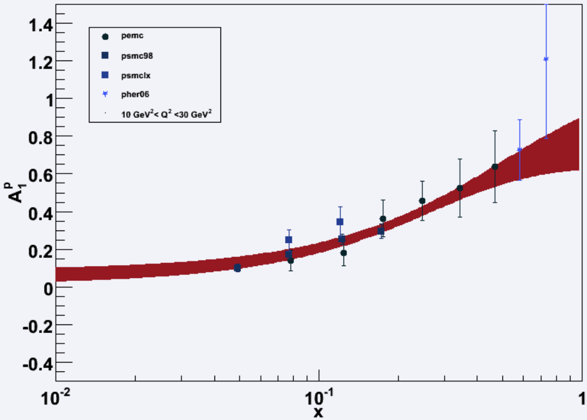
the future could be:

- SIDIS data;
- polarized PDFs.

some fit details

- 1000 replicas;
- 2-3-1 NN architecture (inputs: x, Q^2, output: A_1);
- 13 parameters;
- normalized between -1 and 1;
- minimization done with a genetic algorithm;
- C++ source code;
- 1h run time for 1000 reps. on an Intel 2.66GHz;

hardly compatible data - proton



hardly compatible data - deuteron